

FREEBSD



FreeBSD Handbook

Projeto de Documentação do FreeBSD

[Copyright](#) © 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003 The FreeBSD Documentation Project

Bem vindo ao FreeBSD! Este handbook aborda a instalação e o uso diário do *FreeBSD 4.10-RELEASE* e do *FreeBSD 5.3-RELEASE*. Este manual é um *trabalho em andamento* e é o resultado do trabalho de muitos indivíduos. Muitas sessões ainda não existem e algumas das que existem precisam ser atualizadas. Se você está interessado em ajudar com este projeto, envie um email para [lista de discussão do projeto de documentação do FreeBSD](#). A ultima versão deste documento está sempre disponível no [web site do projeto FreeBSD](#). Ele também pode ser obtido em uma grande variedade de formatos e opções de compactação a partir do [Servidor de FTP do projeto FreeBSD](#), ou em um dos numerosos [sites espelhos](#). Se você preferir ter uma cópia impressa deste handbook, você pode comprar uma na [FreeBSD Mall](#). Você também pode querer [fazer uma busca](#) no conteúdo do handbook.

Índice

[Prefácio](#)

I. [Iniciando](#)

1. [Introdução](#)
 - 1.1. [Sinópsse](#)
 - 1.2. [Bem-vindo ao FreeBSD!](#)
 - 1.3. [Sobre o projeto FreeBSD](#)
2. [Instalando FreeBSD](#)
 - 2.1. [Sinópsse](#)
 - 2.2. [Tarefas de pré-instalação](#)
 - 2.3. [Iniciando a instalação](#)
 - 2.4. [Introdução ao Sysinstall](#)
 - 2.5. [Alocando espaço em disco](#)
 - 2.6. [Escolhendo o que instalar](#)
 - 2.7. [Escolhendo a mídia de instalação](#)
 - 2.8. [Efetivando a instalação](#)
 - 2.9. [Pós-Instalação](#)
 - 2.10. [Hardware Suportado](#)
 - 2.11. [Resolução de problemas](#)
 - 2.12. [Guia avançado de instalação](#)
 - 2.13. [Preparando sua própria mídia de instalação](#)
3. [UNIX Básico](#)
 - 3.1. [Sinópsse](#)
 - 3.2. [Consoles virtuais e terminais](#)
 - 3.3. [Permissões](#)
 - 3.4. [Estrutura de Diretórios](#)
 - 3.5. [Organização de disco](#)
 - 3.6. [Montando e desmontando sistemas de arquivos](#)
 - 3.7. [Processos](#)

- 3.8. [Daemons, sinais e controle de processos](#)
- 3.9. [Shells](#)
- 3.10. [Editores de Texto](#)
- 3.11. [Dispositivos e Nós de Dispositivos](#)
- 3.12. [Formatos de binários](#)
- 3.13. [Para mais informações](#)
- 4. [Instalando aplicativos: Pacotes e Ports](#)
 - 4.1. [Sinópse](#)
 - 4.2. [Visão geral do processo de instalação de *software*](#)
 - 4.3. [Encontrando seu aplicativo](#)
 - 4.4. [Usando o sistema de pacotes](#)
 - 4.5. [Usando a coleção de *ports*](#)
 - 4.6. [Atividades pós-instalação](#)
 - 4.7. [Lidando com ports quebrados](#)
- 5. [O X Window System](#)
 - 5.1. [Sinopse](#)
 - 5.2. [Compreendendo o X](#)
 - 5.3. [Instalando o XFree86™](#)
 - 5.4. [Configuração do XFree86™](#)
 - 5.5. [Usando fontes no XFree86™](#)
 - 5.6. [O gerenciador de exibição do X](#)
 - 5.7. [Ambientes para a área de trabalho](#)

II. [Administração do Sistema](#)

- 6. [Configuração e Ajuste](#)
 - 6.1. [Sinópse](#)
 - 6.2. [Configuração Inicial](#)
 - 6.3. [Configuração Principal](#)
 - 6.4. [Configuração de Aplicações](#)
 - 6.5. [Inicializando Serviços](#)
 - 6.6. [Configurando o Utilitário *cron*](#)
 - 6.7. [Usando *rc* no FreeBSD 5.X](#)
 - 6.8. [Configurando Interfaces de Rede](#)
 - 6.9. [Servidores Virtuais](#)
 - 6.10. [Arquivos de Configuração](#)
 - 6.11. [Ajustando com o *sysctl*](#)
 - 6.12. [Ajustando Discos](#)
 - 6.13. [Ajustando Limites do Kernel](#)
 - 6.14. [Adicionando Espaço de Swap](#)
 - 6.15. [Gerenciamento de Energia e Recursos](#)
- 7. [O processo de inicialização do FreeBSD](#)
 - 7.1. [Sinópse](#)
 - 7.2. [Problema na inicialização](#)
 - 7.3. [A MBR, e os estágios de inicialização Um, Dois, e Três](#)
 - 7.4. [Interação do Kernel durante o boot](#)
 - 7.5. [Dicas de configuração de dispositivos](#)
 - 7.6. [Init: Inicialização do controle de processos](#)
 - 7.7. [Sequência do Shutdown](#)
- 8. [Usuários e Administração Básica de contas](#)
 - 8.1. [Sinópse](#)
 - 8.2. [Introdução](#)

- 8.3. [O super usuário](#)
- 8.4. [Contas do sistema](#)
- 8.5. [Contas de usuários](#)
- 8.6. [Modificando contas](#)
- 8.7. [Limitando usuários](#)
- 8.8. [Personalizando usuários](#)
- 8.9. [Grupos](#)
- 9. [Configurando o *kernel* do FreeBSD](#)
 - 9.1. [Sinópse](#)
 - 9.2. [Porque compilar um *kernel* customizado?](#)
 - 9.3. [Compilando e instalando um *kernel* customizado](#)
 - 9.4. [O arquivo de configuração](#)
 - 9.5. [Criando Dispositivos de Controle](#)
 - 9.6. [Se algo der errado](#)
- 10. [Segurança](#)
 - 10.1. [Sinopse](#)
 - 10.2. [Introdução](#)
 - 10.3. [Protegendo FreeBSD](#)
 - 10.4. [DES, MD5, e Crypt](#)
 - 10.5. [Senhas de uso único](#)
 - 10.6. [KerberosIV](#)
 - 10.7. [Kerberos5](#)
 - 10.8. [Firewalls](#)
 - 10.9. [OpenSSL](#)
 - 10.10. [VPN sobre IPsec](#)
 - 10.11. [OpenSSH](#)
 - 10.12. [Controle de Acesso Obrigatório \(MAC-Mandatory Access Control\)](#)
 - 10.13. [File System Access Control Lists](#)
 - 10.14. [Recomendações de Segurança do FreeBSD](#)
- 11. [Impressão](#)
 - 11.1. [Sinopse](#)
 - 11.2. [Introdução](#)
 - 11.3. [Configuração Básica](#)
 - 11.4. [Configuração Avançada de Impressora](#)
 - 11.5. [Usando as Impessoras](#)
 - 11.6. [Alternativas para o *Spooler* Padrão](#)
 - 11.7. [Correção de Erros](#)
- 12. [Armazenamento](#)
 - 12.1. [Sinópse](#)
 - 12.2. [Nomes de Dispositivos](#)
 - 12.3. [Adicionando Discos](#)
 - 12.4. [RAID](#)
 - 12.5. [Criando e utilizando mídias ópticas \(CDs & DVDs\)](#)
 - 12.6. [Criação e utilização de disquetes](#)
 - 12.7. [Criando e utilizando fitas DAT](#)
 - 12.8. [Backups em Disquetes](#)
 - 12.9. [Considerações básicas sobre backup](#)
 - 12.10. [Rede, Memória, e Sistemas de Arquivos em Imagens](#)
 - 12.11. [Snapshots de sistemas de arquivos](#)
 - 12.12. [Cotas de uso para sistemas de arquivos](#)

- 12.13. [Encriptando partições de um disco](#)
- 13. [O Gerenciador de Volumes Vinum](#)
 - 13.1. [Sinópse](#)
 - 13.2. [Discos são muito pequenos](#)
 - 13.3. [Gargalo de acesso](#)
 - 13.4. [Integridade de dados](#)
 - 13.5. [Objetos do Vinum](#)
 - 13.6. [Alguns exemplos](#)
 - 13.7. [Nomenclatura de objetos](#)
 - 13.8. [Configurando Vinum](#)
 - 13.9. [Usando Vinum para o Sistema de Arquivos Raiz](#)
- 14. [Localização - I18N/L10N Uso e Configuração](#)
 - 14.1. [Sinópse](#)
 - 14.2. [O Básico](#)
 - 14.3. [Usando a localização](#)
 - 14.4. [Compilando programas I18N](#)
 - 14.5. [Localizando o FreeBSD para línguas específicas](#)
- 15. [Aplicações de Desktop](#)
 - 15.1. [Sinópse](#)
 - 15.2. [Navegadores](#)
 - 15.3. [Ferramentas de produtividade](#)
 - 15.4. [Visualizadores de Documentos](#)
 - 15.5. [Aplicações Financeiras](#)
 - 15.6. [Sumário](#)
- 16. [Multimedia](#)
 - 16.1. [Sinópse](#)
 - 16.2. [Configurando a placa de som](#)
 - 16.3. [Audio MP3](#)
 - 16.4. [Reprodução de Video](#)
- 17. [Comunicações Seriais](#)
 - 17.1. [Sinópse](#)
 - 17.2. [Introdução](#)
 - 17.3. [Terminais](#)
 - 17.4. [Serviço de Dial-in](#)
 - 17.5. [Serviço de Dial-out](#)
 - 17.6. [Configurando uma console serial](#)
- 18. [PPP e SLIP](#)
 - 18.1. [Sinópse](#)
 - 18.2. [Utilizando PPP em modo usuário](#)
 - 18.3. [Usando PPP em modo kernel](#)
 - 18.4. [Resolução de Problemas](#)
 - 18.5. [Usando PPP sobre Ethernet \(PPPoE\)](#)
 - 18.6. [Usando **PPP** sobre ATM \(PPPoA\)](#)
 - 18.7. [Usando SLIP](#)
- 19. [Uso Avançado de Redes](#)
 - 19.1. [Índice](#)
 - 19.2. [Gateways e Rotas](#)
 - 19.3. [Rede Sem Fio \(Wireless\)](#)
 - 19.4. [Bluetooth](#)
 - 19.5. [Ligação \(Bridging\)](#)

- 19.6. [NFS](#)
- 19.7. [Operação Sem Disco \(Diskless Operation\)](#)
- 19.8. [Rede Digital de Serviços Integrados - RDSI \(ISDN\)](#)
- 19.9. [NIS/YP](#)
- 19.10. [DHCP](#)
- 19.11. [DNS](#)
- 19.12. [NTP](#)
- 19.13. [NAT - Tradução de Endereço de Rede](#)
- 19.14. [O "Super-Servidor" **inetd**](#)
- 19.15. [PLIP - IP em Linha Paralela](#)
- 19.16. [IPv6](#)
- 20. [Correio eletrônico](#)
 - 20.1. [Sinópse](#)
 - 20.2. [Usando o correio eletrônico](#)
 - 20.3. [Configuração do **sendmail**](#)
 - 20.4. [Trocando seu agente de transferência de mensagens](#)
 - 20.5. [Resolução de problemas](#)
 - 20.6. [Tópicos avançados](#)
 - 20.7. [SMTP com UUCP](#)
 - 20.8. [Usando o correio com uma conexão discada](#)
 - 20.9. [Autenticação SMTP](#)
- 21. [A Versão Mais Recente](#)
 - 21.1. [Sinopse](#)
 - 21.2. [FreeBSD-CURRENT vs. FreeBSD-STABLE](#)
 - 21.3. [Sincronizando Seus Fontes](#)
 - 21.4. [Usando o `make world`](#)
 - 21.5. [Preparando Múltiplas Máquinas](#)
- 22. [Compatibilidade Binária Linux](#)
 - 22.1. [Sinopse](#)
 - 22.2. [Instalação](#)
 - 22.3. [Instalando o Mathematica®](#)
 - 22.4. [Instalando o Maple™](#)
 - 22.5. [Instalando o MATLAB®](#)
 - 22.6. [Instalando Oracle®](#)
 - 22.7. [Instalando o SAP® R/3®](#)
 - 22.8. [Tópicos Avançados](#)

III. [Apêndices](#)

- A. [Obtendo o FreeBSD](#)
 - A.1. [Fornecedores de CDRoms e DVDs](#)
 - A.2. [Sítios para FTP](#)
 - A.3. [CVS Anônimo](#)
 - A.4. [Usando o CTM](#)
 - A.5. [Usando o CVSup](#)
 - A.6. [CVS Tags](#)
 - A.7. [Sítios AFS](#)
 - A.8. [Sítios rsync](#)
- B. [Bibliografia](#)
 - B.1. [Livros & Revistas Específicos sobre FreeBSD](#)
 - B.2. [Guias para Usuários](#)
 - B.3. [Guias para Administradores](#)

- B.4. [Guias para Programadores](#)
- B.5. [Detalhes Internos do Sistema Operacional](#)
- B.6. [Referente a Segurança](#)
- B.7. [Referente a *Hardware*](#)
- B.8. [História do UNIX®](#)
- B.9. [Revistas e Periódicos](#)
- C. [Recursos na Internet](#)
 - C.1. [Listas de Discussão](#)
 - C.2. [Usenet Newsgroups](#)
 - C.3. [World Wide Web Servers](#)
 - C.4. [Email Addresses](#)
 - C.5. [Shell Accounts](#)
- D. [Chaves PGP](#)
 - D.1. [Diretores](#)
 - D.2. [Membros do Core Team](#)
 - D.3. [Desenvolvedores](#)

[Considerações finais](#)

Lista de Tabelas

- 2-1. [Modelo de Inventário de Dispositivos](#)
- 2-2. [Disposição de partições para o primeiro disco](#)
- 2-3. [Disposição de partições para discos subsequentes](#)
- 2-4. [Perfis de segurança disponíveis](#)
- 2-5. [Nomes e significados das imagens ISO do FreeBSD](#)
- 3-1. [Códigos de Dispositivos de Disco](#)
- 12-1. [Convenções de Nomenclatura de Discos Físicos](#)
- 13-1. [Organizações do Plex no Vinum](#)
- 19-1. [Conectando um Cabo Paralelo para Conexão em Rede](#)

Lista de Figuras

- 2-1. [Menu de Configuração do Kernel](#)
- 2-2. [A interface visual do dispositivo de configuração do kernel](#)
- 2-3. [Lista de dispositivos expandida](#)
- 2-4. [Configuração de dispositivos sem conflitos](#)
- 2-5. [Menu principal do Sysinstall](#)
- 2-6. [Resultados típicos do reconhecimento de dispositivos](#)
- 2-7. [Selecione a saída do Sysinstall](#)
- 2-8. [Selecionando a opção *Usage* do menu do Sysinstall](#)
- 2-9. [Selecionando o menu de documentação](#)
- 2-10. [Menu de documentação do sysinstall](#)
- 2-11. [Menu principal do Sysinstall](#)
- 2-12. [Menu de mapeamento do teclado do Sysinstall](#)
- 2-13. [Menu principal do Sysinstall](#)
- 2-14. [Opções do Sysinstall](#)
- 2-15. [Iniciando a instalação padrão](#)
- 2-16. [Selecione o disco para o Fdisk](#)
- 2-17. [Partições típicas do Fdisk antes de sua edição](#)
- 2-18. [Partição Fdisk utilizando todo o disco](#)
- 2-19. [Menu do *Boot Manager* do FreeBSD](#)
- 2-20. [Sair da seleção de discos](#)
- 2-21. [Editor Disklabel do Sysinstall](#)
- 2-22. [Editor Disklabel do Sysinstall com padrões automáticos](#)

- 2-23. [Liberando espaço para a partição raiz](#)
- 2-24. [Editando o tamanho da partição raiz](#)
- 2-25. [Escolhendo o tipo da partição raiz](#)
- 2-26. [Escolhendo o ponto de montagem raiz](#)
- 2-27. [Editor Disklabel do Sysinstall](#)
- 2-28. [Escolha de distribuições](#)
- 2-29. [Confirmar distribuições](#)
- 2-30. [Escolha a mídia de instalação](#)
- 2-31. [Selecionando dispositivos de rede](#)
- 2-32. [Ajustando configuração de rede para ed0](#)
- 2-33. [Editando o arquivo `inetd.conf`](#)
- 2-34. [Configuração padrão para FTP anônimo](#)
- 2-35. [Editando a mensagem de boas-vindas do FTP](#)
- 2-36. [Editando o arquivo `exports`](#)
- 2-37. [Opções do Perfil de Segurança](#)
- 2-38. [Opções de configuração do console do sistema](#)
- 2-39. [Opções da proteção de tela](#)
- 2-40. [Tempo para proteção de tela](#)
- 2-41. [Saida da configuração do console do sistema](#)
- 2-42. [Selecione sua região](#)
- 2-43. [Selecione seu país](#)
- 2-44. [Selecione o Fuso-Horário](#)
- 2-45. [Selecione o tipo do protocolo de mouse](#)
- 2-46. [Definindo o protocolo do mouse](#)
- 2-47. [Configure a porta do mouse](#)
- 2-48. [Definindo a porta do mouse](#)
- 2-49. [Habilite o `daemon` do mouse](#)
- 2-50. [Teste o `daemon` do mouse](#)
- 2-51. [Configuração de alto-nível da rede](#)
- 2-52. [Selecione o MTA padrão](#)
- 2-53. [Configuração Ntpdate](#)
- 2-54. [Configuração da rede em baixo nível](#)
- 2-55. [Menu de seleção do método de configuração](#)
- 2-56. [Escolha o `desktop` padrão](#)
- 2-57. [Selecione a categoria do pacote](#)
- 2-58. [Seleção de pacotes](#)
- 2-59. [Instalação de pacotes](#)
- 2-60. [Confirmar instalação de pacotes](#)
- 2-61. [Escolha ``User''](#)
- 2-62. [Adicionando informações do usuário](#)
- 2-63. [Sair do gerenciamento de usuário e grupo](#)
- 2-64. [Sair da instalação](#)
- 13-1. [Organização Concatenada](#)
- 13-2. [Organização *Striped*](#)
- 13-3. [Organização RAID-5](#)
- 13-4. [Um Volume Simples de Vinum](#)
- 13-5. [Um Volume de Vinum Espelhado](#)
- 13-6. [Um volume Vinum *Striped*](#)
- 13-7. [Um Volume Vinum *Striped* e Espelhado](#)

Lista de Exemplos

- 2-1. [Utilizando como está, uma partição já existente](#)
 - 2-2. [Redimensionando uma partição existente](#)
 - 3-1. [Modelo de disco, partição e nomes](#)
 - 3-2. [Modelo conceitual de um disco](#)
 - 4-1. [Baixando manualmente um pacote e instalando-o localmente](#)
 - 6-1. [Criando um Arquivo de Swap no FreeBSD 4.X](#)
 - 6-2. [Criando um Arquivo de Swap no FreeBSD 5.X](#)
 - 7-1. [boot0 Screenshot](#)
 - 7-2. [boot2 Screenshot](#)
 - 7-3. [Um Console Insecure em /etc/ttys](#)
 - 8-1. [Configurando adduser e adicionando um usuário no FreeBSD 4.X](#)
 - 8-2. [Adicionando um usuário em FreeBSD 5.X](#)
 - 8-3. [rmuser Remoção interativa de contas](#)
 - 8-4. [Usando o chpass de forma interativa pelo super usuário](#)
 - 8-5. [Usando o chpass de forma interativa por um usuário normal](#)
 - 8-6. [Alterando sua senha](#)
 - 8-7. [Alterando a senha de outro usuário, tendo os privilégios do super usuário](#)
 - 8-8. [Adicionando um grupo por meio do pw\(8\)](#)
 - 8-9. [Adicionando alguém a um determinado grupo por meio do pw\(8\)](#)
 - 8-10. [Usando o id\(1\) para determinar grupos associados aos usuários](#)
 - 10-1. [Using SSH to Create a Secure Tunnel for SMTP](#)
 - 12-1. [Utilizando o dump sobre o ssh](#)
 - 12-2. [Um Script para a criação de disquetes de inicialização](#)
 - 12-3. [Utilizando o vnconfig para montar uma imagem de um sistema de arquivos no FreeBSD 4.X](#)
 - 12-4. [Criando uma nova imagem de disco com o vnconfig](#)
 - 12-5. [Usando o mdconfig para montar um arquivo de imagem de um sistema de arquivos no FreeBSD 5.X](#)
 - 12-6. [Criando um novo disco em arquivo de imagem com o mdconfig](#)
 - 12-7. [Discos de memória md no FreeBSD 4.X](#)
 - 12-8. [Criando um novo disco baseado em memória com o mdconfig](#)
 - 12-9. [Criando um novo disco baseado em memória com o mdmfs](#)
 - 17-1. [Adicionando entradas de terminais em /etc/ttys](#)
 - 19-1. [Montando uma Exportação com amd](#)
 - 19-2. [Escritório Filial ou Rede Doméstica](#)
 - 19-3. [Escritório Principal ou Outra Rede Local](#)
 - 19-4. [Enviando um Sinal HangUP ao inetd](#)
 - 20-1. [Configurando o banco de dados de acesso do sendmail](#)
 - 20-2. [Apelidos de email](#)
 - 20-3. [Exemplo de mapeamento de domínio virtual de mensagens](#)
 - A-1. [Sincronizando Alguma Coisa de -CURRENT \(ls\(1\)\) e Apagando-a Depois:](#)
 - A-2. [Sincronizando a Versão do ls\(1\) no Ramo 3.X-STABLE:](#)
 - A-3. [Criando uma Lista de Alterações \(Unified Diffs\) do ls\(1\)](#)
 - A-4. [Procurando que Outros Nomes de Módulos Podem Ser Usados:](#)
-

Prefácio

Audiências Pretendidas

O recém chegado ao FreeBSD encontrará que a primeira seção deste livro guia o usuário com o processo da instalação de FreeBSD e introduz delicadamente os conceitos e as convenções do UNIX®. Trabalhar através desta seção requer pouco mais do que o desejo explorar, e a habilidade de acoplar novos conceitos como são introduzidos.

Uma vez que você viajou esta distância, a segunda, de longe a maior, seção do handbook é uma referência detalhada dos tópicos do interesse aos administradores de sistema do FreeBSD. Alguns destes capítulos podem recomendar que você faça alguma leitura prévia, e isto é anotado na sinópse no começo de cada capítulo.

Para uma lista de fontes adicionais de informação, veja por favor [Apêndice B](#).

Mudanças da primeira edição

Esta segunda edição é a culminação de dois anos de trabalho dos membros dedicados do projeto de documentação do FreeBSD. As principais mudanças desta nova edição estão a seguir:

- Um índice completo foi adicionado.
- Todas as figuras ASCII foram substituídas por diagramas gráficos.
- Uma sinopse padrão foi adicionada a cada capítulo para dar um rápido sumário de que informação o capítulo contém, e o que o leitor espera saber.
- O conteúdo foi reorganizado logicamente em três partes: ``Começando a usar'', ``Administração do Sistema'', e ``Apêndices''.
- [Capítulo 2](#) (``Instalando o FreeBSD'') foi completamente reescrito com muitos *screenshots* para fazê-lo muito mais fácil para novos usuários compreender o texto.
- [Capítulo 3](#) (``UNIX básico'') foi expandido para conter informações adicionais sobre processos, daemons e sinais.
- [Capítulo 4](#) (``Instalando Aplicações'') foi expandido para conter informações adicionais sobre o gerenciamento de pacote binário.
- [Capítulo 5](#) (``O X Window System'') foi completamente reescrito com uma ênfase em uso de modernas tecnologias *desktop* como o **KDE** e **GNOME** no XFree86™ 4.X.
- [Capítulo 7](#) (``O Processo de inicialização do FreeBSD'') foi expandido.
- [Capítulo 12](#) (``Armazenamento'') foi escrito para ser dois capítulos separados em ``Discos'' e ``Backups''. Nós sentimos que os tópicos são mais fáceis de se

- compreender quando apresentados como um único capítulo. Uma seção sobre RAID (ambos hardware e software) foi adicionado também.
- [Capítulo 17](#) ("Comunicações Seriais") foi completamente reorganizada e atualizada para o FreeBSD 4.X/5.X.
 - [Capítulo 18](#) ("PPP e SLIP") foi substancialmente atualizado.
 - Muitas seções novas foram adicionadas ao [Capítulo 19](#) ("Rede avançada").
 - [Capítulo 20](#) ("Correio Eletrônico") foi expandido para incluir mais informações sobre configurando o **sendmail**.
 - [Capítulo 22](#) ("Compatibilidade Binária Linux") foi expandido para incluir informação sobre instalando o **Oracle®** e **SAP® R/3®**.
 - Os seguintes tópicos novos são cobertos nesta segunda edição:
 - Configuração e Ajuste ([Capítulo 6](#)).
 - Multimedia ([Capítulo 16](#))

Organização deste Livro

Este livro é dividido em três seções logicamente distintas. A primeira seção, *Começando a usar*, cobre a instalação e o uso básico do FreeBSD. Espera-se que o leitor seguirá estes capítulos em sequência, possivelmente saltando capítulos que cobrem tópicos familiares. A segunda seção, *Administração do Sistema*, cobre uma broad coleção dos assuntos que são do interesse para usuários mais avançados do FreeBSD. Cada seção começa com uma sinópsese sucinta que descreve o que o capítulo cobre e que o leitor espera já saber. Isto é significa permitir ao leitor ocasionalmente pular capítulos para encontrar os de seu interesse. A terceira seção contém apêndices de informações referênciais.

[Capítulo 1](#), Introdução

Introduz o FreeBSD ao novo usuário. Descreve a história do Projeto FreeBSD, seus objetivos e modelo de desenvolvimento.

[Capítulo 2](#), Instalação

Caminha um usuário através do processo inteiro de instalação. Alguns tópicos avançados da instalação, como instalação através de um console serial, são cobertos também.

[Capítulo 3](#), UNIX básico

Cobre os comandos e as funcionalidades básicos do sistema operacionl FreeBSD. Se você for familiar com o Linux ou outro sabor do UNIX então você pode provavelmente saltar este capítulo.

[Capítulo 4](#), Instalando aplicações

Cobre a instalação de software de terceiros com ambas inovações do FreeBSD, a "Coleção de Ports" e pacotes binários padrão.

[Capítulo 5](#), O X Window System

Descreve o sistema de janelas X em geral e o uso do **XFree86** no FreeBSD em particular. Também descreve ambientes comuns de *desktop* como **KDE** e **GNOME**.

[Capítulo 6](#), *Configuração e Ajuste*

Descreve os parâmetros disponíveis para os administradores de sistema ajustar um sistema FreeBSD para uma ótima performance. Descreve também os vários arquivos de configuração usados no FreeBSD e onde encontrá-los.

[Capítulo 7](#), *O Processo de inicialização*

Descreve o processo de inicialização do FreeBSD e explica como controlar este processo com as opções de configuração.

[Capítulo 8](#), *Usuários e administração básica de contas*

Descreve a criação e manipulação de contas de usuários. Discute também limitações de recursos que podem ser ajustados nos usuários e em outras tarefas de gerenciamento de contas.

[Capítulo 9](#), *Configurando o kernel do FreeBSD*

Explica porque você pode precisar configurar um novo kernel e fornece instruções detalhadas para configuração, construção, e instalação de um kernel customizado.

[Capítulo 10](#), *Segurança*

Descreve diferentes tipos de ferramentas disponíveis para ajudar a manter seu sistema FreeBSD seguro, incluindo Kerberos, IPsec, OpenSSH, e firewalls de rede.

[Capítulo 11](#), *Impressão*

Descreve o gerenciamento de impressoras no FreeBSD, incluindo informação sobre páginas de banner, contas de impressão, e configuração inicial.

[Capítulo 12](#), *Armazenamento*

Descreve como gerenciar meios de armazenamento e sistema de arquivos com o FreeBSD. Iste inclui discos físicos, RAID arrays, mídia óptica e fita, discos de memória-backed, e sistema de arquivos de rede.

[Capítulo 13](#), *Vinum*

Descreve como usar o Vinum, um gerenciador lógico de volume que fornece discos lógicos de dispositivos independentes, e software RAID-0, RAID-1 e RAID-5.

[Capítulo 14](#), *Localização*

Descreve como usar o FreeBSD em outras línguas que não seja o Inglês. Cobre ambos sistemas e localização a nível de aplicações.

[Capítulo 15](#), *Aplicações de desktop*

Lista algumas das aplicações comuns de *desktop* como navegadores web e suites de produtividade, e descreve como instalá-los no FreeBSD.

[Capítulo 16](#), *Multimedia*

Mostra como configurar o suporte a playback de som e vídeo para seu sistema. Também descreve alguns exemplos de aplicações de áudio e vídeo.

[Capítulo 17](#), *Comunicações Seriais*

Explica como conectar terminais e modems ao seu sistema FreeBSD para ambas conexões dial in e dial out.

[Capítulo 18](#), *PPP e SLIP*

Descreve como usar PPP, SLIP, ou PPP over Ethernet para conectar a sistemas remotos com o FreeBSD.

[Capítulo 19](#), *Rede avançado*

Descreve muitos tópicos de rede, incluindo compartilhamento em uma conexão de Internet com outros computadores em sua LAN, usando network filesystems, compartilhando informações de contas via NIS, configurando um servidor de nomes, e muito mais.

[Capítulo 20](#), *Correio Eletrônico*

Explica os componentes diferentes de um servidor de email e entra em tópicos simples de configuração para o mais popular software de servidor de mail: o **sendmail**.

[Capítulo 21](#), *A versão mais recente*

Explica as diferenças entre FreeBSD-STABLE, FreeBSD-CURRENT, e FreeBSD releases. Descreve que usuários beneficiarão seguindo um sistema de desenvolvimento e os esboços do processo.

[Capítulo 22](#), *Compatibilidade Binária Linux*

Descreve as características da compatibilidade com Linux do FreeBSD. Fornece também instruções de instalação detalhadas para muitas aplicações populares de Linux como **Oracle**, **SAP R/3**, and **Mathematica®**.

[Apêndice A](#), *Obtendo o FreeBSD*

Lista fontes diferentes para obtenção da mídia do FreeBSD em CDROM ou DVD tão bem como diferentes sites na Internet que permitem você obter e instalar o FreeBSD.

[Apêndice B](#), *Bibliografia*

Este livro toca em muitos assuntos diferentes que pode deixar você com vontade de uma explicação mais detalhada. As listas bibliografias de muitos livros excelentes que são referenciados no texto.

[Apêndice C](#), *Recursos na Internet*

Descreve muitos forums disponíveis para usuários do FreeBSD enviar perguntas e acoplam em conversações técnicas sobre o FreeBSD.

[Apêndice D](#), *PGP Keys*

Listas das PGP fingerprints dos diversos Desenvolvedores do FreeBSD.

Convenções usadas neste livro

Para fornecer um texto consistente e fácil de ler, diversos convenções são seguidas durante todo o livro.

Convenções Tipográficas

Itálico

Uma fonte *itálico* é usada para nome de arquivos, URLs, texto enfatizado, e o primeiro uso de termos técnicos.

Mono espaço

Uma fonte *mono espaçada* é usada para mensagens de erro, comandos, variáveis de ambiente, nomes de ports, hostnames, nome de usuários, nomes de grupos, nomes de dispositivos, variáveis, e fragmentos de códigos.

Bold

Uma fonte **bold** é usada para aplicações, comandos, e teclas.

Entrada de usuário

Teclas são apresentadas em **bold** para destacar-se de outros textos. Combinações de teclas, que significa serem digitadas simultaneamente são apresentadas com '+' entre as teclas, como:

Ctrl+Alt+Del

Significa que o usuário deve digitar as teclas **Ctrl**, **Alt**, e **Del** ao mesmo tempo.

Teclas para ser digitadas em sequência estarão separadas com vírgulas, por exemplo:

Ctrl+X, Ctrl+S

Significaria que o usuário se espera digitar as teclas **Ctrl** e **X** simultaneamente e então digite as teclas **Ctrl** e **S** simultaneamente.

Exemplos

Exemplos que começam com `E:\>` indica um comando MS-DOS®. A menos que notado de outra maneira, estes comandos podem ser executados de uma janela de "Prompt de comando" em um moderno ambiente Microsoft® Windows®.

```
E:\> tools\fdimage floppies\kern.flp A:
```

Exemplos que começam com `#` indicam que o comando deve ser invocado como o superusuário no FreeBSD. Você pode logar como `root` para digitar o comando, ou logar como sua conta normal e usar [su\(1\)](#) para obter privilégios de superusuário.

```
# dd if=kern.flp of=/dev/fd0
```

Exemplos que começam com `%` indicam um comando que deve ser invocado de uma conta de usuário normal. A menos que notado de outra maneira, a sintaxe C-shell é usada para ajustar variáveis de ambiente e outros comandos shell.

```
% top
```

Reconhecimentos

O livro que você está segurando representa os esforços de muitas centenas de pessoas em todo o mundo. Se enviaram reparos para erros de ortografia, ou submeteram capítulos completos, todas as contribuições foram úteis.

Diversas companhias suportaram o desenvolvimento deste documento pagando autores para trabalhar em tempo integral nele, pagando por publicação, etc. Em particular, a BSDi (adquirida subseqüentemente pela [Wind River Systems](#)) pagou membros do Projeto de Documentação para trabalhar em melhorias deste livro em tempo integral que conduziu à publicação da primeira edição impressa em Março de 2000 (ISBN 1-57176-241-8). A Wind River Systems então pagou diversos autores adicionais para fazer um número de melhorias a infraestrutura de saída para impressão e para adicionar capítulos adicionais ao texto. Este trabalho culminou na publicação da segunda edição impressa em Novembro de 2001 (ISBN 1-57176-303-1).

I. Iniciando

Esta parte do handbook é destinada aos usuários e administradores que são iniciantes no FreeBSD. Estes capítulos:

- Introduzem você ao FreeBSD.
- Guiam você através do processo de instalação.
- Ensinam UNIX® básico e os seus fundamentos.
- Mostra como instalar de forma segura as aplicações de terceiros disponíveis para o FreeBSD.
- Introduzem você ao X, o ambiente gráfico do UNIX, e detalha como configurar um ambiente desktop para que você se torne mais produtivo.

Nós tentamos manter o número de referências à informações a serem abordadas nas sessões futuras ao mínimo, de modo que você pudesse ler esta seção do manual do começo ao final, sem ter que ficar indo e voltando nas páginas.

Índice

1. [Introdução](#)
 2. [Instalando FreeBSD](#)
 3. [UNIX Básico](#)
 4. [Instalando aplicativos: Pacotes e Ports](#)
 5. [O X Window System](#)
-

Capítulo 1. Introdução

Índice

1.1. [Sinópse](#)

1.2. [Bem-vindo ao FreeBSD!](#)

1.3. [Sobre o projeto FreeBSD](#)

Reestruturado, reorganizado e parcialmente reescrito por Jim Mock.

1.1. Sinópse

Muito obrigado por seu interesse no FreeBSD! O capítulo a seguir cobre o Projeto FreeBSD em vários aspectos, como sua história, objetivos, modelo de desenvolvimento e outros.

Após a leitura deste capítulo, você saberá:

- Como FreeBSD se compara a outros sistemas operacionais de computadores.
- A história do projeto FreeBSD.
- Os objetivos do Projeto FreeBSD.
- A base do modelo *open-source* de desenvolvimento do FreeBSD.
- E é claro: de onde o nome "FreeBSD" surgiu.

1.2. Bem-vindo ao FreeBSD!

FreeBSD é um sistema operacional baseado no 4.4BSD-Lite para computadores Intel (x86), DEC Alpha™, e Sun UltraSPARC®. Versões para outras arquiteturas também estão a caminho. Você também pode ler sobre [a história do FreeBSD](#), ou a [versão corrente](#). Se estiver interessando em contribuir com o projeto (código, equipamento, contas diversas), veja o artigo [Contribuindo com o FreeBSD](#).

1.2.1. O que o FreeBSD pode fazer?

FreeBSD tem muitas características valiosas. Algumas destas são:

- *Multitarefa preemptiva* com ajustes dinâmicos de prioridade que garantem compartilhamento claro e racional do computador entre as aplicações e usuários, mesmo sob a mais intensa demanda.
- *Características multi-usuário* que permite várias pessoas utilizarem um sistema FreeBSD de forma simultânea, para uma variedade de coisas. Isto implica, por exemplo, que os periféricos do sistema como impressoras e dispositivos de fita serão apropriadamente compartilhados entre todos usuários no sistema ou na rede, e que limites individuais possam ser definidos para usuários e grupos de usuários, protegendo recursos críticos do sistema de sobrecarga.
- Forte *rede TCP/IP* com suporte a padrões industriais como SLIP, PPP, NFS, DHCP e NIS. Isto significa que sua estação FreeBSD pode interagir facilmente com outros sistemas da mesma forma que pode agir como um servidor corporativo, oferecendo funções vitais como NFS (acesso remoto à arquivos) e

- serviços de correio eletrônico, ou então colocando sua empresa na Internet com serviços de WWW, FTP, roteamento e firewall (segurança).
- *Proteção de memória* garante que aplicações (ou usuários) não interferirão entre si. A falha de uma aplicação não afetará outras de forma alguma.
 - FreeBSD é um sistema operacional *32-bit* (*64-bit* em plataforma Alpha e UltraSPARC) e foi projetado como tal desde seu princípio.
 - O sistema de interface gráfica, *X Window System* (X11R6) padrão industrial provê uma interface gráfica com o usuário (GUI) ao custo de uma placa VGA comum e um monitor, e ainda vem com código fonte completo.
 - *Compatibilidade binária* com quaisquer programas compilados para Linux, SCO, SVR4, BSDI and NetBSD.
 - Milhares de aplicações prontas para imediata utilização (*ready-to-run*) estão disponíveis a partir da coleção de *ports* e *packages* do FreeBSD. Por quê procurar na rede quando é possível encontrar tudo bem aqui?
 - Milhares de aplicações adicionais e de fácil portabilidade estão disponíveis na Internet. FreeBSD tem código fonte compatível com a maioria dos sistemas UNIX® comerciais mais populares, e devido a isto a maioria das aplicações requerem pouca, ou nenhuma modificação para compilar corretamente.
 - *Memória virtual* paginada por demanda e uma concepção eficiente (*Merged VM/buffer cache*) que satisfaz a necessidade de recursos de aplicações com grande apetite para memória, ao mesmo tempo que mantém resposta interativa aos outros usuários.
 - Suporte *SMP* para máquinas com múltiplas CPUs.
 - Conjunto completo de ferramentas de desenvolvimento em linguagem *C*, *C++*, *Fortran*, e *Perl*. Muitas linguagens adicionais para pesquisa e desenvolvimento avançado também estão disponíveis na coleção de *ports* e *packages*.
 - *Código fonte* disponível para todo o sistema significa que você tem o nível mais completo de controle sobre seu ambiente. Por quê manter-se preso à soluções proprietárias e à mercê do vendedor quando se pode ter um sistema verdadeiramente aberto?
 - Extensa *documentação online*.
 - *E muito mais!*

FreeBSD é baseado a versão 4.4BSD-Lite do Grupo de Pesquisa em Sistemas Computacionais (CSRG) da Universidade da Califórnia, Berkeley, e carrega consigo a distinta tradição do desenvolvimento de sistemas BSD. Em adição ao aprimorado trabalho realizado pelo CSRG, o projeto FreeBSD acrescentou ainda muitos milhares de horas no aperfeiçoamento do sistema para máxima performance e confiabilidade em situações de demanda da vida real. Enquanto muitos dos gigantes comerciais se esforçam no campo dos sistemas operacionais para PC afim de oferecer tais funcionalidades com performance e confiabilidade, o FreeBSD os oferece *agora!*

As aplicações para as quais o FreeBSD pode ser utilizado só estão limitadas por sua própria imaginação. Do desenvolvimento de software à automação industrial, controle de inventário à correção remota da orientação de antenas de satélite; se pode ser feito com um produto UNIX comercial, é muito provável que possa ser feito com FreeBSD também! FreeBSD também se beneficia de forma significativa de literalmente milhares de aplicações de alta qualidade desenvolvidas por centros de pesquisa e universidades ao redor do mundo, usualmente disponíveis à baixo ou nenhum custo. Aplicações comerciais também estão disponíveis e surgindo em grande número a cada dia.

Pelo fato do código fonte para o FreeBSD estar geralmente disponível, o sistema pode também ser customizado a um grau incrível para aplicações ou projetos especiais, e de formas geralmente não viáveis com sistemas operacionais dos principais vendedores comerciais. Aqui está apenas uma amostra das aplicações onde o FreeBSD está sendo utilizado:

- *Serviços Internet:* A infra-estrutura robusta de rede TCP/IP criada no FreeBSD torna-o plataforma ideal para uma variedade de serviços Internet tais como:
 - servidores FTP
 - servidores *Web* (padrão ou seguro [SSL])
 - Servidores de interligação de redes (*gateway*), *firewall* e NAT (``Substituição de IPs").
 - Servidores de Correio Eletrônico
 - Grupo de notícias USENET ou sistemas de BBS
 - E mais...

Com FreeBSD, você pode facilmente começar com um ambiente pequeno com um PC 386, de baixo custo, e atualizar todo o sistema para um Xeon quadri-processado com sistema de armazenamento RAID, de acordo com o crescimento de sua empresa.

- *Educação:* Você é estudante de ciências computacionais ou algum campo relacionado à engenharia ? Não há melhor maneira para aprender sobre sistema operacionais, arquitetura de computadores ou redes do que colocando a mão na massa, a partir da base de experiência que o FreeBSD pode oferecer. Uma série de aplicações CAD grauitamente disponíveis, pacotes de projeto e desenho gráfico e de matemática tornam o sistema altamente útil também para aqueles cujo interesse primário em um computador é realizar *outras* tarefas de forma prática !
- *Pesquisa:* Com código fonte do sistema inteiro disponível, o FreeBSD é uma plataforma excelente para pesquisa em sistema operacionais bem como outros vários ramos de ciências computacionais. A natureza livre do FreeBSD também possibilita que grupos remotos colaborem mutuamente entre sí com idéias ou desenvolvimento compartilhado sem a necessidade de se preocupar com acordos especiais de licenciamento ou limitações do que pode ser discutido em um fórum aberto.
- *Rede:* Precisa de um novo roteador ? Um novo servidor de nomes (DNS) ? Um *firewall* para manter outras pessoas fora da sua rede interna ? O FreeBSD pode facilmente tornar aquele PC 386 ou 486 parado em um canto qualquer, em um avançado roteador com sofisticada capacidade de filtragem de pacotes.
- *Estação de trabalho gráfica:* FreeBSD é uma boa escolha para solução de um terminal gráfico de baixo custo, seja utilizando o servidor XFree86™ (disponível livremente) ou um dos excelentes servidores comerciais oferecidos pela [Xi Graphics](#). Diferente de um terminal X, o FreeBSD possibilita que várias aplicações possam ser executadas localmente, se desejado, desta forma aliviando a carga de um servidor central. FreeBSD pode até mesmo ser inicializado sem depender da existência de uma unidade de disco local (``diskless"), tornando estações de trabalho ainda mais baratas e mais fáceis de administrar.

- *Desenvolvimento de Software*: Um sistema FreeBSD básico vem com um conjunto completo de ferramentas de desenvolvimento, incluindo o renomado compilador e depurador C/C++ GNU.

FreeBSD está disponível tanto em código fonte quanto de forma binária em CDROM, DVD, e por meio de FTP anônimo. Refira-se ao [Apêndice A](#) para mais informações sobre como obter o FreeBSD.

1.2.2. Quem utiliza FreeBSD?

FreeBSD é utilizado para servir alguns dos maiores sítios da Internet, incluindo:

- [Yahoo!](#)
- [Apache](#)
- [Blue Mountain Arts](#)
- [Pair Networks](#)
- [Sony do Japão](#)
- [Netcraft](#)
- [Weathernews](#)
- [Supervalu](#)
- [TELEHOUSE America](#)
- [Anti-Vírus Sophos](#)
- [JMA Wired](#)

e muito mais.

1.3. Sobre o projeto FreeBSD

A seção a seguir oferece algumas informações básicas sobre o projeto, incluindo uma breve história, seus objetivos e seu modelo de desenvolvimento.

1.3.1. Uma Breve História do FreeBSD

Contribuição de Jordan Hubbard.

O projeto FreeBSD teve seu nascimento no início de 1993, em parte como uma consequência do conjunto de manutenção não-oficial do 386BSD ("Unofficial 386BSD Patchkit") pelos seus 3 últimos coordenadores: Nate Williams, Rod Grimes e eu.

Nosso objetivo original era produzir um *snapshot* intermediário do 386BSD, de forma a poder corrigir uma série de problemas com este sistema, que o mecanismo de manutenção não era capaz de resolver. Alguns se lembrarão do nome inicial do projeto que era "386BSD 0.5" ou "386BSD Interim" em referência a este fato.

386BSD era o sistema operacional de Bill Jolitz, que já estava naquele instante sofrendo quase um ano de negligência. Como o mecanismo de manutenção *patchkit* se tornava mais e mais desconfortável a cada dia que passava, fomos unânimes em decidir que algo

tinha que ser feito e decidimos ajudar Bill oferecendo a ele este *snapshot* "interim". Tais planos foram bruscamente interrompidos quando Bill Jollitz repentinamente decidiu retirar sua sanção ao projeto sem nenhuma indicação clara do que deveria ser feito.

Não levou muito para decidirmos que o objetivo continuava a valer a pena, mesmo sem a ajuda de Bill, e então adotamos o nome "FreeBSD", sugerido por David Greenman. Nossos objetivos iniciais foram definidos depois de consultar os usuários recentes do sistema e, uma vez estando claro que o projeto estava na estrada para, talvez, tornar-se uma realidade, entrei em contato com a Walnut Creek CDROM, com o olho aberto à possibilidade de aperfeiçoar os canais de distribuição do FreeBSD para as pessoas que não tinham acesso à Internet. Walnut Creek CDROM não apenas aprovou a idéia de distribuir o FreeBSD em CD, mas também foi mais longe, ao ponto de oferecer ao projeto uma máquina para trabalho dedicado e uma conexão rápida com a Internet. Sem esta confiança, sem precedentes, da Walnut Creek CDROM no que era, naquele momento, um projeto completamente desconhecido, é muito provável que o FreeBSD não tivesse chegado tão longe e tão rápido ao ponto em que está hoje.

A primeira distribuição em CDROM (e na Internet em geral) foi o FreeBSD 1.0, lançado em Dezembro de 1993. Era baseado na fita 4.3BSD-Lite ("Net/2") da Universidade da Califórnia, Berkeley (U.C. Berkeley), com muitos componentes originados do 386BSD e da Fundação do Software Livre (*Free Software Foundation*). Foi um sucesso razoavelmente grande para uma primeira aparição, e nós continuamos o ciclo com uma versão altamente bem sucedida, o FreeBSD 1.1 *release* de Maio de 1994.

Por volta desta época, algumas núvens de tempestade inesperadas começaram a se formar no horizonte, conforme a Novell e U.C. Berkeley acertaram ao longo do processo penal entre ambas, a respeito da situação legal da fita contendo o Net/2 de Berkeley. Uma das condições do acordo eram as concessões da U.C. Berkeley que implicava que grandes trechos do Net/2 fossem códigos "impedidos" e de propriedade da Novell, que havia por sua vez adquirido-os da AT&T algum tempo antes. O que Berkeley recebeu em retribuição foi a "benção" da Novell para o lançamento da versão 4.4BSD-Lite, que quando acontecesse, seria declarado como "impedido" e todos os usuários do Net/2 seriam fortemente encorajados a mudar de sistema para a nova versão. Isso incluiu o FreeBSD, ao projeto foi dado o prazo final de Julho de 1994 para parar de distribuir seu produto baseado na versão Net/2. Sob tais termos de acordo, o projeto poderia lançar uma última versão antes do prazo em questão, o que originou o FreeBSD 1.1.5.1.

O FreeBSD definiu então a árdua tarefa de literalmente se reinventar à partir de um sistema completamente novo e consideravelmente incompleto, o 4.4BSD-Lite. As versões "Lite" continham grandes blocos de código a menos, removidos pelo CSRG de Berkeley (devido a várias decisões legais), códigos necessários para a construção de um sistema inicializável e que podia ser utilizado em produção e o fato é, que a conversão do 4.4 para a plataforma Intel era altamente incompleta. O projeto levou até Novembro de 1994 para concluir esta transição, quando lançou a versão 2.0 do FreeBSD na rede mundial e em CDROM (em Dezembro). Apesar de um pouco bruta naquele instante, a versão teve um sucesso significativo, e foi seguida pelo FreeBSD 2.0.5, mais robusto e de mais fácil instalação, em Junho de 1995.

Lançamos o FreeBSD 2.1.5 em Agosto de 1996, que foi bastante popular entre os provedores de internet (ISP) e as empresas a ponto de justificar a viabilidade de outra versão no ramo 2.1-STABLE. Esta versão foi o FreeBSD 2.1.7.1, lançado em Fevereiro de 1997, que marcou o término do desenvolvimento *mainstream* do 2.1-STABLE. Agora em manutenção, apenas aperfeiçoamentos de segurança e outras correções críticas são realizadas neste ramo (RELENG_2_1_0).

O ramo 2.2 do FreeBSD foi iniciado a partir da série parcial de desenvolvimento ("`-CURRENT") em Novembro de 1996, foi intitulado ramo RELENG_2_2, e a primeira versão completa (2.2.1) foi lançada em Abril de 1997. Versões posteriores ao longo do ramo 2.2 foram criadas no verão e outono de 1998, sendo a última delas (2.2.8) lançada em Novembro de 1998, marcando o início do fim do ramo 2.2.

A árvore foi ramificada mais uma vez, em 20 de Janeiro de 1999, iniciando os ramos 4.0-CURRENT e 3.X-STABLE. A partir da 3.X-STABLE, a versão 3.1 foi lançada, em 15 de Fevereiro de 1999; a versão 3.2 foi lançada em 15 de Maio de 1999; a 3.3 em 16 de Setembro de 1999; a versão 3.4 em 20 de Dezembro de 1999, e a 3.5 em 24 de Junho de 2000, que foi complementada um pouco depois com uma pequena atualização de segurança, o 3.5.1, que incorporava algumas correções de segurança de última hora para o Kerberos. Esta se tornou a versão final para o ramo 3.X.

Outro ramo foi iniciado em 13 de Março de 2000, de forma emergencial na metade do ramo 4.X-STABLE, considerado agora o "`ramo -stable corrente". Posteriormente houveram várias versões desta série: 4.0-RELEASE foi apresentado ao mundo em Março de 2000, e a versão mais recente, 4.10-RELEASE surgiu em May 2004. Existirão versões adicionais ao longo do ramo 4.X-STABLE (RELENG_4) ainda em 2003.

A versão 5.0-RELEASE, muito aguardada, foi anunciada em 19 de Janeiro de 2003. O resultado culminante de aproximadamente três anos de trabalho, esta versão colocou o FreeBSD no caminho do suporte avançado a multiprocessamento simétrico, suporte avançado a aplicações *multithread* e apresentou ao público suporte às plataformas UltraSPARC® e *ia64*. Esta versão foi seguida pela 5.1 em Junho de 2003. Além de um número muito grande de novas funcionalidades, as versões 5.X do FreeBSD contem ainda uma série de trechos em desenvolvimento em todas as arquiteturas de sistemas relacionadas. Por tal razão, as versões 5.X são consideradas versões de "`Nova Tecnologia", enquanto a série 4.X atua como versões de "`Produção". No momento apropriado, a série 5.X será declarada estável e o trabalho mais recente será destinado ao próximo ramo de desenvolvimento, o 6.0-CURRENT.

Por hora, projetos de desenvolvimento que requerem mais tempo continuam sendo realizados no ramo 5.X-CURRENT, e versões *SNAPSHOT* da série em CDROM (e lógico, na rede) estão continuamente sendo disponibilizados à partir do [servidor de snapshots](#) e apresentado como o trabalho em progresso.

1.3.2. Objetivos do Projeto FreeBSD

Contribuição de Jordan Hubbard.

Os objetivos do Projeto FreeBSD são de oferecer *software* que pode ser utilizado para qualquer propósito, sem restrições impeditivas. Muitos de nós investimos

significativamente no código (e no projeto) e certamente não seria uma surpresa se de vez em quando recebessemos uma pequena compensação financeira, mas definitivamente não insistimos nisso. Acreditamos que nossa primeira e principal "missão" é oferecer código para qualquer usuário, e para qualquer propósito, de forma que o código em questão chegue o mais longe que puder e que traga os maiores benefícios que forem possíveis. Este é, acreditamos, um dos principais fundamentos do projeto e do Software Livre, e um dos quais nós apoiamos de forma entusiasmada.

O código em nossa árvore fonte sob a Licença Pública Geral GNU (GPL) ou Licença Pública Menos Geral GNU (LGPL) contém consideravelmente mais restrições impositivas, contudo, felizmente mais pelo lado de forçar o acesso irrestrito ao código do que o oposto tradicional. Devido às complexidades adicionais que podem envolver o uso comercial de *software* GPL, nós temos preferência pelos programas lançados sob a licença de direito autoral BSD sempre que possível, por ser uma licença consideravelmente mais flexível.

1.3.3. O Modelo de Desenvolvimento FreeBSD

Contribuição de Satoshi Asami.

O desenvolvimento do FreeBSD é um processo muito aberto e flexível, literalmente composto pela contribuição de centenas de pessoas ao redor do mundo, como pode ser visto em nossa [lista de contribuidores](#). A infraestrutura de desenvolvimento do FreeBSD permite que estas centenas de desenvolvedores colaborem mutuamente pela Internet. Estamos constantemente à procura de novos desenvolvedores e novas idéias, e aqueles dispostos a se tornarem mais envolvidos com o projeto precisam apenas nos contatar através da lista [lista de discussões técnicas FreeBSD \(hackers\)](#). A lista [lista de discussão dos anúncios FreeBSD](#) também está disponível aos que querem ficar mais informados sobre as principais áreas de trabalho do projeto FreeBSD.

Informações úteis de se conhecer sobre o projeto FreeBSD e seu processo de desenvolvimento, seja trabalhando de forma independente ou em cooperação mútua:

O repositório CVS

A árvore de fontes central do FreeBSD é mantida pelo [CVS](#) (Sistema de Versões Concorrentes), uma ferramenta de controle de código fonte disponível gratuitamente, e que vem junto com o FreeBSD. O [repositório CVS](#) primário está localizado em Santa Clara, Califórnia, nos EUA, onde ele é replicado em várias máquinas espelho espalhadas ao redor do mundo. A árvore CVS, que contém os ramos [-CURRENT](#) e [-STABLE](#), podem ser completamente replicadas, de forma extremamente fácil, à sua própria máquina local. Refira-se à seção [Sincronizando seu código fonte](#) para obter mais informações sobre isso.

A lista de *committers*

Os *committers* são as pessoas que tem acesso de *escrita* na árvore CVS, e tem autorização para realizar modificações diretas na árvore fonte do FreeBSD (o termo "committer" se origina no comando `commit` do [cvs\(1\)](#), que é utilizado para incluir novas mudanças no repositório CVS). Ou seja, os *committers* são os

desenvolvedores do FreeBSD. A melhor maneira de submeter modificações/contribuições para revisão dos *committers* é utilizar o comando [send-pr\(1\)](#). Se algo estiver fora de conformidades com o sistema de [send-pr\(1\)](#), pode ser interessante entrar em contato com os desenvolvedores enviando uma mensagem de correio para a lista lista de discussão de FreeBSD *committers*.

O Grupo Central (*Core Team*) do FreeBSD

O *FreeBSD core team* seria equivalente ao corpo de diretores, se o Projeto FreeBSD fosse uma empresa. A tarefa primária do *core team* é garantir que o projeto, de forma geral, esteja em boa forma e caminhando nas direções corretas, bem como o recrutamento de novos membros para o *core team* conforme outros vão se desligando do cargo. O *core team* atual foi eleito a partir de um grupo de desenvolvedores que se candidataram ao cargo, em Junho de 2002. Eleições para o *core team* acontecem a cada 2 anos.

Alguns membros do *core team* tem ainda áreas específicas de responsabilidade no projeto, o que significa que eles são responsáveis por garantir que um trecho bastante relevante e grande do sistema funcionem como se espera. Para a lista completa dos desenvolvedores do FreeBSD e suas respectivas áreas de responsabilidade, por gentileza, consulte a [Lista de Contribuidores](#).

Nota: A maioria dos membros do *core team* são voluntários no desenvolvimento do FreeBSD, e não se beneficiam financeiramente do projeto, de forma que "desenvolvimento" não pode ser confundido com "suporte garantido" por parte destes. A analogia com o "corpo de diretores" feita acima não é muito precisa, melhor seria dizermos que estas são as pessoas que desistiram de suas vidas em favor do FreeBSD.

Contribuidores externos

Por último, mas definitivamente não em último, o maior grupo de desenvolvedores são os próprios usuários do sistema, que oferecem retorno e correções de problemas ao projeto de maneira consideravelmente constante. A forma principal de se manter em sintonia com os trechos de desenvolvimento menos centralizados no FreeBSD, é se inscrever na lista [lista de discussões técnicas FreeBSD \(hackers\)](#) onde tais assuntos são discutidos. Veja [Apêndice C](#) para obter mais informações sobre as inúmeras listas de discussão do projeto FreeBSD.

[A Lista de contribuidores do FreeBSD](#) é grande e está em constante crescimento, então por que você não entra nessa lista hoje, contribuindo com alguma coisa para o projeto FreeBSD?

Oferecer código não é a única maneira de contribuir com o projeto; para uma lista mais completa do que precisa ser feito, por gentileza, refira-se ao [sítio WWW do Projeto FreeBSD](#).

Em resumo, nosso modelo de desenvolvimento é um conjunto de círculos concêntricos. O modelo centralizado é projetado com o objetivo de oferecer conveniência aos

usuários do FreeBSD, que são agraciados com um modelo simples para acompanhar uma base centralizada de códigos, e não para afastar contribuidores em potencial ! Nosso desejo é apresentar um sistema operacional estável com um amplo e coerente conjunto de [programas aplicativos](#) que os usuários podem instalar e utilizar de forma fácil. O modelo atual funciona muito bem na realização deste objetivo.

Tudo que pedimos aos que desejam se juntar a nós como desenvolvedores FreeBSD, é um pouco da mesma dedicação que os membros atuais tem, pra prover o contínuo sucesso do projeto!

1.3.4. A Versão Atual do FreeBSD

FreeBSD é um sistema cujos fontes foram inteiramente baseados no 4.4BSD-Lite, disponível livremente para computadores baseados em Intel i386™, i486™, Pentium®, Pentium Pro, Celeron®, Pentium II, Pentium III, Pentium 4 (ou compatível), Xeon™, DEC Alpha™ e Sun UltraSPARC. O sistema é baseado primariamente em programas do CSRG (Grupo de Pesquisa em Sistemas Computacionais) da U.C. Berkeley, com algumas melhorias incorporadas do NetBSD, OpenBSD, 386BSD e da Fundação do Software Livre (FSF).

Desde o lançamento do FreeBSD 2.0 no final de 1994, a performance, as funcionalidades, e a estabilidade do FreeBSD foram dramaticamente incrementados. A maior modificação foi a melhoria do sistema de memória virtual com a inclusão do *merged VM/file buffer cache* que não apenas melhora a performance, mas também reduz pedaços perdidos de memória no FreeBSD, tornando a configuração mínima de 5 MB um requerimento mais aceitável. Outras melhorias incluem suporte completo à cliente e servidor NIS, suporte à *transaction TCP*, discagem PPP por demanda, suporte DHCP integrado, um subsistema SCSI aperfeiçoado, suporte ISDN, suporte ATM, FDDI, adaptadores de rede *Fast Ethernet* e *Gigabit Ethernet* (1000 Mbit), suporte aperfeiçoado às adaptadoras Adaptec mais recentes, e muitos milhares de correções de problemas.

Além da distribuição base, o FreeBSD oferece ainda uma coleção de aplicações (*ports*) composta de milhares de programas populares. No momento da preparação deste documento, haviam mais de 11,800 *ports* ! A lista de *ports* varia de servidores http (WWW) à jogos, linguagens de programação, editores de texto e praticamente tudo o mais que se encaixe entre eles. A coleção de *ports* inteira requer aproximadamente 300 MB de espaço para armazenamento, sendo todos *ports* identificados como ``deltas" à seus fontes originais. Isto torna o processo de atualização dos *ports* consideravelmente mais fácil para nós, e reduz muito a demanda por espaço em disco, especialmente quando comparado à versão 1.0 da *Coleção de Ports*. Para compilar uma aplicação basta entrar no diretório do *port* que você deseja instalar, digitar `make install` e deixar o sistema cuidar do resto. A distribuição original para cada aplicação é obtida dinamicamente do CDROM ou de um sítio FTP, portanto tudo que você precisa é ter espaço em disco o bastante para compilar os programas desejados. Quase todos os programas também estão disponíveis como ``pacotes" pré-compilados, que podem ser instalados com um simples comando (`pkg_add`) por aqueles que preferem não compilar suas próprias aplicações à partir dos fontes. Mais informações sobre pacotes e sobre *ports* podem ser encontradas no [Capítulo 4](#).

Uma série de documentos adicionais que eventualmente você pode jogar muito úteis durante o processo de instalação e utilização do FreeBSD podem ser encontrados também no diretório `/usr/share/doc` em qualquer estação FreeBSD recente. Você pode visualizar os manuais instalados localmente com qualquer navegador HTML, utilizando as seguintes URLs:

O FreeBSD Handbook

</usr/share/doc/handbook/index.html>

As FreeBSD FAQ

</usr/share/doc/faq/index.html>

Você ainda pode acessar as cópias mestre (e frequentemente mais atualizadas) em <http://www.FreeBSD.org/>.

Capítulo 2. Instalando FreeBSD

Índice

- 2.1. [Sinótese](#)
- 2.2. [Tarefas de pré-instalação](#)
- 2.3. [Iniciando a instalação](#)
- 2.4. [Introdução ao Sysinstall](#)
- 2.5. [Alocando espaço em disco](#)
- 2.6. [Escolhendo o que instalar](#)
- 2.7. [Escolhendo a mídia de instalação](#)
- 2.8. [Efetivando a instalação](#)
- 2.9. [Pós-Instalação](#)
- 2.10. [Hardware Suportado](#)
- 2.11. [Resolução de problemas](#)
- 2.12. [Guia avançado de instalação](#)
- 2.13. [Preparando sua própria mídia de instalação](#)

Reestruturado, reorganizado e parcialmente rescrito por Jim Mock. O passeio através do sysinstall, imagens da tela, e cópia geral por Randy Pratt.

2.1. Sinótese

FreeBSD é composto de um programa de fácil utilização, em modo texto, chamado de **sysinstall**. Este é o programa padrão de instalação para o FreeBSD, apesar dos distribuidores terem liberdade para oferecer seu próprio programa de instalação, se desejado. Este capítulo descreve a utilização do **sysinstall** para realizar a instalação do FreeBSD.

Após a leitura deste capítulo, você saberá:

- Como criar os discos de instalação do FreeBSD.
- Como FreeBSD se refere e faz as subdivisões dos discos rígidos.
- Como iniciar o **sysinstall**.
- As perguntas que o **sysinstall** fará durante sua utilização, o que cada uma significa, e como respondê-las.

Antes de ler este capítulo, você é aconselhado:

- Ler a lista de *hardware* suportado que acompanha a versão do FreeBSD que você está instalando, e verificar se os seus equipamentos são suportados.

Nota: Em geral, estas instruções de instalação são escritas para computadores de arquitetura i386™ (``compatíveis PC "). Onde necessário, serão listadas instruções específicas para outras plataformas (por exemplo, Alpha). Embora este guia seja mantido o mais atualizado o possível, poderão ser encontradas algumas pequenas diferenças entre a aplicação instaladora e o que é apresentado neste documento. É aconselhável que este capítulo seja utilizado como um guia geral ao invés de um rigoroso passo a passo de instalação.

2.2. Tarefas de pré-instalação

2.2.1. Inventário de seu computador

Antes de instalar o FreeBSD, é aconselhável tentar levantar todas as informações sobre os componentes presentes em seu computador. As rotinas de instalação do FreeBSD mostrarão estes componentes (discos rígidos, placas de rede, dispositivos de CDROM, e vários outros) com seus respectivos números de modelo e fabricação. O FreeBSD tentará também determinar a configuração correta para estes dispositivos, o que inclui informações sobre IRQ e utilização de endereços de E/S. Devido aos caprichos de alguns equipamentos para PC, esse processo não é sempre realizado com sucesso, e pode ser necessário corrigir as definições do FreeBSD na sua configuração.

Caso tenha um outro sistema operacional instalado, como Windows® ou Linux, é uma boa idéia utilizar os recursos oferecidos por estes sistemas para saber como seu equipamento já está configurado. Caso não esteja certo das configurações de alguma placa de expansão, pode ser possível encontra-las impressas na própria placa. Valores populares para endereços de interrupção (IRQ) são 3, 5 e 7, e o endereçamento de E/S normalmente é definido em números hexadecimais, como 0x330.

Recomendamos que você imprima ou escreva estas informações antes de iniciar o processo de instalação do FreeBSD. A utilização de uma tabela, como abaixo, pode ajudar:

Tabela 2-1. Modelo de Inventário de Dispositivos

Nome do dispositivo	IRQ	Endereço(s) de E/S	Notas
Primeiro disco rígido	N/A	N/A	Disco de 40 GB, marca Seagate, primeira IDE primária
CDROM	N/A	N/A	Primeira IDE secundária
Segundo disco rígido	N/A	N/A	Disco de 20 GB, marca IBM, segunda IDE primária
Primeira controladora IDE	14	0x1f0	
Placa de rede	N/A	N/A	Intel® 10/100
Modem	N/A	N/A	3Com® 56K faxmodem, na COM1
...			

2.2.2. Faça cópias de segurança de seus dados

Caso a estação onde você instalará o FreeBSD contenha dados relevantes, garanta que você tenha cópias de segurança (*backup*) destes dados, e que as cópias de segurança tenham sido testadas antes de iniciar a instalação do FreeBSD. A rotina de instalação do FreeBSD pedirá confirmação antes de escrever quaisquer dados no seu disco, mas uma vez iniciado, este processo não pode ser desfeito.

2.2.3. Decida onde instalar o FreeBSD

Se você deseja que o FreeBSD utilize todo o seu disco rígido, então não há com o que se preocupar neste momento -- você pode pular esta seção.

Contudo, se você precisa que o FreeBSD co-exista com outro sistema operacional, então é necessário uma certa compreensão sobre como os dados são dispostos em disco, e como isso pode afetar você.

2.2.3.1. Disposições de disco no i386™

Um disco de PC pode ser dividido em partes separadas. Tais partes são chamadas de *partições*. A forma como o PC é projetado possibilita que apenas quatro partições sejam suportadas por disco. Estas partições são chamadas de *partições primárias*. Como paliativo para esta limitação foi criado um novo tipo de partições, chamado de *partições extendidas*. Um disco pode conter apenas uma partição extendida. Partições especiais, chamadas de *partições lógicas* podem ser criadas dentro destas partições extendidas.

Cada partição possui um identificador único, chamado de *partition ID*, que é um número utilizado para identificar o tipo de dado naquela partição. Partições FreeBSD tem o ID de identificação número 165 .

De forma geral, cada sistema operacional utilizado identificará as partições de forma particular. Por exemplo, DOS e seus descendentes como Windows atribuem a cada partição primária e lógica uma *letra de dispositivo*, começando a partir do `C:` .

FreeBSD deve obrigatoriamente ser instalado em uma partição primária. FreeBSD pode manter todos seus dados, incluindo qualquer arquivo que você crie nesta única partição. Contudo, se você possuir múltiplos discos, você pode criar uma partição FreeBSD em cada um deles, ou apenas em alguns. Quando instalar o FreeBSD, ao menos uma partição deve estar disponível. Pode ser uma partição vazia, que você preparou, ou então pode ser uma partição já existente contendo dados que não são mais necessários.

Caso esteja utilizando todas as partições primárias de todos seus discos, então será necessário liberar uma delas para o FreeBSD, utilizando as ferramentas pelo sistema operacional que você utiliza (por exemplo, `fdisk` no DOS ou Windows).

Se você tem uma partição sobressalente, pode utiliza-la. Contudo, pode ser necessário modificar primeiro uma ou mais das suas partições atuais.

A instalação mínima do FreeBSD ocupa apenas 100 MB de espaço em disco. Contudo, esta é uma instalação *extremamente* mínima, não deixando quase espaço algum para seus próprios arquivos. Um valor mínimo mais realista é 250 MB sem ambiente gráfico, ou 350 MB (ou até mais) se deseja o ambiente gráfico. Se você planeja instalar muitas aplicações de terceiros também, então será necessário mais espaço.

Você pode utilizar alguma ferramenta comercial como **PartitionMagic**® para redimensionar suas partições de forma a criar algum espaço para o FreeBSD. O diretório `tools` no CDROM contém dois programas gratuitos que podem realizar esta tarefa, chamados de **FIPS** e **PResizer**. **FIPS**, **PResizer**, e **PartitionMagic** pode redimensionar partições FAT16 e FAT32 -- usadas do MS-DOS® ao Windows ME. **PartitionMagic** é a única ferramenta capaz de redimensionar sistema de arquivos NTFS. Documentações sobre ambas aplicações estão disponíveis no mesmo diretório.

Atenção: O uso incorreto destas ferramentas podem apagar os dados em seu disco. Garanta que existam cópias de seguranças recentes e funcionais antes de utiliza-las.

Exemplo 2-1. Utilizando como está, uma partição já existente

Suponha que você tenha um computador com um único disco rígido de 4 GB que já tem uma versão do Windows instalada, e você separou separou o disco em dois dispositivos, o dispositivo `C:` e o `D:` cada qual com 2 GB de tamanho. Você dispõe de 1GB de dados no `C:` e 0.5 GB de dados no `D:` .

Isto significa que seu disco tem duas partições, uma por letra de dispositivo. Você pode copiar todos os dados atuais da `D:` na `C:` o que liberará a segunda partição, que estará pronta para o FreeBSD.

Exemplo 2-2. Redimensionando uma partição existente

Suponha que você tem um computador com um único disco rígido de 4 GB que já tem uma versão do Windows instalada. Ao instalar o Windows você criou uma única partição, atribuindo ao seu dispositivo c: o tamanho de 4 G. Atualmente tem 1.5GB de espaço em uso e você deseja alocar 2 GB de espaço para o FreeBSD.

Para instalar o FreeBSD será preciso:

1. Fazer cópia de segurança dos dados de seu Windows e reinstalar o Windows definindo uma partição de 2 GB de espaço no momento da instalação,
2. Utilizar uma das ferramentas como o **PartitionMagic**, descrito anteriormente para redimensionar sua partição Windows.

2.2.3.2. Disposições de Disco no Alpha

Será necessário um disco dedicado para o FreeBSD em um Alpha. Não é possível compartilhar o disco com outro sistema operacional hoje. Dependendo da máquina Alpha específica que você tem em mãos, o disco pode ser SCSI ou IDE, com tanto que sua máquina seja capaz de inicializar por meio deste.

Seguindo as convenções do manual da Digital / Compaq todas as entradas referentes ao SRM serão apresentadas em letras maiúsculas. O SRM não faz distinção entre maiúsculas e minúsculas.

Para encontrar os nomes e tipos de discos disponíveis na máquina, use o comando *SHOW DEVICE* na tela do console SRM:

```
>>>SHOW DEVICE
dka0.0.0.4.0          DKA0          TOSHIBA CD-ROM XM-57 3476
dkc0.0.0.1009.0      DKC0          RZ1BB-BS    0658
dkc100.1.0.1009.0    DKC100       SEAGATE ST34501W  0015
dva0.0.0.0.1         DVA0
ewa0.0.0.3.0         EWA0          00-00-F8-75-6D-01
pkc0.7.0.1009.0     PKC0          SCSI Bus ID 7    5.27
pqa0.0.0.4.0         PQA0          PCI EIDE
pqb0.0.1.4.0         PQB0          PCI EIDE
```

Este exemplo é de uma *Digital Personal Workstation* modelo 433au e apresenta três discos ligados na estação. O primeiro é um CDROM chamado de DKA0 e os outros dois são discos rígidos, e são chamados DKC0 e DKC100 respectivamente.

Discos cujos nomes estão dispostos no formato DKx são discos SCSI. Por exemplo, DKA100 se refere ao disco SCSI com ID SCSI número 1 no primeiro barramento SCSI (A), enquanto DKC300 se refere a um disco SCSI com ID SCSI número 3 no terceiro barramento SCSI (C). O dispositivo PKx se refere à controladora SCSI. Como foi visto na saída do comando *SHOW DEVICE*, dispositivos de CDROM SCSI são tratados como disco rígido SCSI qualquer.

Discos IDE tem nomes similares a DQx, enquanto PQx é a controladora IDE associada.

2.2.4. Obtenha detalhes de configuração da interface de rede

Caso você pretenda se conectar em rede como parte do processo de instalação de seu FreeBSD (por exemplo, caso a realização seja feita a partir de um sítio FTP ou servidor NFS), então é necessário conhecer a configuração de sua rede. Você será indagado por estas informações durante a instalação do FreeBSD, de forma que o sistema possa se conectar à rede e concluir a instalação.

2.2.4.1. Conectando-se a uma rede *Ethernet* ou um modem DSL ou cabo.

Se você se conecta à uma rede Ethernet, ou tem uma conexão Internet com uma placa de rede via cabo ou DSL, então será necessário saber as seguintes informações:

1. Endereço IP
2. Endereço IP do *gateway* padrão
3. Nome da estação (*hostname*)
4. Endereço IP do servidor DNS
5. Máscara de subrede

Caso não saiba tais informações, pergunte ao seu administrador de sistemas ou provedor de serviços. Eles podem dizer que tais informações são atribuídas automaticamente, usando *DHCP*. Se for o caso, anote isto.

2.2.4.2. Conectando-se por meio de um modem

Se você disca para um provedor de serviços Internet (ISP) utilizando um modem convencional, ainda é possível instalar o FreeBSD na internet, contudo isso demandará algum tempo a mais.

Será necessário saber:

1. O número de telefone do seu ISP
2. A porta COM onde seu modem está ligado
3. Usuário e senha da conta no seu provedor

2.2.5. Verifique a Errata do FreeBSD

Mesmo considerando que o projeto FreeBSD se esmera para garantir que cada versão do FreeBSD seja tão estável quanto for possível, falhas ocasionalmente podem abalar este processo. Em situações extremamente raras tais falhas podem afetar o processo de instalação do FreeBSD. Conforme estes problemas são descobertos e corrigidos, eles são anotados na [Errata do FreeBSD](#), que pode ser encontrada no sítio WWW do FreeBSD. Você deve verificar esta errata antes de iniciar a instalação para saber se existem problemas com os quais você deve ficar atento.

Informações sobre todas as versões, incluindo a errata para cada uma, podem ser encontradas na seção de [informação de versões](#) do [sítio WWW do FreeBSD](#).

2.2.6. Obtenha os arquivos de instalação do FreeBSD

O processo de instalação do FreeBSD pode instalar o sistema à partir de arquivos armazenados nos seguintes locais:

Mídia Local

- Um CDROM ou DVD
- Uma partição DOS no mesmo computador
- Fita SCSI ou QIC
- Disquetes

Rede

- Sítio FTP atrás de um firewall, ou usando um *proxy* HTTP, se necessário
- Servidor NFS
- Conexão paralela ou serial dedicada

Caso tenha adquirido FreeBSD em CD ou DVD, então você já tem tudo que precisa, e pode pular para a próxima seção ([Preparando a mídia de inicialização \(Boot\)](#)).

Caso não tenha obtido os arquivos de instalação do FreeBSD, você pode se adiantar para [Seção 2.13](#) que explica como preparar o FreeBSD para a instalação por qualquer das mídias acima citadas. Após a leitura desta seção, volte a este ponto e leia [Seção 2.2.7](#).

2.2.7. Preparando a mídia de inicialização (Boot)

O processo de instalação do FreeBSD começa ao iniciar o computador (dar *boot*) com o instalador do sistema (não é um programa para ser executado a partir de outro sistema operacional). O computador normalmente é inicializado a partir do sistema operacional que já está instalado em seu disco, mas também é possível configurar o sistema de modo a fazer uso de um disquete "inicializável". Os computadores mais recentes podem ser inicializados à partir de um CD do FreeBSD no dispositivo de de CDROM.

Dica: Caso tenha o FreeBSD em CDROM ou DVD (indiferentemente se você comprou o disco ou o fez manualmente), e seu computador permite a inicialização por estes dispositivos (normalmente a opção no BIOS chamada de "Boot Order" ou similar), então você pode pular esta seção. As imagens de CDROM e DVD do FreeBSD são inicializáveis ("bootáveis") e podem ser utilizadas para instalar o FreeBSD sem qualquer outro procedimento especial.

Para criar disquetes de inicialização, siga os seguintes passos:

1. Adquirir as imagens de disquetes de inicialização

Os discos de inicialização estão disponíveis a partir de sua mídia de instalação, no diretório `floppies/` e podem também ser baixadas à partir do [diretório de](#)

[imagens de disquetes](#) para a arquitetura i386™ e a partir do [diretório de imagens de disquetes](#) para a arquitetura Alpha.

As imagens de disquete tem a extensão `.flp`. O diretório `floppies/` tem uma série de imagens distintas, e as que você vai precisar usar dependem da versão do FreeBSD que você pretende instalar, e em alguns casos, dos equipamentos onde você irá realizar a instalação. Na maioria dos casos serão necessárias apenas duas imagens, `kern.flp` e `mfsroot.flp`. Alguns dispositivos de controle adicionais poderão ser necessários em alguns sistemas. Estes dispositivos estão disponíveis na imagem `drivers.flp`. Verifique o arquivo `README.TXT` no mesmo diretório para obter as informações mais atualizadas sobre estas imagens de disquetes.

Importante: O programa FTP que você for utilizar deve estar ativo em *modo binário* no momento de baixar as imagens de disquete da Internet. Alguns navegadores WWW conhecidamente utilizam modo texto *text* (ou *ASCII*), o que se tornará claro quando perceber que não pode inicializar o sistema por meio destes discos.

2. Preparando os disquetes

É necessário preparar um disquete por imagem que você baixou da Internet. É imperativo que tais discos estejam livres de defeitos. A melhor maneira de tirar isso a limpo é formatando os discos. Não confie em disquetes pré-formatados. O utilitário de formatação do Windows não informará sobre a presença de blocos defeituosos no disquete, esta ferramenta apenas os marca como ```bad"` (defeituoso) e os ignora, portanto não confie nela também. É aconselhado utilizar disquetes novos caso esta seja sua opção de instalação.

Importante: Ao instalar o FreeBSD, se o programa de instalação travar, congelar ou falhar, uma das primeiras coisas a se suspeitar são os disquetes. Grave as imagens em um novo disquete e inicie o processo de instalação novamente.

3. Gravando as imagens nos disquetes

Os arquivos `.flp` *não* são arquivos comuns os quais basta copiá-los para o disco. São imagens do conteúdo completo do disco. Isto significa que você *não pode* apenas copiar os arquivos de um disco para outro. Ao invés disso é necessário utilizar ferramentas específicas para escrever as imagens diretamente no disco.

Caso esteja criando os disquetes a partir de um computador que roda MS-DOS/Windows, disponibilizamos uma ferramenta que faz este serviço, chamada de `fdimage`.

Caso esteja utilizando as imagens de disquetes disponíveis no CDROM, e seu dispositivo de CDROM tem a letra `E:`, basta executar este comando:

```
E:\> tools\fdimage floppies\kern.flp A:
```

Repita este comando para cada imagem `.flp` trocando o disquete a cada vez, e garantindo que cada um esteja sendo rotulado com o nome da imagem que acabou de ser utilizada. Ajuste a linha de comando conforme necessário, dependendo onde as imagens `.flp` estão localizadas. Caso você não disponha do CDROM, o programa `fdimage` pode ser baixado da Internet à partir do [diretório tools](#) no sítio FTP do FreeBSD.

Caso esteja gravando os disquetes a partir de um sistema UNIX® (como um outro sistema FreeBSD por exemplo) pode utilizar o comando [dd\(1\)](#) para gravar a imagem. No FreeBSD bastaria executar:

```
# dd if=kern.flp of=/dev/fd0
```

No FreeBSD, `/dev/fd0` se refere ao primeiro dispositivo de disquete (o A:). `/dev/fd1` seria o dispositivo B: e assim sucessivamente. Outras variantes UNIX podem ter nomes distintos para os dispositivos de disquetes, e será necessário verificar a documentação do sistema conforme for necessário.

Agora você está pronto para iniciar a instalação do FreeBSD.

2.3. Iniciando a instalação

Importante: Por padrão, o programa de instalação não fará qualquer modificação em seus discos até que você veja a seguinte mensagem:

```
Last Chance: Are you SURE you want continue the installation?
```

```
If you're running this on a disk with data you wish to save then WE  
STRONGLY ENCOURAGE YOU TO MAKE PROPER BACKUPS before proceeding!
```

```
We can take no responsibility for lost disk contents!
```

A instalação pode ser interrompida a qualquer instante, antes desta advertência final, sem modificar qualquer conteúdo em seu disco rígido. Se você estiver em dúvida sobre ter configurado algo incorretamente, pode simplesmente desligar o computador antes deste ponto, e dano algum será causado.

2.3.1. Booting

2.3.1.1. Processo de inicialização no i386™

1. Inicie com seu computador desligado.
2. Ligue o computador. Durante sua inicialização ele mostrará uma opção para entrar no menu de configuração do sistema, ou o BIOS, normalmente conseguido ao utilizar teclas como **F2**, **F10**, **Del**, ou **Alt+S**. Utilize o conjunto de teclas que for indicado na tela. Em alguns casos o computador poderá exibir uma tela gráfica ao iniciar. Normalmente basta pressionar **Esc** para eliminar a imagem gráfica e permitir que você visualize as mensagens necessárias.

3. Encontre as configurações que controlam o dispositivo de inicialização do sistema. Normalmente esta opção é intitulada "Boot Order" e costuma ser apresentada como uma lista de dispositivos como *Floppy*, *CDROM*, *First Hard Disk*, e assim por diante.

Caso seja necessário preparar disquetes de inicialização, garanta que a opção de disquetes seja selecionado. Caso esteja iniciando o sistema pelo CDROM garanta então que este seja o dispositivo selecionado. Em caso de dúvida consulte o manual do seu computador e/ou da placa-mãe.

Faça a modificação, salve e saia do BIOS. O computador deverá se reiniciar.

4. Caso tenha necessidade de preparar disquetes de inicialização como foi descrito em [Seção 2.2.7](#), então um deles será o primeiro disco de inicialização; na verdade o disco contendo a imagem `kern.flp`. Insira este disco no seu dispositivo de disquetes.

Caso esteja iniciando o sistema a partir do CDROM, será necessário ligar o computador e inserir o CDROM na primeira oportunidade.

Caso o computador inicie normalmente e carregue o sistema operacional já instalado, então alguma destas situações estão ocorrendo:

1. Os discos não foram inseridos à tempo durante o processo de inicialização. Mantenha-os inseridos e reinicie seu computador.
 2. As modificações anteriormente realizadas no BIOS não funcionaram corretamente. Refaça este passo até que a opção correta seja encontrada.
 3. Seu BIOS em particular não suporte inicialização do sistema à partir da mídia desejada.
5. O FreeBSD começará a ser iniciado. Caso esteja iniciando o sistema à partir do CDROM você verá algumas mensagens similares a estas (as informações sobre a versão do sistema foram omitidas):

```
6. Verifying DMI Pool Data .....
7. Boot from ATAPI CD-ROM :
8. 1.  FD 2.88MB System Type-(00)
9. Uncompressing ... done
10.
11. BTX loader 1.00 BTX version is 1.01
12. Console: internal video/keyboard
13. BIOS drive A: is disk0
14. BIOS drive B: is disk1
15. BIOS drive C: is disk2
16. BIOS drive C: is disk3
17. BIOS 639kB/261120kB available memory
18.
19. FreeBSD/i386 bootstrap loader, Revision 0.8
20.
21. /kernel text=0x277391 data=0x3268c+0x332a8 |
22.
23. |
24. Hit [Enter] to boot immediately, or any other key for command
    prompt.
25. Booting [kernel] in 9 seconds... _
```

Caso esteja iniciando à partir de disquetes, você verá algumas mensagens similares a estas (as informações sobre a versão do sistema foram omitidas):

```
Verifying DMI Pool Data .....  
  
BTX loader 1.00 BTX version is 1.01  
Console: internal video/keyboard  
BIOS drive A: is disk0  
BIOS drive C: is disk1  
BIOS 639kB/261120kB available memory  
  
FreeBSD/i386 bootstrap loader, Revision 0.8  
  
/kernel text=0x277391 data=0x3268c+0x332a8 |  
  
Please insert MFS root floppy and press enter:
```

Siga as instruções acima, removendo o disco com a imagem `kern.flp` e inserindo o disco com a imagem `mfsroot.flp`, e depois aperte **Enter**.

26. Indiferente se o sistema foi iniciado por disquete ou CDROM, o processo de *boot* chegará a este ponto:
27. Hit [Enter] to boot immediately, or any other key for command prompt.
28. Booting [kernel] in 9 seconds... _

Espere dez segundos ou aperte **Enter**. Isto iniciará o menu de configuração do kernel.

2.3.1.2. Processo de inicialização no Alpha

1. Comece com seu sistema desligado.
2. Ligue o computador e espere pela tela do monitor de inicialização.
3. Caso tenha sido preciso fazer discos de inicialização, como descrito em [Seção 2.2.7](#) então um dos discos será o primeiro disco de inicialização; na verdade o disco gerado a partir da imagem do `kern.flp`. Insira este disco no seu dispositivo de disquetes e digite o seguinte comando para carregá-lo (substituindo o nome do seu dispositivo de disco caso necessário):
4. >>>BOOT DVA0 -FLAGS '' -FILE ''

Caso esteja iniciando à partir do CDROM, insira o CDROM no dispositivo e digite o seguinte comando para iniciar a instalação (substituindo o nome apropriado para o dispositivo de CDROM, se necessário):

```
>>>BOOT DKA0 -FLAGS '' -FILE ''
```

5. O FreeBSD vai começar a ser iniciado. Caso esteja iniciando o sistema à partir de um disquete, em dado momento será apresentada a mensagem:
6. Please insert MFS root floppy and press enter:

Siga as instruções acima, removendo o disco com a imagem do `kern.flp` e inserindo o disco com a imagem do `mfsroot.flp` e aperte **Enter**.

7. Indiferente se o sistema foi iniciado por disquete ou CDROM, o processo de *boot* chegará a este ponto:
8. Hit [Enter] to boot immediately, or any other key for command prompt.
9. Booting [kernel] in 9 seconds... _

Espreze dez segundos ou aperte **Enter**. Isto iniciará o menu de configuração do kernel.

2.3.2. Configuração do Kernel

Nota: À partir da série 5 do FreeBSD, o menu de configuração do kernel foi aposentado, em favor do novo método [device.hints\(5\)](#). Para mais informações sobre o [device.hints\(5\)](#) por gentileza visite [Seção 7.5](#)

O *kernel* é o núcleo do sistema operacional. Ele é o responsável por inúmeras funções fundamentais, incluindo acesso à todos os dispositivos de controle que possam existir em seu sistema, como disco rígido, placas de rede, placas de som e outros. Todo equipamento suportado pelo kernel do FreeBSD tem um dispositivo associado à ele. Cada dispositivo tem um nome de duas ou três letras, como *sa* para o dispositivo de acesso sequencial SCSI, ou *sio* para o dispositivo de E/S serial (que gerencia portas COM).

Quando o kernel é carregado, cada dispositivo varre seu sistema para tentar identificar se o equipamento que ele controle existe na máquina. Caso exista, o dispositivo configura o equipamento e o torna disponível para o kernel.

Esta verificação normalmente é chamada de *device probing* (identificação de dispositivos). Infelizmente nem sempre é possível realiza-la de forma segura. Alguns dispositivos de controle não coexistem de forma cordial e a identificação de um equipamento pode levar um outro à um estado inconsistente de disponibilidade. Esta é uma limitação básica da concepção do PC.

Muitos dispositivos mais antigos são chamados de dispositivos ISA, que operam de forma diferente dos dispositivos PCI, mais atuais. A especificação ISA obriga que algumas informações sejam afixadas à placa, tipicamente linha de pedidos de interrupção (IRQ) e endereço de E/S que o equipamento utiliza. Estas informações normalmente podem ser configuradas utilizando *jumpers* físicos na placa, ou utilitários baseados em DOS para este fim específico.

Isso sempre foi origem de problemas, porque não é possível ter dois dispositivos ISA que compartilhem a mesma IRQ ou endereço de memória.

Os equipamentos mais novos seguem as especificações PCI, que não necessitam destas informações afixadas, porque o equipamento supostamente deve cooperar com o BIOS que indica quais recursos o dispositivo deve utilizar.

Caso tenha algum equipamento ISA no seu computador, o dispositivo de controle do FreeBSD para este dispositivo terá que ser configurado com a IRQ e o endereço de E/S

de acordo com o definido no equipamento. É por isso que fazer um inventário de seus equipamentos (veja [Seção 2.2.1](#)) pode se tornar muito útil.

Infelizmente, as IRQs e endereços de memória padrão utilizados por alguns dispositivos costumam entrar em conflito. Isso se deve ao fato de alguns equipamentos ISA serem vendidos com configurações conflitantes por padrão. O padrão no dispositivo de controle do kernel do FreeBSD são deliberadamente definidos como espelho dos padrões do fabricante, de forma que o maior número possível de dispositivos funcionarão sem necessitar modificações.

Normalmente isso não é um problema para o uso diário do FreeBSD. Normalmente seu computador não terá dois dispositivos que conflitam, pois se houver algum deles não irá funcionar (indiferente do sistema operacional que você estiver usando).

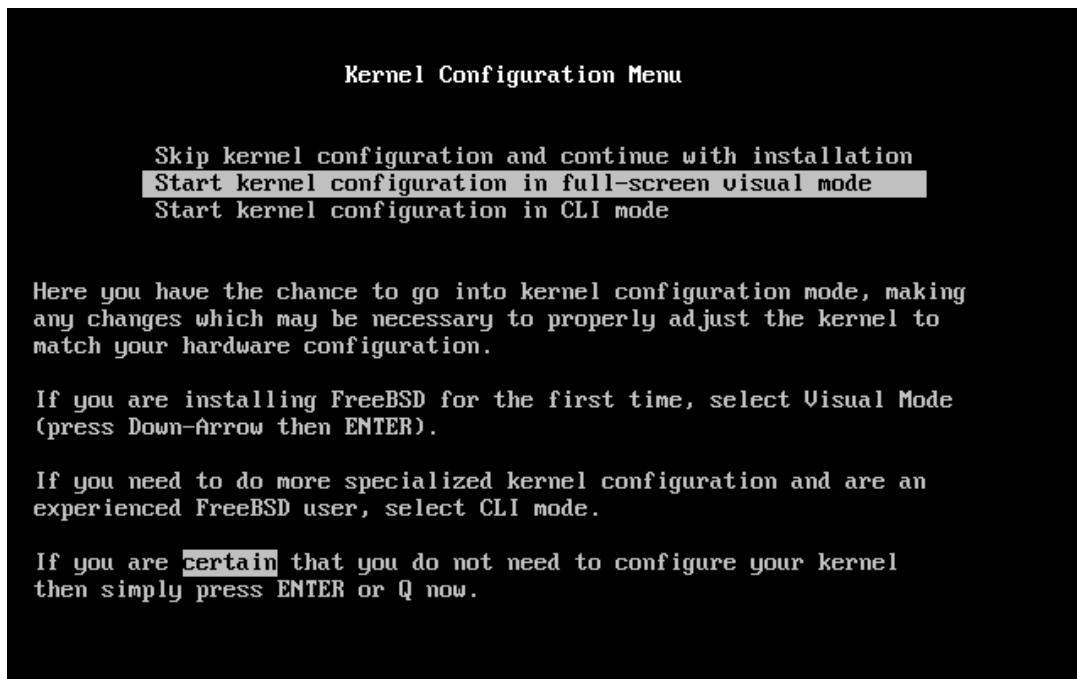
Normalmente o problema ocorre ao instalar o FreeBSD pela primeira vez, porque o kernel utilizado durante a instalação contém o maior número possível de dispositivos, de forma que a instalação possa suportar um grande número de equipamentos distintos. Isso significa que alguns destes equipamentos podem ter configurações conflitantes. Os equipamentos são reconhecidos em uma ordem estrita, e se existir um dispositivo que for reconhecido depois, então pode acontecer do mesmo não funcionar corretamente durante a instalação do FreeBSD.

Por tal motivo, a primeira coisa que você tem a possibilidade de fazer ao instalar o FreeBSD é analisar a lista de dispositivos suportados pelo kernel, desabilitar ou habilitá-los, caso você não tenha equipamentos compatíveis com aquele suporte, ou confirmar e modificar a configuração de recursos do mesmo, caso o padrão esteja incorreto.

Na verdade isso parece mais complicado do que na realidade é.

[Figura 2-1](#) apresenta o primeiro menu de configuração do kernel. Recomendamos que sua escolha seja a opção de modo visual, a *Start kernel configuration in full-screen visual mode*, pois apresenta a interface mais fácil para novos usuários.

Figura 2-1. Menu de Configuração do Kernel



A tela de configuração do kernel ([Figura 2-2](#)) é dividida em quatro seções:

1. Uma lista agrupada de todos os dispositivos que estão atualmente definidos como "ativos" (*active*, subdivididos em grupos como *Storage* e *Network*. Cada dispositivo é apresentado com uma descrição, o nome do dispositivo e os endereços de IRQ e memória utilizados pelo mesmo. Ainda, se um dispositivo ativo conflitar com outro também ativo, a expressão *CONF* é apresentada próxima ao nome do dispositivo. A seção também mostra o número total de equipamentos conflitantes cujo suporte está marcado como ativo.
2. Dispositivos marcados inativo permanecem no kernel, mas não tentarão reconhecer os equipamentos em questão quando o kernel carregar. Eles são divididos em grupos com o mesmo nome disposto na lista de dispositivos ativos.
3. Mais informações sobre o dispositivo selecionado, incluindo IRQ e endereço de memória e E/S.
4. Informações sobre as teclas de comando válidas nesta tela.

Figura 2-2. A interface visual do dispositivo de configuração do kernel

```
-----Active Drivers-----7 conflicts-----Dev---IRQ---Port---
Storage : (Collapsed)
Network : (Collapsed)
Communications : (Collapsed)
Input : (Collapsed)
Multimedia :
Miscellaneous : (Collapsed)

-----Inactive Drivers-----Dev-----
Storage :
Network :
Communications : (Collapsed)
Input :
Multimedia :
Miscellaneous : (Collapsed)

-----

[Enter] Expand device list [X] Expand all lists
[TAB] Change fields [Q] Save and Exit [?] Help
```

Não se preocupe se houverem conflitos listados, isso é esperado uma vez que todos dispositivos estão habilitados, e como já foi explicado, alguns terão as mesmas configurações de recursos que outros.

Agora você deve utilizar a lista de dispositivos afim de resolver os conflitos pendentes.

Resolvendo conflitos de dispositivos

1. Aperte **X**. Este comando irá expandir completamente a lista de dispositivos de forma que você possa visualiza-los. Será preciso usar as teclas de setinhas para pagnar através da lista de dispositivos ativos.

[Figura 2-3](#) apresenta o resultado de ter pressionado a tecla **X**.

Figura 2-3. Lista de dispositivos expandida

Active Drivers	7 conflicts	Dev	IRQ	Port
Storage :				
AdvanSys SCSI narrow controller		adv0		
Adaptec 154x SCSI controller		aha0		
Adaptec 152x SCSI and compatible sound cards		aic0		
ATA/ATAPI compatible disk controller		ata0	14	0x1f0
ATA/ATAPI compatible disk controller		ata1	15	0x170
Buslogic SCSI controller		bt0		
Floppy disk controller		fdc0	6	0x3f0
Inactive Drivers				
Storage :				
Network :				
Communications :	(Collapsed)			
Input :				
Multimedia :				
Miscellaneous :	(Collapsed)			

[Enter] Collapse device list	[C] Collapse all lists			
[TAB] Change fields	[Q] Save and Exit			[?] Help

- Desabilite todos os dispositivos de controle para equipamentos que você não possui. Para desabilitar um dispositivo, selecione-o (ficará clareado) com as teclas de setinhas e aperte **Del**. O dispositivo será movido para a lista de inativos, *Inactive Drivers*.

Caso inadvertidamente você remova o suporte de um dispositivo que é necessário, aperte **Tab** para alternar para o menu de dispositivos inativos, *Inactive Drivers*, selecione o controlador desabilitado e pressione **Enter** para coloca-lo de volta na lista de controladores ativos.

Atenção Não desabilite `sc0`. Ele controla o suporte à unidade de saída padrão, o monitor, e ele é completamente necessário, a não ser que você esteja instalando o sistema por meio de um cabo serial.

Atenção Apenas desabilite `atkbd0` se estiver usando um teclado USB. Se tiver teclado tradicional então mantenha `atkbd0`, o dispositivo de entrada padrão.

- Se não existirem conflitos listados, você pode pular o próximo passo. Do contrário os conflitos restantes devem ser examinados. Se eles não tem indicação de um conflito permitido - *allowed conflict* - na área de mensagem, o endereço de IRQ ou de memória deverão ser modificados, *ou* a IRQ e endereço de *hardware* deverão ser trocados.

Para modificar a configuração do dispositivo pela IRQ e endereço de E/S do equipamento, selecione o ítem em questão e aperte **Enter**. O cursor irá para a

terceira seção da tela e você pode modificar estes valores. Deverá informar as informações de recursos conseguidas quando o inventarário dos equipamentos foi realizado. Pressione a tecla **Q** para terminar a edição da configuração do dispositivo e voltar para a lista de itens ativos.

Se você não tem a menor idéia de que valores devem ser definidos, experimente usar **-I**. Alguns dispositivos de controle do FreeBSD podem, de forma segura, varrer os equipamentos em busca das configurações corretas de cada dispositivo, e a opção **-I** indica que ele deve fazê-lo.

O procedimento para modificar o endereço de acesso à memória varia de dispositivo para dispositivo. Para alguns equipamentos pode ser necessário remover fisicamente a placa e configurar os *jumpers* ou os alternadores DIP (DIP switches). Outras placas oferecem um disquete DOS que contem programas para reconfigura-la. Em todos os casos você deve se referir à documentação que acompanha o equipamento. Obviamente isto implicará na reinicialização do seu computador, de forma que será preciso voltar à rotina de instalação do FreeBSD depois que esta placa tenha sido reconfigurada.

4. Quando todos conflitos estiverem sanados, a tela estará similar a [Figura 2-4](#).

Figura 2-4. Configuração de dispositivos sem conflitos

Active Drivers		Dev	IRQ	Port
Storage :				
ATA/ATAPI compatible disk controller		ata0	14	0x1f0
ATA/ATAPI compatible disk controller		ata1	15	0x170
Floppy disk controller		fdc0	6	0x3f0
Network :				
NE1000,NE2000,3C503,WD/SMC80xx Ethernet adapters		ed0	10	0x280
Communications :				
Parallel Port chipset		ppc0	7	
Inactive Drivers		Dev		
Storage :				(Collapsed)
Network :				(Collapsed)
Communications :				(Collapsed)
Input :				
Multimedia :				
Miscellaneous :				(Collapsed)

Port address :		0x1f0		
IRQ number :		14		
Flags :		0x0		

[Enter] Edit device parameters		[DEL] Disable device		
[TAB] Change fields		[Q] Save and Exit		[?] Help

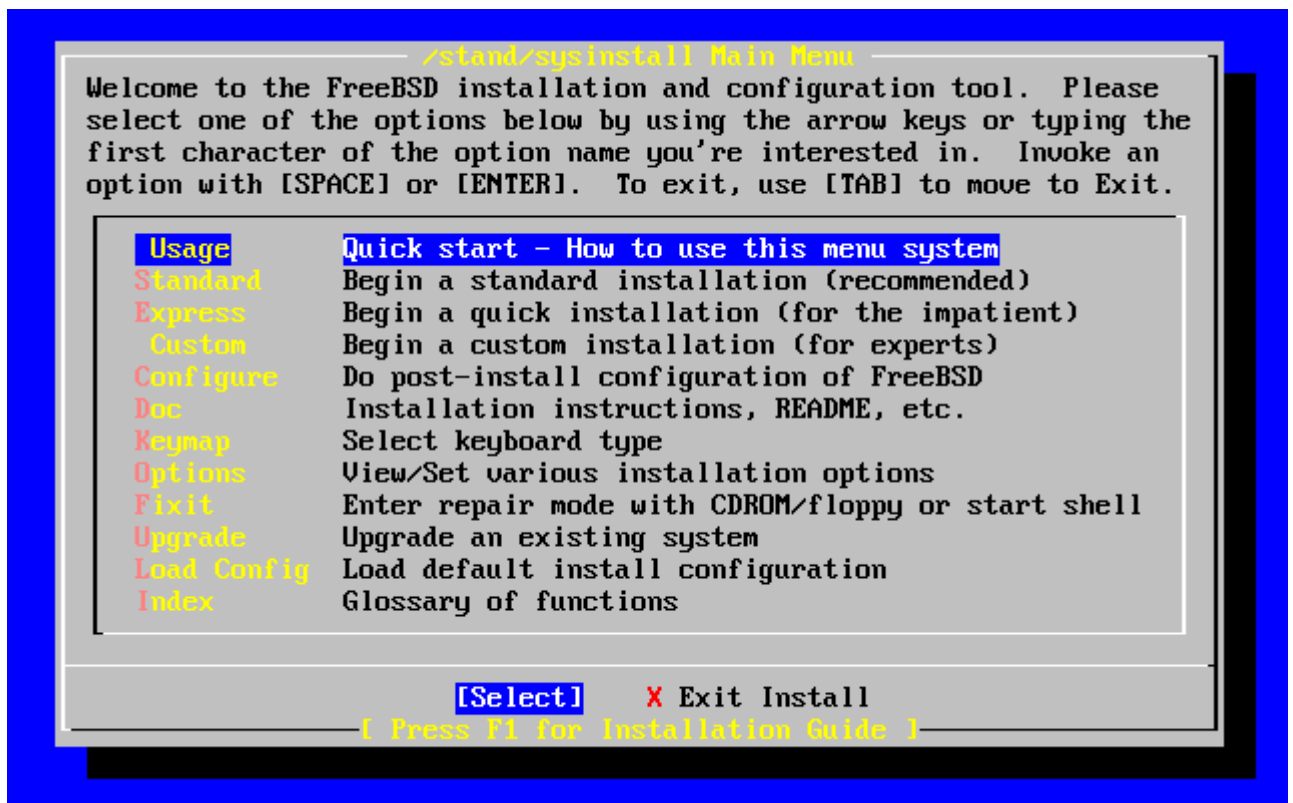
Como você pode ter reparado, a lista de dispositivos ativos agora está bem menor, apenas com os itens para os equipamentos que realmente existem estão listados.

Agora é possível salvar tais modificações e ir para o próximo passo do processo de instalação. Aperte **Q** para sair da interface de configuração de dispositivos. A seguinte mensagem surgirá:

```
Save these parameters before exiting? ([Y]es/[N]o/[C]ancel)
```

Responda **Y** (SIM) para salvar os parâmetros em memória (eles serão salvos em disco quando a instalação for concluída) e o reconhecimento dos equipamentos será iniciado. Após apresentar os resultados do reconhecimento em mensagens de texto preto e branco, o **sysinstall** irá iniciar e apresentará seu menu principal (Figura 2-5).

Figura 2-5. Menu principal do Sysinstall



2.3.3. Revisando o resultado do reconhecimento de dispositivos

As últimas centenas de linhas que foram apresentadas na tela são armazenadas e podem ser revisadas.

Para revisar o *buffer*, aperte **Scroll Lock**. Esta tecla ativa o paginamento da tela. Você pode utilizar as teclas de setinhas ou **PageUp** e **PageDown** para visualizar os resultados do processo de reconhecimento de dispositivos. Aperte **Scroll Lock** novamente para desativar o paginamento.

Faça isso agora para revisar o texto que foi paginado para cima da tela quando o kernel estava sendo carregado e os dispositivos de controle sendo reconhecidos. Você verá

informações textuais similares à [Figura 2-6](#), embora o conteúdo preciso será diferente dependendo de quais dispositivos existirem no seu computador.

Figura 2-6. Resultados típicos do reconhecimento de dispositivos

```
avail memory = 253050880 (247120K bytes)
Preloaded elf kernel "kernel" at 0xc0817000.
Preloaded mfs_root "/mfsroot" at 0xc0817084.
md0: Preloaded image </mfsroot> 4423680 bytes at 0xc03ddcd4

mdl: Malloc disk
Using $PIR table, 4 entries at 0xc00fde60
npx0: <math processor> on motherboard
npx0: INT 16 interface
pcib0: <Host to PCI bridge> on motherboard
pci0: <PCI bus> on pcib0
pcib1:<VIA 82C598MVP (Apollo MVP3) PCI-PCI (AGP) bridge> at device 1.0
on pci0
pcil: <PCI bus> on pcib1
pcil: <Matrox MGA G200 AGP graphics accelerator> at 0.0 irq 11
isab0: <VIA 82C586 PCI-ISA bridge> at device 7.0 on pci0
isa0: <iSA bus> on isab0
atapci0: <VIA 82C586 ATA33 controller> port 0xe000-0xe00f at device
7.1 on pci0
ata0: at 0x1f0 irq 14 on atapci0
ata1: at 0x170 irq 15 on atapci0
uhci0 <VIA 83C572 USB controller> port 0xe400-0xe41f irq 10 at device
7.2 on pci
0
usb0: <VIA 83572 USB controller> on uhci0
usb0: USB revision 1.0
uhub0: VIA UHCI root hub, class 9/0, rev 1.00/1.00, addr1
uhub0: 2 ports with 2 removable, self powered
pci0: <unknown card> (vendor=0x1106, dev=0x3040) at 7.3
dc0: <ADMtek AN985 10/100BaseTX> port 0xe800-0xe8ff mem 0xdb000000-
0xeb0003ff ir
q 11 at device 8.0 on pci0
dc0: Ethernet address: 00:04:5a:74:6b:b5
miibus0: <MII bus> on dc0
ukphy0: <Generic IEEE 802.3u media interface> on miibus0
ukphy0: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
ed0: <NE2000 PCI Ethernet (RealTek 8029)> port 0xec00-0xec1f irq 9 at
device 10.
0 on pci0
ed0 address 52:54:05:de:73:1b, type NE2000 (16 bit)
isa0: too many dependant configs (8)
isa0: unexpected small tag 14
orm0: <Option ROM> at iomem 0xc0000-0xc7fff on isa0
fdc0: <NEC 72065B or clone> at port 0x3f0-0x3f5,0x3f7 irq 6 drq2 on
isa0
fdc0: FIFO enabled, 8 bytes threshold
fd0: <1440-KB 3.5" drive> on fdc0 drive 0
atkbd0: <Keyboard controller (i8042)> at port 0x60,0x64 on isa0
atkbd0: <AT Keyboard> flags 0x1 irq1 on atkbd0
kbd0 at atkbd0
psm0: <PS/2 Mouse> irq 12 on atkbd0
psm0: model Generic PS/@ mouse, device ID 0
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff on
isa0
sc0: <System console> at flags 0x100 on isa0
```

```

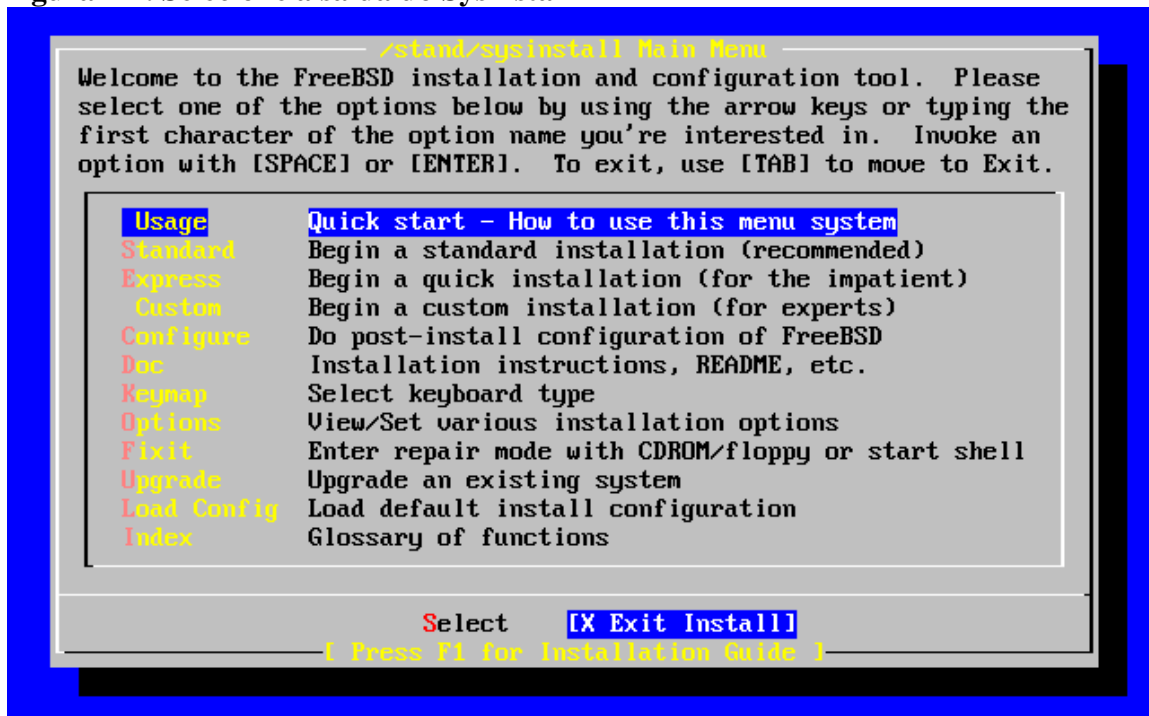
sc0: VGA <16 virtual consoles, flags=0x300>
sio0 at port 0x3f8-0x3ff irq 4 flags 0x10 on isa0
sio0: type 16550A
sio1 at port 0x2f8-0x2ff irq 3 on isa0
sio1: type 16550A
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
pppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/15 bytes threshold
plip0: <PLIP network interface> on ppbus0
ad0: 8063MB <IBM-DHEA-38451> [16383/16/63] at ata0-master UDMA33
acd0: CD-RW <LITE-ON LTR-1210B> at ata1-slave PIO4
Mounting root from ufs:/dev/md0c
/stand/sysinstall running as init on vty0

```

Verifique os resultados do reconhecimento cuidadosamente para garantir que o FreeBSD encontrou todos os dispositivos que você esperava. Se algum equipamento não foi encontrando, ele não será listado. Se o item em questão precisou ter seus recursos de IRQ e endereço de memória configurados, verifique se estes valores foram definidos corretamente.

Caso seja necessário modificar as configurações do reconhecimento de dispositivos, é simples sair do **sysinstall** e recomeçar o processo de instalação. Esta também é uma ótima forma de se tornar mais familiar com o processo.

Figura 2-7. Selecione a saída do Sysinstall



Utilize as teclas de setas para selecionar a opção Exit Install do menu principal de instalação. A seguinte mensagem será exibida:

```

      User Confirmation Requested
Are you sure you wish to exit? The system will reboot
(be sure to remove any floppies from the drives).

      [ Yes ]      No

```

O programa de instalação será reiniciado se o CDROM for mantido no leitor de CD e a opção [Yes] for selecionada.

Se estiver iniciando o sistema por disquetes, será necessário retirar o disco do `mfsroot.flp` e troca-lo pelo `kern.flp` antes da reinicialização.

2.4. Introdução ao Sysinstall

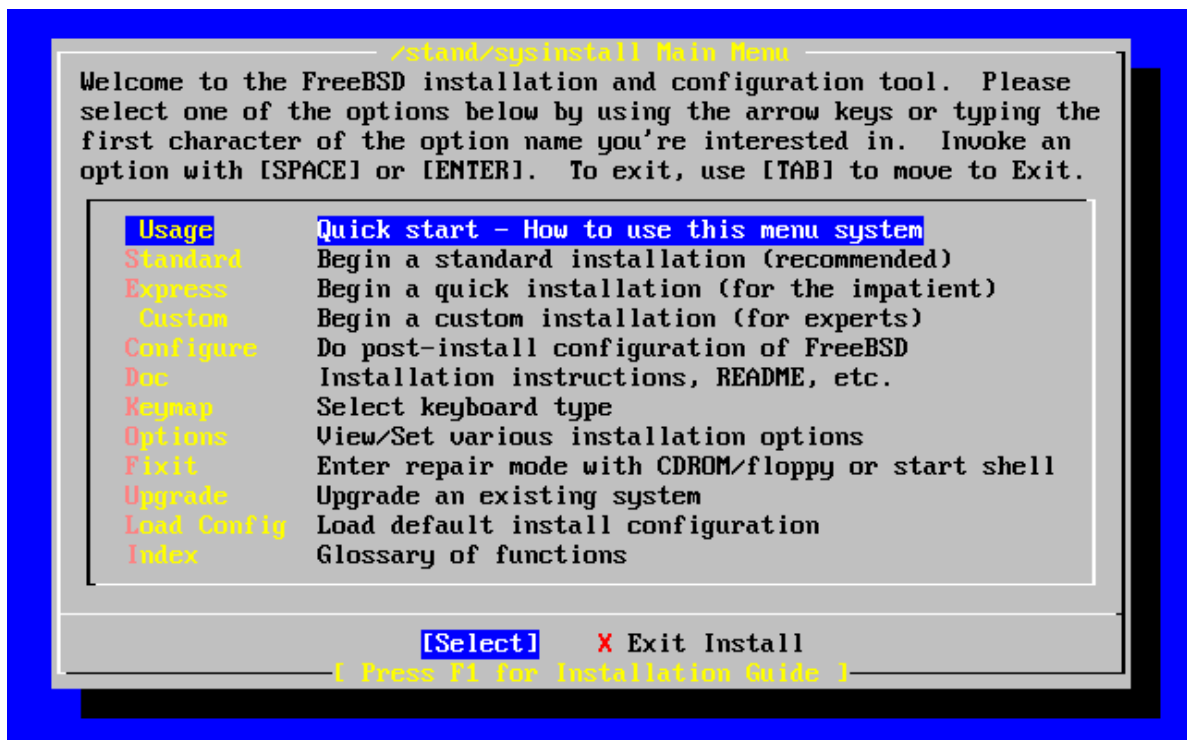
O utilitário **sysinstall** é o programa de instalação oferecido pelo Projeto FreeBSD. É baseado em console e é dividido em vários menus e telas que podem ser utilizados para configurar e controlar o processo de instalação.

O sistema de menu do **sysinstall** é controlado pelas teclas de flechas, **Enter**, **Space**, e outras teclas. Descrição detalhada destas teclas e o que elas fazem está disponível nas informações de utilização do **sysinstall**.

Para visualizar tais informações, garanta que a opção Usage está marcada, no menu do programa, e que o ítem [Select] está selecionado, como apresentados em [Figura 2-8](#), depois aperte **Enter**.

As instruções para utilizar esse sistema de menu serão apresentadas. Depois de revisá-las, aperte **Enter** para voltar ao menu principal.

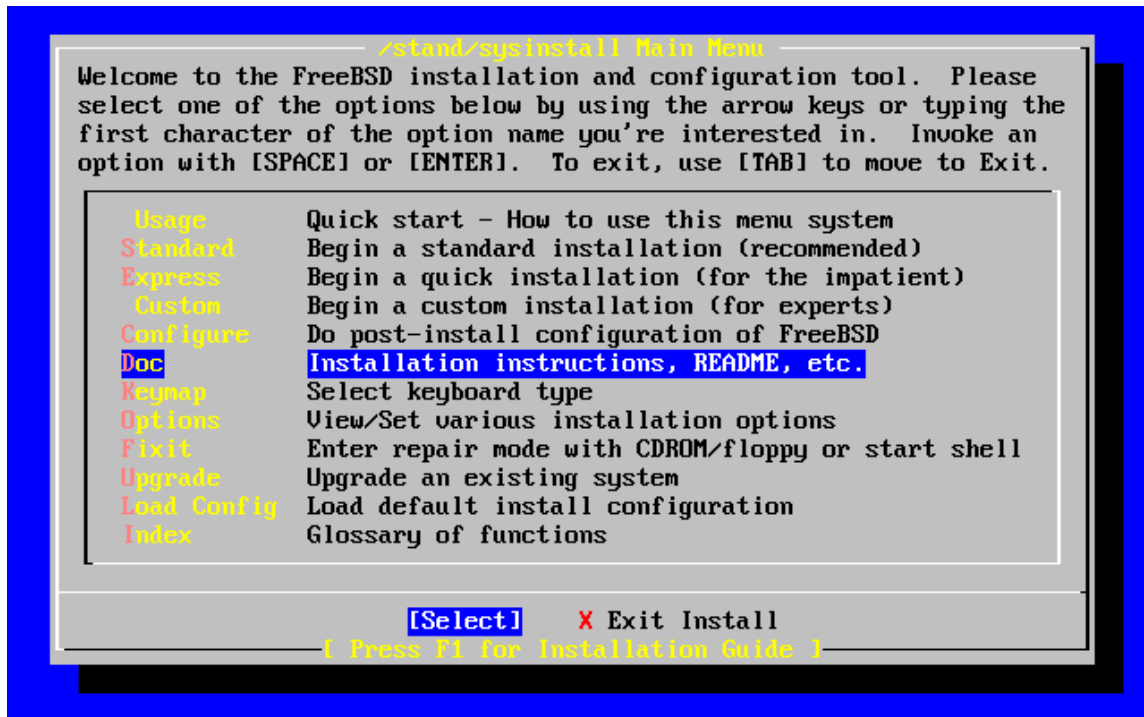
Figura 2-8. Selecionando a opção *Usage* do menu do Sysinstall



2.4.1. Selecionando o menu de documentação

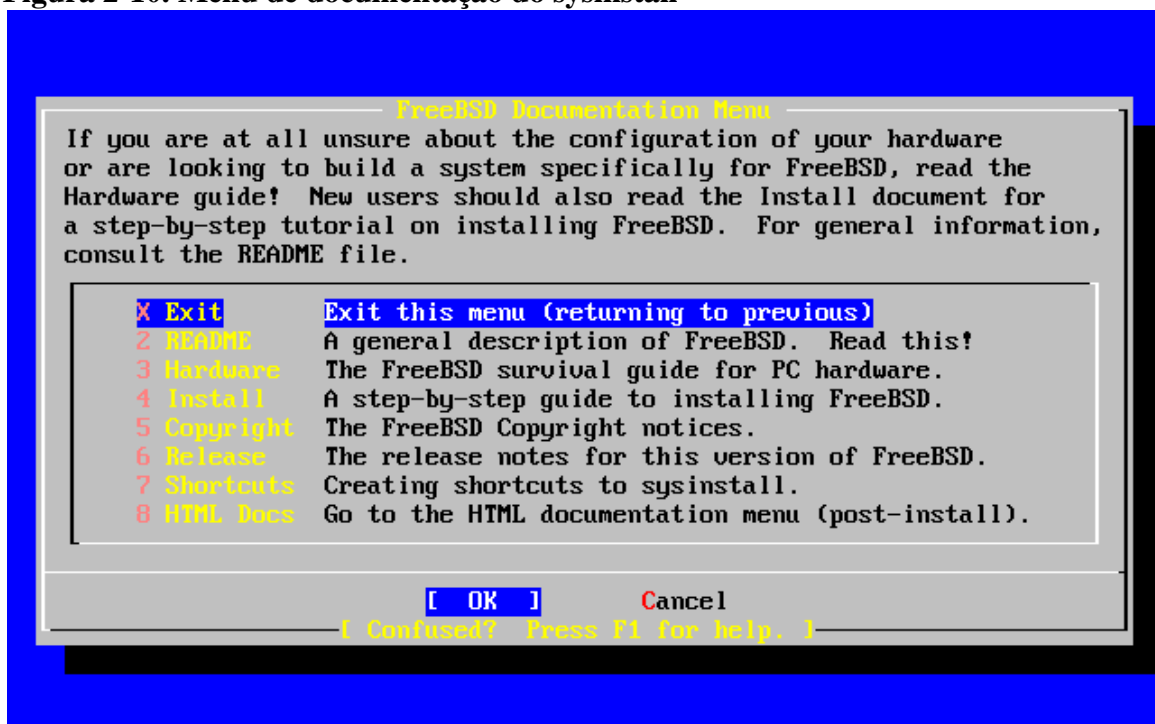
No menu principal, selecione a opção Doc com as setas e aperte **Enter**.

Figura 2-9. Selecionando o menu de documentação



Isso apresentará o menu de documentação.

Figura 2-10. Menu de documentação do sysinstall



É importante ler os documentos disponibilizados pelo menu em questão.

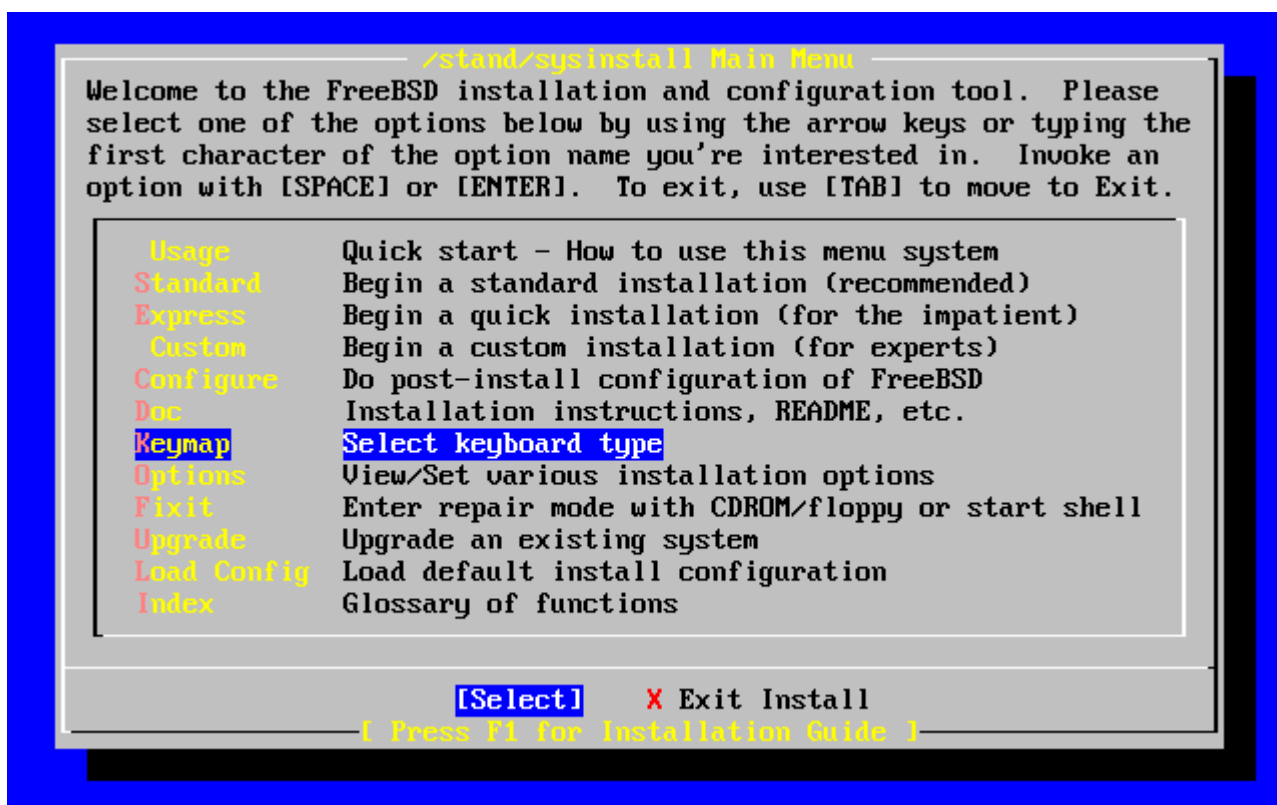
Para visualizar algum dos documentos, selecione-o com as setas e aperte **Enter**. Quando terminar a leitura do documento, pressione **Enter** novamente para retornar ao menu de documentação.

Para voltar ao menu principal de instalação, selecione Exit com as teclas de setas e pressione **Enter**.

2.4.2. Selecionando o menu de mapeamento do teclado

Para modificar o mapeamento de teclado, utilize as teclas de setinhas do menu e tecla **Enter**. Isso é necessário se o seu teclado não for padrão ou for não-US.

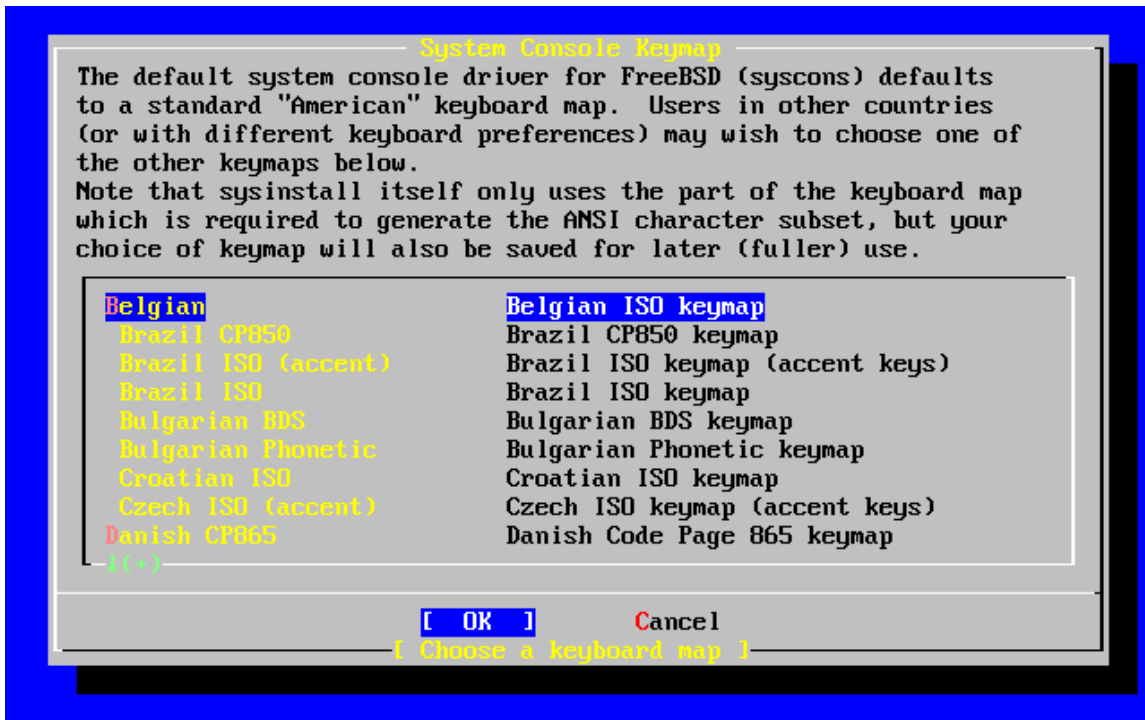
Figura 2-11. Menu principal do Sysinstall



O mapeamento distinto do teclado pode ser selecionado escolhendo o ítem em questão no menu principal e pressionando **Space**. Apertar **Space** novamente remove a seleção deste ítem. Quando terminado, escolha a opção [OK] utilizando as teclas de setinhas e aperte **Enter**.

Uma lista parcial é apresentada na representação desta tela. Selecionar [Cancel] apertando **Tab** forçará a utilização do mapeamento padrão, e voltará ao menu principal de instalação.

Figura 2-12. Menu de mapeamento do teclado do Sysinstall



2.4.3. Tela de opções de instalação

Selecione Options e aperte **Enter**.

Figura 2-13. Menu principal do Sysinstall

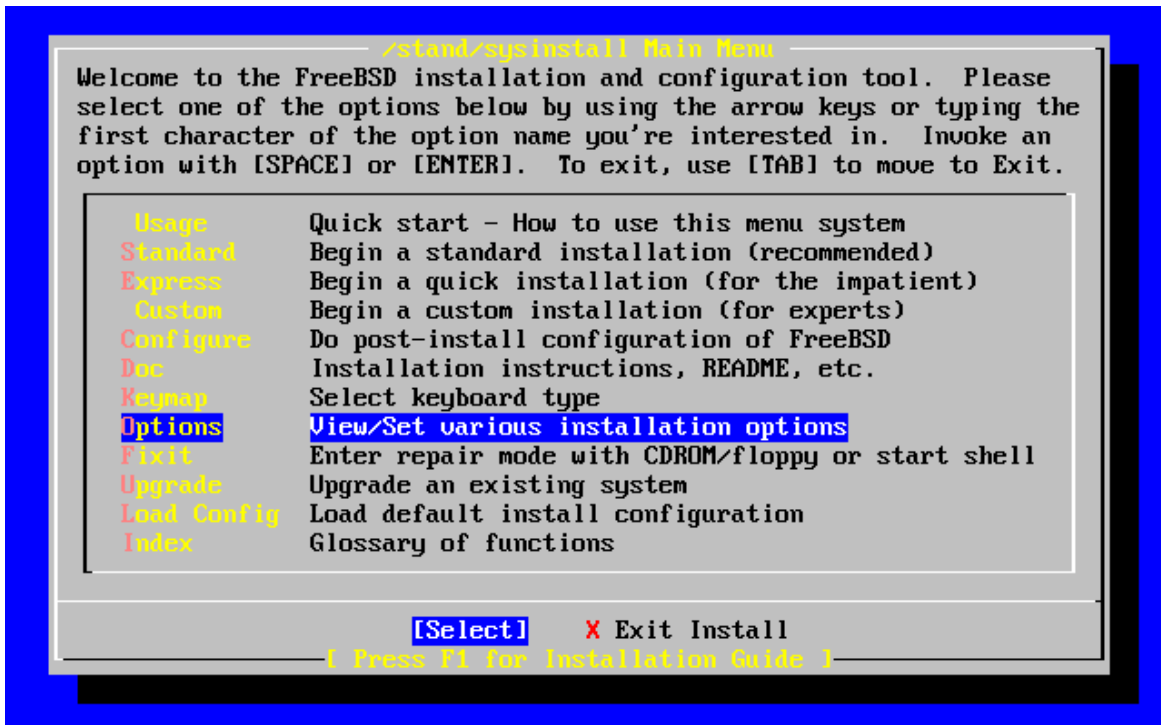


Figura 2-14. Opções do Sysinstall

```
Options Editor
Name          Value          Name          Value
-----
NFS Secure    YES           Media Type    <not yet set>
NFS Slow      NO            Media Timeout 300
Debugging     NO            Package Temp  /var/tmp
No Warnings   NO            Newfs Args    -b 16384 -f 2048
Yes to All    NO            Fixit Console serial
DHCP          NO            Config save   YES
IPv6          NO            Re-scan Devices <*>
FTP username  ftp           Use Defaults  [RESET!]
Editor        /usr/bin/ee
Tape Blocksize 20
Extract Detail high
Release Name   4.7-RELEASE
Install Root   /
Browser package links
Browser Exec   /usr/local/bin/links

Use SPACE to select/toggle an option, arrow keys to move,
? or F1 for more help. When you're done, type Q to Quit.

NFS server talks only on a secure port
```

Os valores padrão normalmente são o bastante para a maioria dos usuários, e não tem a necessidade de serem modificados. O nome do *release* (*release name*) irá variar de acordo com a versão do sistema que está sendo instalada.

A descrição do item selecionado aparecerá na base da tela, em cores azuis. Note que uma das opções é Use Defaults que tem a função de redefinir todos os itens para seus valores padrão.

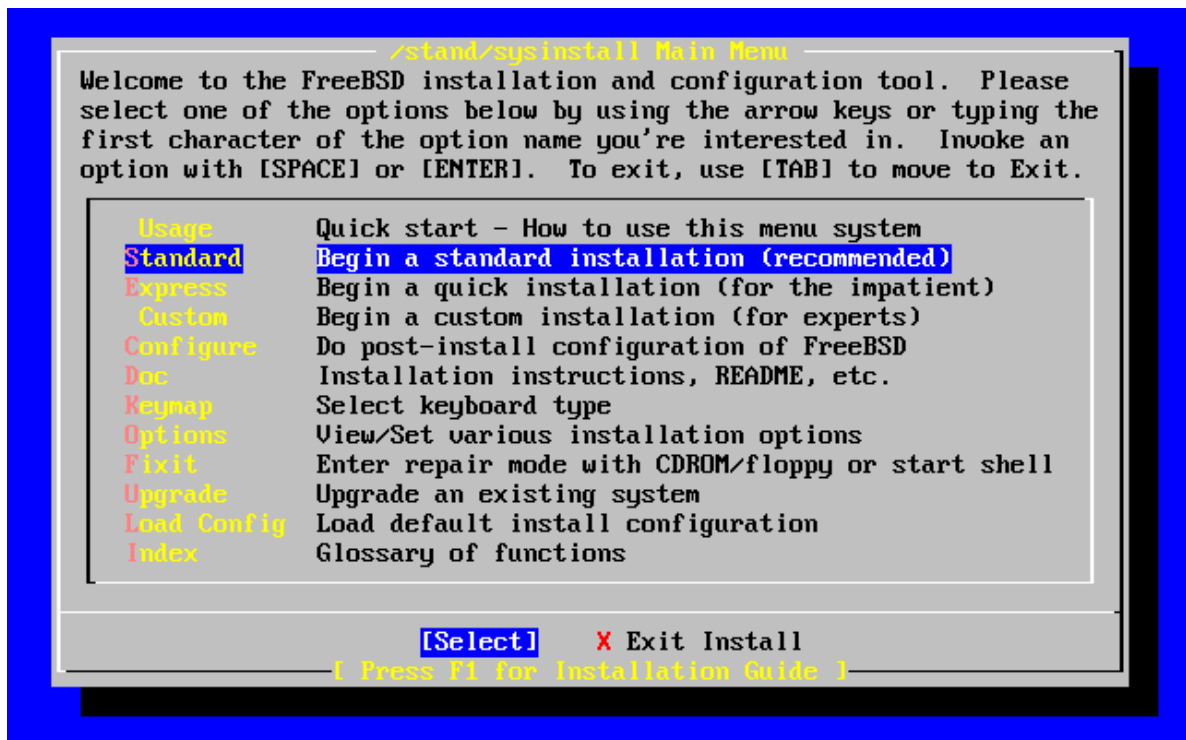
Aperte **F1** para ler a tela de ajuda sobre as várias opções.

Ao apertar a tecla **Q** o foco volta ao menu principal de instalação.

2.4.4. Iniciando a instalação padrão

A instalação padrão (Standard) é a opção recomendada aos usuários recém chegados ao ambiente UNIX® ou FreeBSD. Utilize as teclas de setinhas para selecionar a opção Standard e então aperte **Enter** para iniciar a instalação.

Figura 2-15. Iniciando a instalação padrão



2.5. Alocando espaço em disco

A primeira tarefa é alocar espaço em disco para o FreeBSD e rotular este espaço de forma que o **sysinstall** possa prepará-lo. Para fazer isso é necessário saber como o FreeBSD pretende encontrar as informações em disco.

2.5.1. Numeração dos discos no BIOS

Antes de instalar e configurar o FreeBSD em seu sistema, existe um ítem fundamental com o qual você deve ter atenção, especialmente se possuir múltiplos discos rígidos.

Em um PC que utiliza um sistema operacional dependente do BIOS (*BIOS-dependent OS*) como MS-DOS® ou Microsoft® Windows®, o BIOS é capaz de abstrair a ordem normal dos discos, e o sistema operacional reconhece tal modificação. Isso permite que o usuário inicie o sistema à partir de um disco qualquer, ao invés do chamado "primário mestre". Isso é especialmente conveniente para usuários que acreditam que a forma mais rápida e barata de manter o backup de um sistema é comprando um segundo disco rígido idêntico ao primeiro e realizar rotinas de cópias do primeiro disco no segundo, utilizando os programas **Ghost®** ou **XCOPY**. Dessa forma, se o primeiro dispositivo falhar ou for atacado por um vírus, ou mesmo se for danificado por defeito no sistema operacional, é fácil recuperar o sistema simplesmente instruindo o BIOS para modificar a unidade de inicialização. É como mudar os cabos do disco, mas sem ter que abrir o gabinete.

Sistemas mais caros com controladoras SCSI costumam incluir extensões no BIOS que permitem que os discos SCSI sejam reordenados de forma similar para vários discos.

Usuários que tem costume de tirar vantagens destas características podem se surpreender ao perceber que o resultado no FreeBSD não é como o esperado. FreeBSD não utiliza o BIOS dessa forma e simplesmente ignora o "mapeamento lógico dos dispositivos no BIOS". Isso pode levar o usuário à situações perplexas especialmente quando a geometria dos discos é exatamente igual e foram clonados de um para outro disco.

Ao usar o FreeBSD, sempre restaure a numeração usual do BIOS antes de iniciar a instalação do sistema e mantenha-a desta forma. Se for necessário alternar entre os discos, então faça-o da forma difícil, abrindo o gabinete e modificando os *jumpers* e cabos.

Ilustrando as excepcionais aventuras dos arquivos de Bill e Fred:

Bill resolve destruir outra máquina Wintel para montar outro servidor FreeBSD para o Fred. Bill configura um único disco SCSI na unidade SCSI número zero, e instala FreeBSD nela.

Fred passa a utilizar o sistema, mas depois de vários dias nota que o disco SCSI antigo passa e reportar vários erros lógicos e relato este fato ao Bill.

Depois de vários dias mais, Bill decide que é hora de resolver a situação, então ele consegue um disco SCSI idêntico no "armazém" de discos na sala dos fundos. Uma varredura inicial de superfície indica que o dispositivo funciona corretamente, então Bill instala este disco SCSI na unidade número quatro e faz uma imagem cópia dos dados no disco da unidade zero para o disco na quatro. Agora o dispositivo está instalado e funcionando corretamente. Bill decide então utilizar o disco e faz uso da opção no BIOS SCSI para re-ordenar os dispositivos de disco de forma que o sistema inicialize à partir da unidade quatro. FreeBSD inicia e entra no ar de forma correta.

Fred continua seu trabalho por vários dias afincado, e logo Bill e Fred decidem que é hora de uma nova aventura -- hora de atualizar para uma versão mais recente do FreeBSD. Bill retira a unidade zero SCSI e troca ela por um outro disco idêntico, lá do mesmo "armazém". Bill reinstala o FreeBSD no novo disco na unidade zero utilizando os disquetes mágicos de FTP via Internet do Fred. A instalação termina corretamente.

Fred utiliza a nova versão do FreeBSD por alguns dias e se certifica que ela é boa o bastante para colocar no departamento de engenharia. É hora de copiar todo seu trabalho do sistema antigo. Fred monta a unidade SCSI quatro (a última cópia da versão anterior do FreeBSD). Fred quase desmaia ao constatar que nenhum de seus tão preciosos dados de trabalho está presente na unidade SCSI quatro.

Onde os dados foram parar?

Quando Bill fez uma cópia do disco SCSI original da unidade zero na unidade SCSI quatro, a unidade quatro se tornou o "novo clone". Quando Bill re-ordenou o BIOS SCSI de forma que pudesse iniciar a partir da unidade quatro, ele estava apenas enganando a si mesmo. FreeBSD continuava rodando à partir da unidade SCSI zero. Realizar esse tipo de modificação no BIOS faz com que todo o código do carregador (

Loader) e do setor de inicialização seja iniciado do disco selecionado, mas quando o FreeBSD decola, a ordem das unidades é ignorada e o FreeBSD reconhece a numeração física (original) dos discos. Nesta ilustração o sistema continua sendo utilizado à partir da unidade zero, e todos os dados do Fred continuavam lá, e não na unidade SCSI número quatro. O fato de o sistema parecer estar sendo executado na unidade 4 era apenas artifício da expectativas humanas.

Felizmente, ficamos felizes de mencionar que nenhum byte sequer foi perdido ou danificado com isso. A unidade SCSI zero antiga foi realinhada, e todo o trabalho de Fred voltou às suas mãos (e agora Bill sabe que ele pode contar além de zero).

Embora discos SCSI tenham sido utilizados nesta ilustração, o conceito se aplica igualmente à dispositivos IDE.

2.5.2. Criando partições com FDisk

Nota: Nenhuma alteração que seja realizada neste momento será escrita em disco. Se você acreditar que tenha cometido algum erro e deseja iniciar o processo novamente, pode utilizar o menu para sair do **sysinstall** e começar de novo, ou apertar **U** para utilizar a opção de ``desfazer" (*Undo*). Se você ficar confuso e não estiver certo sobre como sair, lembre-se que você pode sempre desligar seu computador.

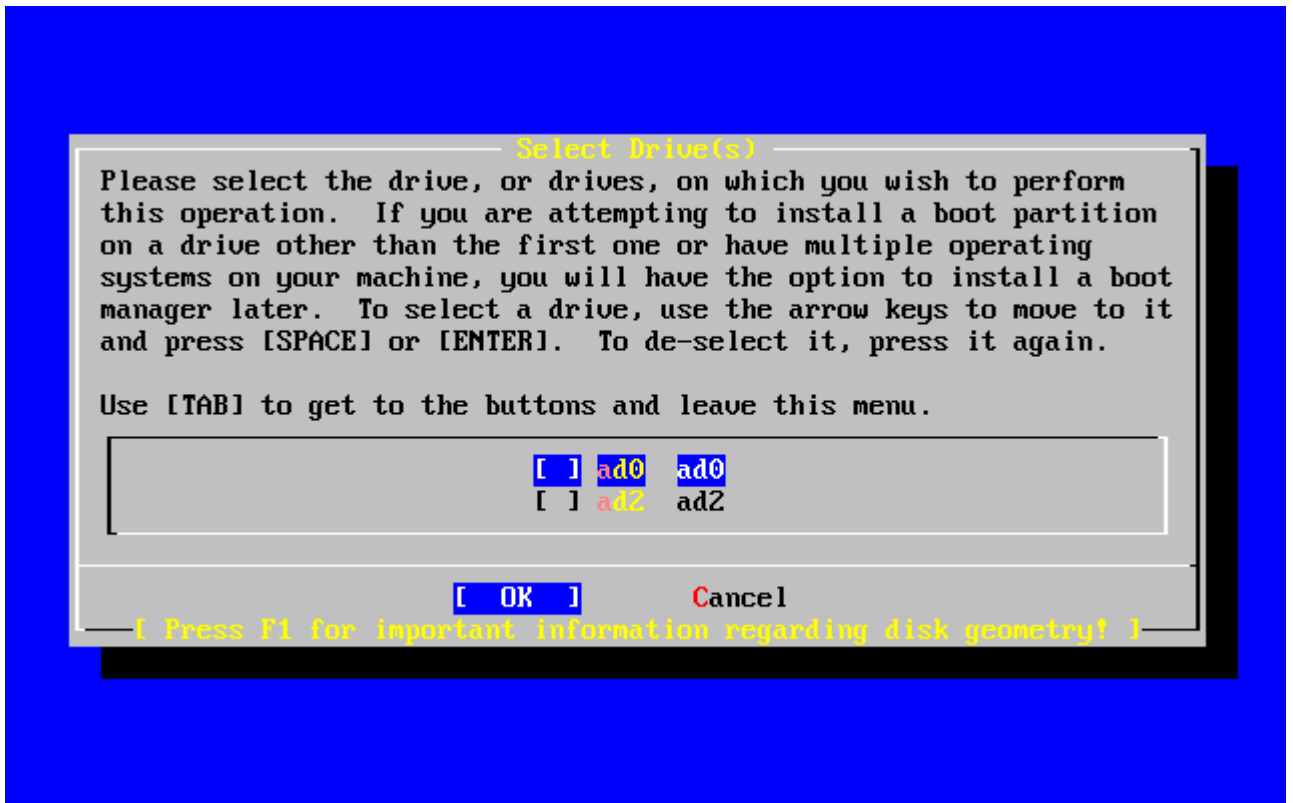
Após selecionar a instalação padrão do **sysinstall** você verá esta mensagem:

```
Message
In the next menu, you will need to set up a DOS-style ("fdisk")
partitioning scheme for your hard disk. If you simply wish to devote
all disk space to FreeBSD (overwriting anything else that might be on
the disk(s) selected) then use the (A)ll command to select the
default
partitioning scheme followed by a (Q)uit. If you wish to allocate
only
free space to FreeBSD, move to a partition marked "unused" and use
the
(C)reate command.
[ OK ]

[ Press enter or space ]
```

Aperte **Enter** como indicado. Você verá uma lista de discos rígidos que o kernel encontrou quando foi carregado. [Figura 2-16](#) apresenta um exemplo de sistema com dois discos IDE. Eles foram chamados de `ad0` e `ad2`.

Figura 2-16. Seleção do disco para o fdisk



Você deve estar se perguntando por que `ad1` não é listado aqui. Por que ela foi pulada?

Considere o que aconteceria se você dispusesse de dois discos IDE, um como primário na primeira controladora IDE, e um como primário na segunda controladora IDE. Se o FreeBSD os numerasse conforme os encontrasse, como dispositivo `ad0` e `ad1`, então tudo funcionaria.

Mas então, se você adicionasse um terceiro disco, como secundário na primeira controladora IDE, seriam agora `ad1`, e o `ad1` anterior seria chamado de `ad2`. Considerando que o nome dos dispositivos (como, por exemplo `ad1s1a`) são utilizados para identificar os sistemas de arquivos, repentinamente você pode perceber que seus sistemas de arquivos não estão mais dispostos corretamente, e teria que modificar sua configuração do FreeBSD.

Para resolver isso, o kernel pode ser configurado para nomear discos IDE com base em sua disposição física, e não na ordem em que eles são encontrados. Com este esquema, o disco primário na segunda controladora IDE será *sempre* o `ad2`, mesmo que não existam dispositivos `ad0` ou `ad1`.

Esta é a configuração padrão no kernel do FreeBSD, que nesta tela apresenta `ad0` e `ad2`. A estação de onde este *screenshot* foi tirado dispunha de dois discos IDE nos dois canais primários de cada controladora IDE, e nenhum disco nos canais secundários.

Você deve escolher o disco onde o FreeBSD deve ser instalado, e depois disso apertar [OK]. O **FDisk** será iniciado, com uma interface similar à esta apresentada na [Figura 2-17](#).

A tela do **FDisk** está dividida em três seções.

A primeira seção cobre as duas primeiras linhas da aplicação, e mostra os detalhes do disco atualmente selecionado, incluindo a nomenclatura FreeBSD para o nome, a geometria de disco e o tamanho total do mesmo.

A segunda seção apresenta as partições que estão no disco, onde elas iniciam e terminam, qual seu tamanho, o nome que o FreeBSD atribui à elas, e suas descrições e sub-tipos. Este exemplo apresenta duas pequenas partições não utilizadas, que são artifícios do esquema de disposição de discos no PC. Também apresenta uma grande partição FAT, que quase sempre aparece como `c:` no MS-DOS / Windows e uma partição estendida, que pode conter outras letras de dispositivos para o MS-DOS / Windows.

A terceira seção apresenta os comandos que são disponibilizados pelo **FDisk**.

Figura 2-17. Partições típicas do Fdisk antes de sua edição

```
Disk name:      ad0                               FDISK Partition Editor
DISK Geometry: 16383 cyls/16 heads/63 sectors = 16514064 sectors (8063MB)

Offset      Size(ST)      End      Name  PType      Desc  Subtype  Flags
-----
0           63           62      -     6      unused    0
63         4193217      4193279  ad0s1  2      fat       14      >
4193280     1008        4194287  -     6      unused    0      >
4194288     12319776    16514063 ad0s2  4      extended  15      >

The following commands are supported (in upper or lower case):

A = Use Entire Disk      G = set Drive Geometry  C = Create Slice      F = `DD' mode
D = Delete Slice        Z = Toggle Size Units   S = Set Bootable     I = Wizard m.
T = Change Type         U = Undo All Changes    Q = Finish

Use F1 or ? to get more help, arrow keys to select.
```

O que você for fazer agora depende de como você pretende particionar seu disco.

Se a intenção é utilizar o FreeBSD no disco todo (o que irá apagar todos outros dados presentes no disco ao confirmar a continuidade do processo de instalação no **sysinstall**) basta apertar a tecla **A** que significa "utilizar o disco inteiro" (Use Entire Disk). As

partição existentes serão apagadas e substituídas por uma pequena área marcada como *unused* (mais uma vez, característica das disposições do PC), e uma grande partição para o FreeBSD. Ao fazer isso, é aconselhável selecionar a nova partição FreeBSD com as setas, e apertar a tecla **S** para marcar a partição como inicializável. A tela terá aparência similar à [Figura 2-18](#). Note a opção *A* na coluna *Flags*, que indica que esta partição é uma partição ``ativa" (*active*) e servirá como dispositivo de inicialização.

Se a intenção é apagar uma partição existente de forma a conseguir espaço em disco para o FreeBSD, então você deve selecionar a partição e apertar **D**. Em seguida pode fazer uso da tecla **C** e responder qual o tamanho pretendido para a nova partição que você pretende criar. Entre com o valor apropriado e aperte **Enter**. O valor padrão aqui indica o tamanho máximo possível para esta partição, que na realidade é o maior bloco contíguo de espaço não alocado, ou o tamanho do disco inteiro.

Se já existe espaço para o FreeBSD (por exemplo, se você garantiu espaço com alguma aplicação como o **PartitionMagic®**) então pode utilizar a tecla **C** para criar uma nova partição. Mais uma vez, será necessário informar o tamanho da nova partição à ser criada.

Figura 2-18. Partição Fdisk utilizando todo o disco

```

Disk name:      ad0                               FDISK Partition Editor
DISK Geometry: 16383 cyls/16 heads/63 sectors = 16514064 sectors (8063MB)

Offset      Size(ST)      End      Name  PType      Desc  Subtype  Flags
-----
0           63             62      -     6          unused  0
63          16514001      16514063 ad0s1  3          freebsd 165      CA

The following commands are supported (in upper or lower case):

A = Use Entire Disk      G = set Drive Geometry  C = Create Slice  F = `DD' mode
D = Delete Slice        Z = Toggle Size Units   S = Set Bootable  I = Wizard m.
T = Change Type         U = Undo All Changes    Q = Finish

Use F1 or ? to get more help, arrow keys to select.

```

Ao terminar, aperte **Q**. Suas modificações serão salvas pelo **sysinstall**, mas por enquanto não serão efetivamente aplicadas em disco.

2.5.3. Instale um Gerenciador de Inicialização (*Boot Manager*)

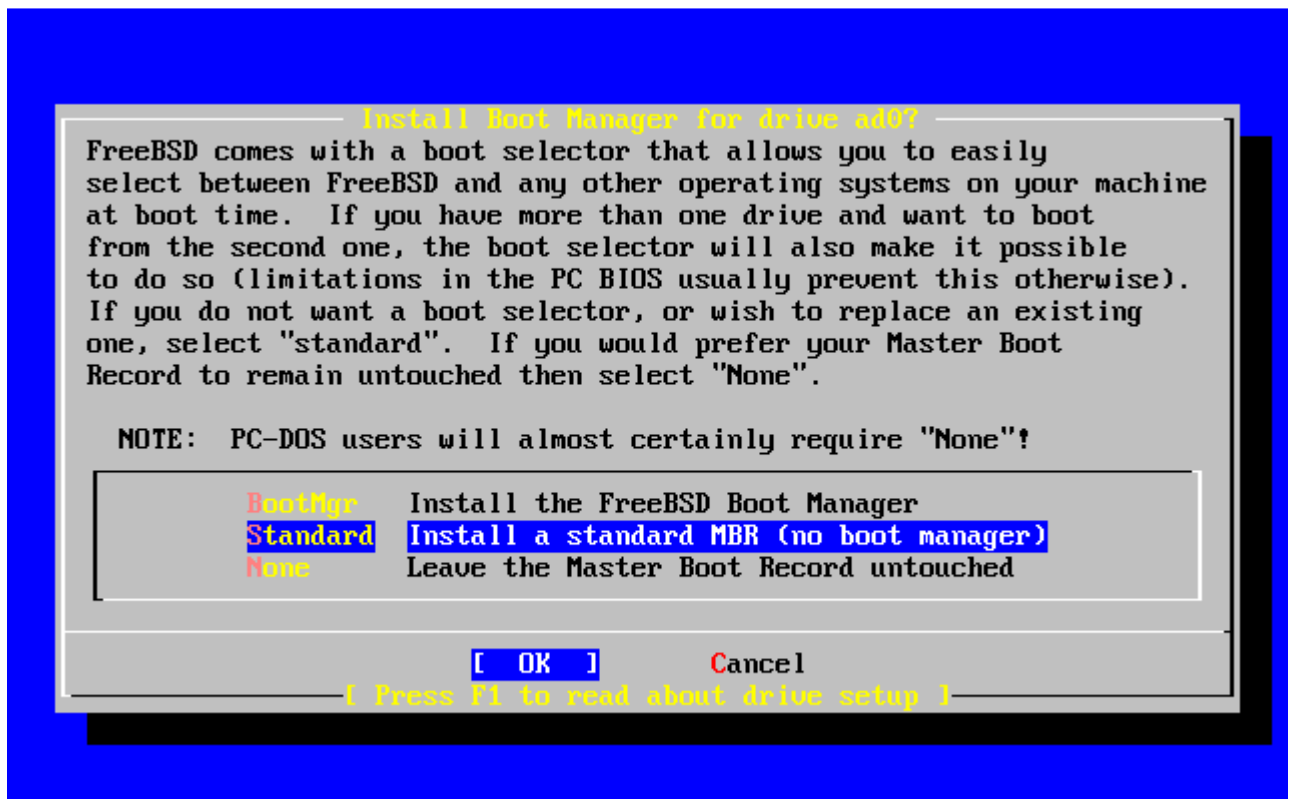
Agora você tem a opção de instalar um gerenciador do processo de inicialização do sistema, ou um gerenciador de *Boot* como é conhecido. É indicado instalar o gerenciador de inicialização nativo do FreeBSD, se:

- Você tem mais de um disco, e instalou o FreeBSD em algum disco que não seja o primeiro.
- Você instalou o FreeBSD em conjunto com outro sistema operacional, no mesmo disco, e deseja escolher em que momento o FreeBSD ou o outro sistema devem ser carregados.

Se o FreeBSD será o único sistema operacional nesta estação, instalado no primeiro disco rígido, o inicializador padrão do FreeBSD (opção Standard) é o recomendado. Escolha a opção None se a intenção for utilizar um outro gerenciador de inicialização não nativo, ou se deseja não modificar o setor mestre de inicialização do disco.

Escolha sua opção e aperte **Enter**.

Figura 2-19. Menu do *Boot Manager* do FreeBSD



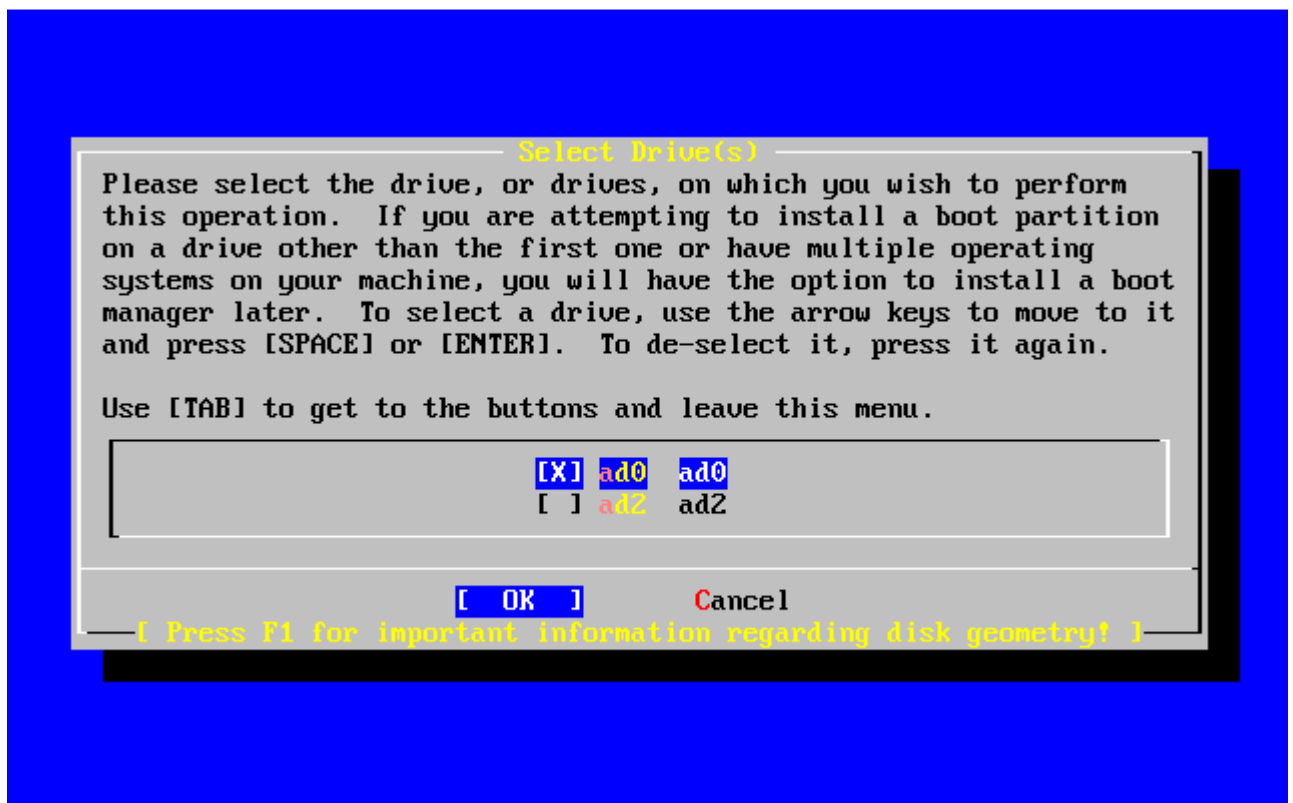
A tela de ajuda oferecida ao pressionar **F1**, oferece algumas informações sobre os problemas implícitos no compartilhamento de discos rígidos entre múltiplos sistemas operacionais.

2.5.4. Criando partições em outro disco

Se houver mais de um disco, o menu voltará à tela de seleção de unidades de armazenamento após a seleção do gerenciador de inicialização. Se a intenção for instalar FreeBSD em mais de um disco, então pode selecionar o outro disco e repetir o processo de particionamento usando o **FDisk**.

Importante: Se estiver instalando o FreeBSD em um disco diferente do primeiro, então o gerenciador de inicialização do FreeBSD deve ser instalado em ambos os discos.

Figura 2-20. Sair da seleção de discos



A tecla **Tab** alterna entre as opções [OK] e [Cancel].

Aperte a tecla **Tab** uma única vez para selecionar [OK] e depois aperte **Enter** para continuar a instalação.

2.5.5. Criando partições com o Disklabel

Agora você deve criar subpartições dentro de cada partição primária que você acabou de criar. Lembre-se que cada partição é identificada com letras de *a* à *h*, e que as partições *b*, *c* e *d* tem significados convencionais que devem ser notados aqui.

Algumas aplicações se beneficiam de alguns esquemas particulares de particionamento, especialmente se você está utilizando partições de tal disposição que utilize mais de um disco rígido. Contudo, para sua primeira instalação, você não precisa se empenhar tanto na definição do particionamento do seu FreeBSD. É mais importante instalar o FreeBSD e aprender a utilizá-lo. Você pode sempre reinstalar o sistema e modificar o esquema de particionamento várias vezes, até se tornar familiar o suficiente com o sistema operacional e seu processo de instalação, para aí sim passar a otimizar seu particionamento.

O esquema atual apresenta quatro partições -- uma para o espaço de *swap*, e três para sistemas de arquivos.

Tabela 2-2. Disposição de partições para o primeiro disco

Partição	Sistema de arquivos	Tamanho	Descrição
<i>a</i>	/	100 MB	Este é o sistema de arquivos raiz. Todos os outros sistemas de arquivos serão montados em algum ponto abaixo deste. Não é necessário armazenar muitos dados neste sistema de arquivos. Em uma instalação regular do FreeBSD, serão utilizados cerca de 40 MB de dados aqui. O espaço restante é para dados temporários e também oferece espaço para expansão se eventualmente versões futuras do FreeBSD precisarem de mais espaço na /.
<i>b</i>	N/D	2-3 x RAM	O espaço de <i>swap</i> do sistema é mantido nesta partição. Escolher a quantidade correta de espaço para <i>swap</i> é uma questão um tanto quanto artística. Uma boa regra base é multiplicar entre duas e três vezes o espaço total de memória física (RAM) disponível. É recomendado que, se você tem menos de 32 MB de memória, que defina ao menos 64 MB de <i>swap</i> .
<i>e</i>	/var	50 MB	Se você dispõe de mais de um disco, pode colocar uma partição de <i>swap</i> em cada. FreeBSD vai utilizar ambas para <i>swap</i> , e o fato do acesso estar distribuído em mais de uma partição de troca, em discos distintos, a velocidade efetiva da ação de <i>swap</i> é incrementada. Neste caso, calcule o espaço total de <i>swap</i> necessário e divida pela quantidade de discos disponíveis (digamos, 128 MB em dois discos), alocando a quantidade resultante em cada um, nesse exemplo, 64 MB de <i>swap</i> em cada disco.
			O diretório /var contém arquivos que são constantemente acessados; arquivos de <i>logs</i> , e outros arquivos administrativos.

Partição	Sistema de arquivos	Tamanho	Descrição
			A maioria destes arquivos são de escrita ou leitura intensivas no dia-a-dia da utilização do FreeBSD. Adicionar tais arquivos a outro sistema de arquivos permite que o acesso a estes arquivos seja otimizado sem afetar outros arquivos e diretórios que não tem o mesmo padrão de acesso.
<i>f</i>	<code>/usr</code>	Restante do disco	Todos seus outros arquivos serão armazenados tipicamente sob o <code>/usr</code> e seus subdiretórios.

Se o FreeBSD for ser instalado em um servidor com mais de um disco é ideal criar partições nos outros discos também. A forma mais básica de fazê-lo é criar duas partições no mínimo nos discos auxiliares uma para a *swap* e uma para ao menos um sistema de arquivos.

Tabela 2-3. Disposição de partições para discos subsequentes

Partição	Sistema de arquivos	Tamanho	Descrição
<i>b</i>	N/D	Ver descrições	Como discutido, o espaço de <i>swap</i> pode ser dividido em cada disco. Mesmo considerando que a partição <i>a</i> esteja livre, a convenção dita que as partições de troca (<i>swap</i>) devem ficar na partição <i>b</i> .
<i>e</i>	<code>/diskn</code>	Restante do disk	O restante do disco é utilizado por uma partição grande. Esta partição poderia ser a <i>a</i> , ao invés da partição <i>e</i> . Contudo, por convenção a partição <i>a</i> é reservada ao sistema de arquivos raiz (<code>/</code>). Você não tem obrigação de seguir este padrão, mas o sysinstall o faz, de forma que seguir este padrão torna sua instalação mais limpa. Você pode escolher montar este sistema de arquivos em qualquer local; este exemplo sugere que eles sejam montados como os diretórios <code>/disk n</code> , onde <i>n</i> corresponde ao número que muda de acordo com a ordem de cada disco. Mas você está livre para utilizar um outro esquema, se preferir.

Tendo definido a disposição de seu esquema de particionamento, você pode criá-lo utilizando o **sysinstall**. Você verá a seguinte mensagem:

```
Message
Now, you need to create BSD partitions inside of the fdisk
partition(s) just created.  If you have a reasonable amount of disk
space (200MB or more) and don't have any special requirements, simply
use the (A)uto command to allocate space automatically.  If you have
more specific needs or just don't care for the layout chosen by
(A)uto, press F1 for more information on manual layout.

[ OK ]
[ Press enter or space ]
```

Aperte **Enter** para iniciar o editor de partições do FreeBSD, chamado **Disklabel**.

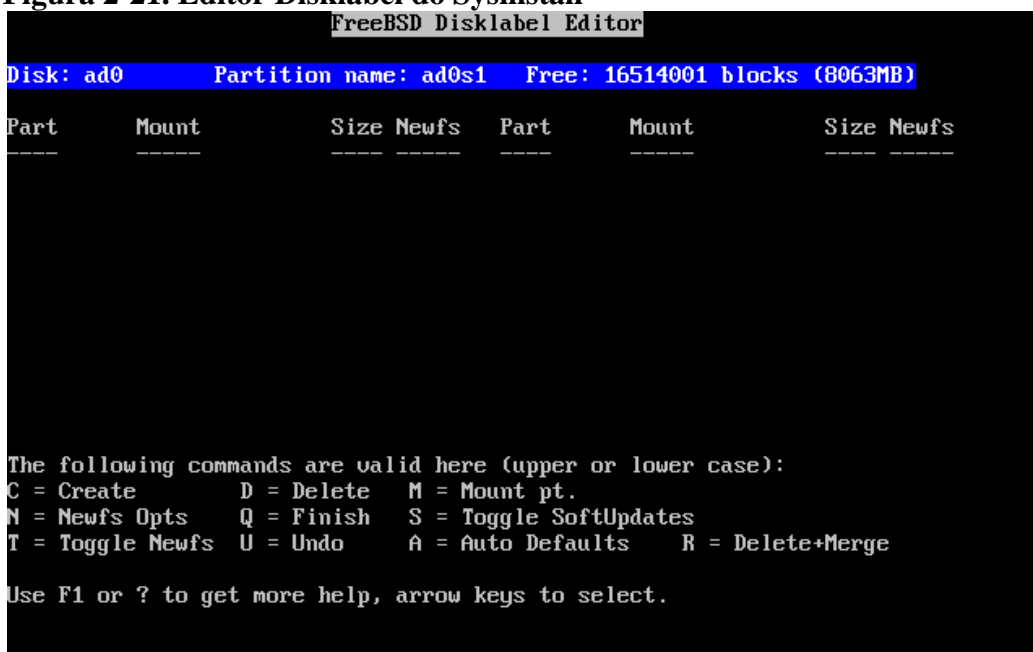
[Figura 2-21](#) apresenta o resultado da primeira execução do **Disklabel**. A tela é dividida em três seções.

As primeiras linhas mostra o nome do disco onde você está trabalhando e a partição onde está o sistema de arquivos que será criado (neste instante o **Disklabel** chama-o de "Nome da Partição", ou *Partition name*, ao invés do nome da subpartição). A tela apresenta também a quantidade de espaço livre disponível na partição primária, ou seja o espaço definido ao FreeBSD, mas que ainda não foi definido atribuído a nenhuma partição.

O meio da tela apresenta as partições que foram criadas, o nome do sistema de arquivos em cada partição, seu tamanho e algumas opções sobre a criação do sistema de arquivos.

A base da tela, na terceira imagem, apresenta as teclas de comandos que são validas no **Disklabel**.

Figura 2-21. Editor Disklabel do Sysinstall



```
FreeBSD Disklabel Editor
Disk: ad0 Partition name: ad0s1 Free: 16514001 blocks (8063MB)
Part Mount Size Newfs Part Mount Size Newfs
-----
The following commands are valid here (upper or lower case):
C = Create D = Delete M = Mount pt.
N = Newfs Opts Q = Finish S = Toggle SoftUpdates
T = Toggle Newfs U = Undo A = Auto Defaults R = Delete+Merge
Use F1 or ? to get more help, arrow keys to select.
```

O **Disklabel** pode criar partições automaticamente e atribuir espaço padrão para cada uma. Experimente esta opção agora, pressionando a tecla **A**. Você vai ver mensagens similares as que são apresentadas na [Figura 2-22](#). Dependendo do tamanho do disco que você tem disponível, os valores padrão vão variar para o tamanho apropriado. Mas isso não importa, afinal você não tem que aceitar o padrão.

Nota: A partir do FreeBSD 4.5, o particionamento padrão atribui uma partição dedicada ao diretório `/tmp` ao invés de fazer parte do sistema de arquivos raiz (`/`). Entre outras justificativas, isso ajuda a evitar que a raiz seja lotada de arquivos temporários.

Figura 2-22. Editor Disklabel do Sysinstall com padrões automáticos

```

FreeBSD Disklabel Editor
Disk: ad0      Partition name: ad0s1  Free: 0 blocks (0MB)

Part      Mount          Size Newfs  Part      Mount          Size Newfs
----      -
ad0s1a    /               128MB UFS     Y
ad0s1b    swap            503MB SWAP
ad0s1e    /var            256MB UFS+S Y
ad0s1f    /tmp            256MB UFS+S Y
ad0s1g    /usr            6919MB UFS+S Y

The following commands are valid here (upper or lower case):
C = Create          D = Delete      M = Mount pt.   W = Write
N = Newfs Opts     Q = Finish     S = Toggle SoftUpdates
T = Toggle Newfs   U = Undo       A = Auto Defaults R = Delete+Merge

Use F1 or ? to get more help, arrow keys to select.

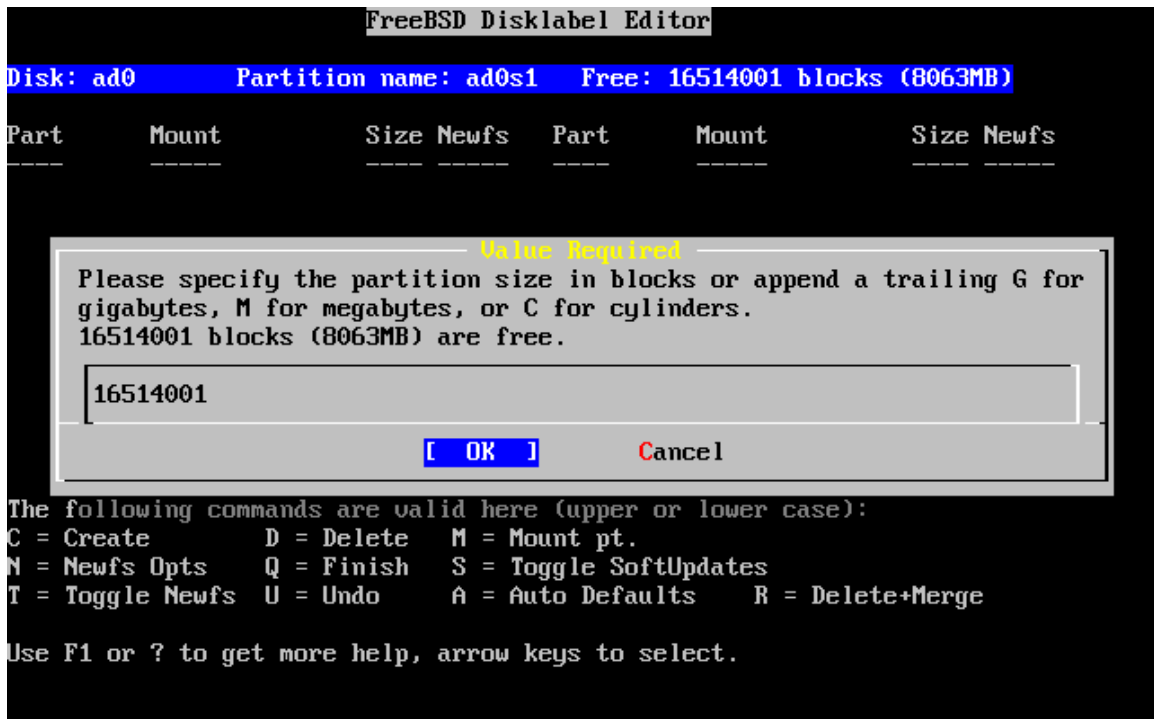
```

Se sua escolha for não utilizar o particionamento padrão, e você pretende substituí-lo com sua própria definição, use as teclas indicativas para selecionar a primeira partição, aperte **D** para apaga-la. Repita isto para apagar todas as partições sugeridas.

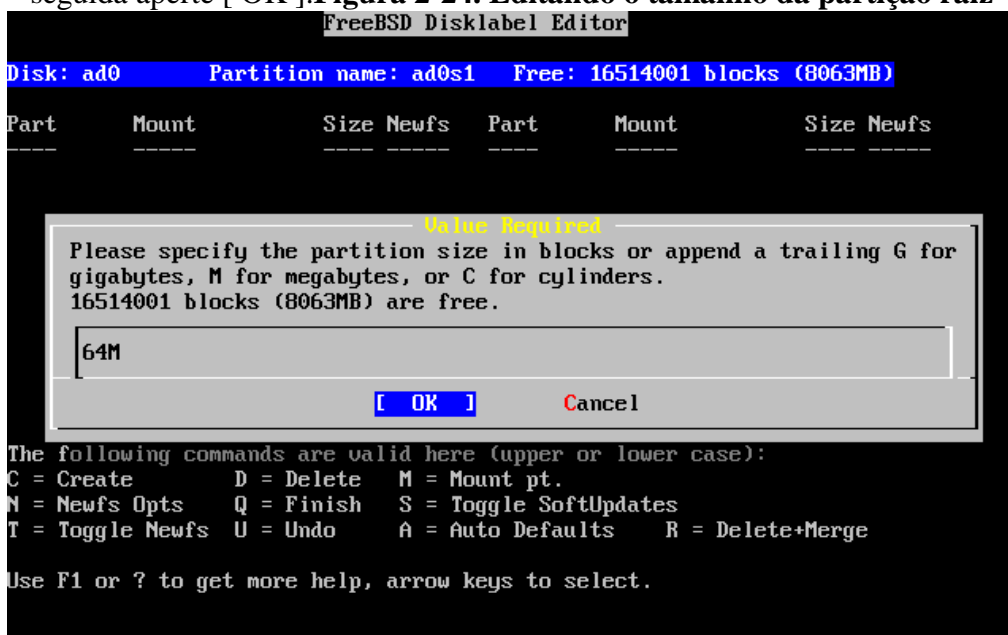
Para criar a primeira partição (*a*, montada como `/` -- raiz), garanta que a partição apropriada no topo da tela está selecionada e aperte **C**. Uma caixa de diálogos aparecerá perguntando qual o tamanho desejado para a nova partição (como apresentado em [Figura 2-23](#)). Você pode definir o número de blocos que deseja usar, ou um valor seguido por *M* para indicar "megabytes", *G* para "gigabytes", ou *C* para cilindros.

Nota: A partir do FreeBSD 5.X os usuários podem escolher o uso do UFS2 utilizando a opção *Custom Newfs (Z)*, criar rótulos com *Auto Defaults* e modifica-los com a opção *Custom Newfs* ou adicionar *-O 2* durante a criação regular do sistema de arquivos. Não se esqueça de manter a opção *-U* para indicar o uso do *SoftUpdates* se você fizer uso da opção de customização do *newfs*, *Custom Newfs*!

Figura 2-23. Liberando espaço para a partição raiz



O tamanho padrão apresentado criará uma partição que ocupa o restante do disco. Se estiver usando os tamanhos de partições descritos no exemplo, apague a definição existente utilizando **Backspace** e digite 64M, como apresentado na [Figura 2-24](#). Em seguida aperte [OK].



Tendo escolhido o tamanho da partição, o instalador solicitará que você informe se a partição vai conter um sistema de arquivos ou espaço para *swap*. A caixa de diálogo é a apresentada em [Figura 2-25](#). A primeira partição terá um sistema de arquivos, então marque o ítem FS e aperte **Enter**.

Figura 2-25. Escolhendo o tipo da partição raiz



Finalmente, por estar criando um sistema de arquivos, é necessário informar ao **Disklabel** onde o sistema de arquivos em questão deve ser montado. A caixa de diálogo é apresentada na [Figura 2-26](#). O ponto de montagem para a partição raiz é /, portanto digite / e aperte **Enter**.

Figura 2-26. Escolhendo o ponto de montagem raiz



A tela será atualizada para apresentar as partições recém criadas. Este processo deve ser repetido para as outras partições. Quando for criar a partição *swap* o sistema não pedirá o ponto de montagem da partição. Ao criar a partição final, o `/usr`, você pode manter o tamanho sugerido intacto, para utilizar o resto do disco.

A tela final do editor "DiskLabel" do FreeBSD terá aparência similar a [Figura 2-27](#), embora os valores escolhidos por você podem estar diferentes. Aperte **Q** para terminar.

Figura 2-27. Editor Disklabel do Sysinstall

```
FreeBSD Disklabel Editor
Disk: ad0      Partition name: ad0s1  Free: 0 blocks (0MB)
Part      Mount      Size Newfs  Part      Mount      Size Newfs
-----
ad0s1a    /              64MB UFS    Y
ad0s1b    swap           512MB SWAP
ad0s1e    /var           256MB UFS+S Y
ad0s1f    /usr           7231MB UFS+S Y

The following commands are valid here (upper or lower case):
C = Create      D = Delete    M = Mount pt.
N = Newfs Opts  Q = Finish    S = Toggle SoftUpdates
T = Toggle Newfs U = Undo      A = Auto Defaults  R = Delete+Merge

Use F1 or ? to get more help, arrow keys to select.
```

2.6. Escolhendo o que instalar

2.6.1. Seleccione o conjunto de distribuição

Decidir que conjunto de distribuição instalar depende muito da intenção de uso do sistema e da quantidade de espaço em disco disponível. As opções predefinidas vão da menor instalação possível à instalar tudo que estiver disponível. Usuários ainda não familiarizados com o ambiente UNIX® ou FreeBSD costumam escolher entre esses dois extremos. Customizar o conjunto de instalação tipicamente é uma tarefa para usuários mais experientes.

Aperte **F1** para obter mais informações sobre cada conjunto de distribuição e seu conteúdo. Ao terminar a leitura do arquivo de ajuda, aperte **Enter** para voltar ao menu de seleção de distribuições.

Se uma interface gráfica com o usuário for desejável, as distribuições precedidas por um **X** devem ser escolhidas. A configuração do **XFree86™** e seleção de um *desktop* padrão faz parte dos passos de pós-instalação do sistema.

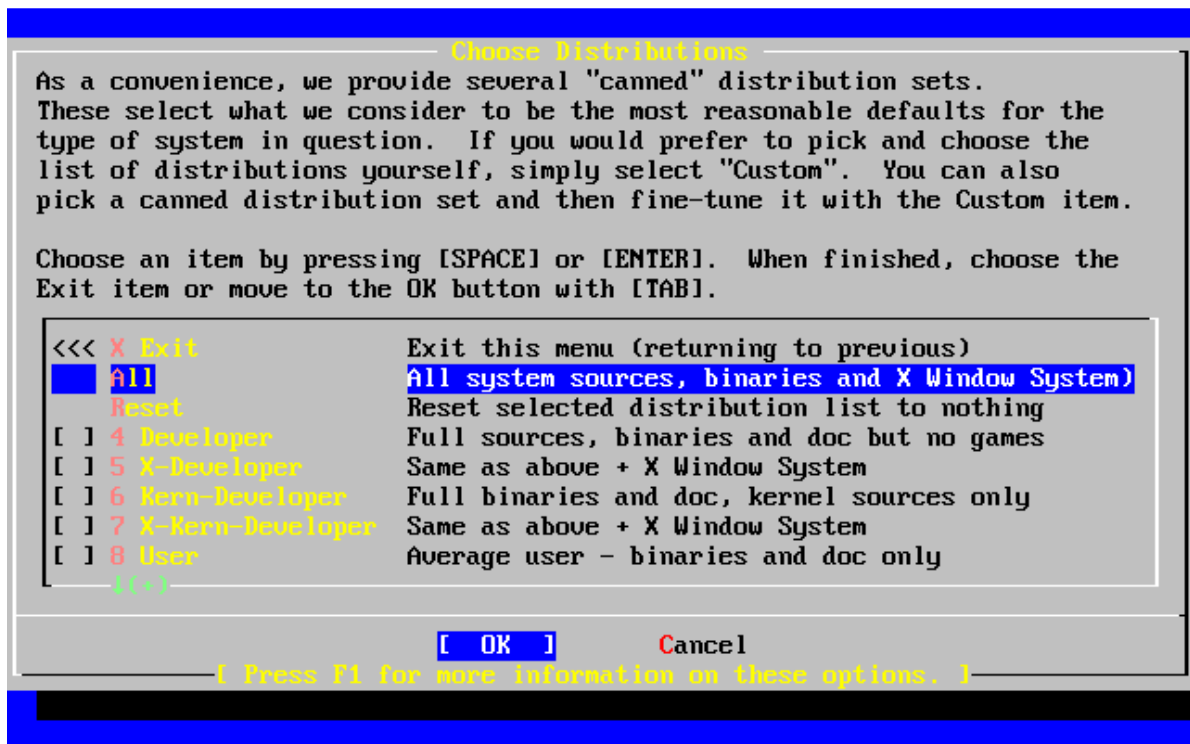
A versão padrão do **XFree86** que está instalada depende da versão do FreeBSD que você está instalando. Em versões do FreeBSD anteriores à 4.6 o **XFree86 3.X** é o instalado. Para o FreeBSD 4.6 e posteriores, **XFree86 4.X** é a série instalada.

Você deve verificar se sua placa de vídeo é suportada no sítio WWW do [XFree86](http://www.xfree86.org) e. Se a placa não for suportada pela versão padrão que o FreeBSD instalará, você deve desmarcar a distribuição em questão em favor de uma distribuição sem a instalação do X. Após a instalação do sistema, instale e configure a versão apropriada do **XFree86** usando a coleção de *ports*.

Se a compilação de um novo kernel é prevista, selecione uma opção que inclua o código fonte. Para mais detalhes sobre os motivos pelos quais um kernel customizado deve ser compilado ou sobre como compilar um novo kernel, veja a [Capítulo 9](#).

Obviamente o sistema mais versátil é o que inclui tudo que estiver disponível. Se tiver espaço em disco adequado, selecione a opção All como apresentado em [Figura 2-28](#) utilizando as teclas indicadoras e aperte **Enter**. Se existir preocupação sobre o espaço em disco considere utilizar a opção que melhor se adequar à sua situação. Não perca tempo tentando encontrar o conjunto perfeito de aplicações, tendo em vista que outras distribuições podem ser adicionadas após a instalação.

Figura 2-28. Escolha de distribuições



2.6.2. Instalando a Coleção de Ports

Após escolher a distribuição desejada, é apresentada a oportunidade de instalar a "Coleção de *Ports* do FreeBSD". A coleção de *ports* é uma maneira prática e conveniente de instalar programas. A coleção de *ports* não contém os códigos fontes necessários para compilar os programas. Ao invés disto, uma coleção de arquivos

automatiza o *download*, compilação e instalação e pacotes de programas de terceiros. O [Capítulo 4](#) discute a utilização da coleção de *ports* .

O programa de instalação não verifica se você tem espaço o bastante para instalar a coleção de *ports*. Selecione esta opção apenas se você realmente tiver espaço o bastante. Por volta do FreeBSD 5.3, A Coleção de *Ports* do FreeBSD requer até 300 MB de espaço em disco. A cada nova versão do FreeBSD um valor maior pode seguramente ser assumido.

```
User Confirmation Requested
Would you like to install the FreeBSD ports collection?

This will give you ready access to over 11,800 ported software
packages,
at a cost of around 300 MB of disk space when "clean" and possibly
much
more than that if a lot of the distribution tarballs are loaded
(unless you have the extra CDs from a FreeBSD CD/DVD distribution
available and can mount it on /cdrom, in which case this is far less
of a problem).

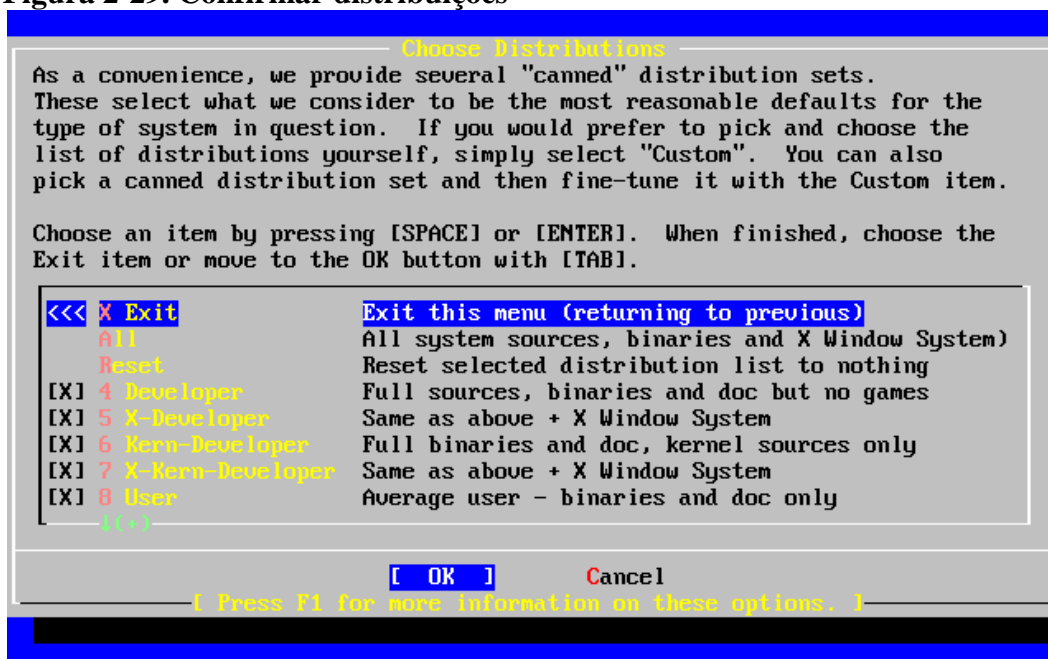
The ports collection is a very valuable resource and well worth
having
on your /usr partition, so it is advisable to say Yes to this option.

For more information on the ports collection & the latest ports,
visit:
http://www.FreeBSD.org/ports

[ Yes ]      No
```

Selecione [Yes] com as teclas indicativas para instalar a coleção de *ports* ou [No] para pular esta opção. Aperte **Enter** para continuar. O menu de seleção de distribuição aparecerá novamente.

Figura 2-29. Confirmar distribuições



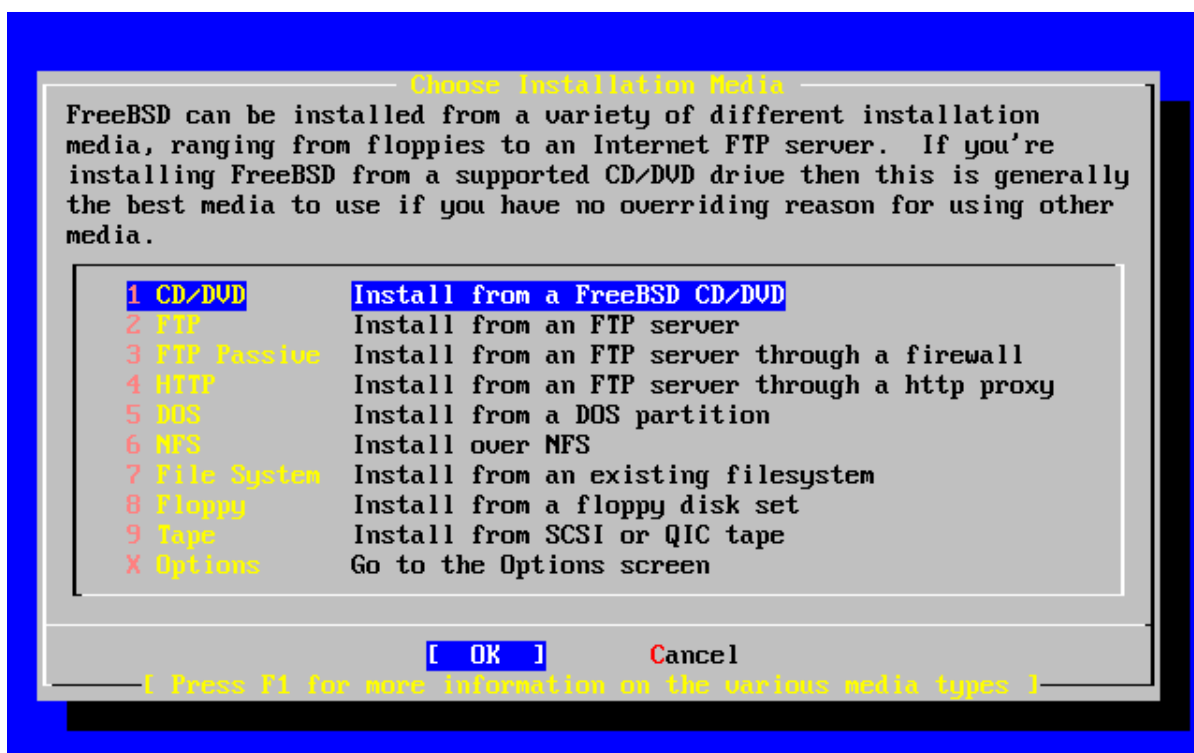
2.7. Escolhendo a mídia de instalação

Se a instalação estiver sendo realizada à partir de CDROM ou DVD use as setas indicativas para destacar a opção Install from a FreeBSD CD/DVD. Garanta que a opção [OK] está marcada e aperte **Enter** para continuar com o processo de instalação.

Para outros métodos de instalação, selecione a opção apropriada e siga as instruções.

Aperte **F1** para visualizar a ajuda instantânea sobre as mídias de instalação. Aperte **Enter** para voltar ao menu de seleção de mídias.

Figura 2-30. Escolha a mídia de instalação



Modos de instalação FTP: Existem três modos de instalação FTP que você pode escolher: FTP ativo, FTP passivo e FTP via proxy.

FTP Ativo: Instalação à partir de um servidor FTP

Esta opção fará toda transferência FTP ser realizada em modo "ativo". Não funcionará atrás de um *firewall*, mas frequentemente funcionará com servidores FTP antigos, que não suportam modo passivo. Se sua conexão travar em modo passivo (o padrão), tente modo ativo!

FTP Passivo: Instalação à partir de um servidor FTP através de um *firewall*

Essa opção instrui o **sysinstall** a utilizar modo "Passivo" para todas operações FTP. Isso permite que o usuário passe através de *firewall* que não permite conexões entrantes em portas TCP randômicas.

FTP através de proxy HTTP: Instalação à partir de um servidor FTP através de um proxy HTTP

Essa opção instrui o **sysinstall** à utilizar um protocolo HTTP (como se fosse um navegador) para se conectar em um *proxy* e realizar operações FTP. O *proxy* traduzirá as solicitações e as encaminhará ao servidor FTP. Isso permite que usuários passem por um *firewall* que não permite FTP, mas oferece um *proxy* HTTP. Neste caso é necessário especificar o *proxy* além do servidor FTP.

Para o servidor FTP normalmente basta definir o nome do servidor desejado como parte do nome de usuário, após um caract er ``@". O servidor *proxy* ``simula" o servidor real. Por exemplo, vamos assumir que voc e deseja instalar o sistema à partir de `ftp.FreeBSD.org`, usando o *proxy* FTP no servidor `algun.exemplo.com`, que escuta na porta 1024.

Nesse caso, voc e vai no menu de op oes, define o usu rio FTP como sendo `ftp@ftp.FreeBSD.org`, e define seu endere o de correio eletr nico como sendo a senha. Como m dia de instala o, especifique FTP (ou FTP passivo, se o *proxy* o suportar), e a URL `ftp://foo.example.com:1234/pub/FreeBSD`.

Partindo do princ pio que `/pub/FreeBSD` em `ftp.FreeBSD.org`   acessado via procura o (*proxy*), voc e poder  instalar o sistema à partir *daquela* m quina (que ir  acessar os arquivos de `ftp.FreeBSD.org` conforme seu processo de instala o os requisita).

2.8. Efetivando a instala o

Se desejado, a instala o pode continuar agora. Esta tamb m   a  ltima chance que voc e tem para abortar a instala o e prevenir qualquer modifica o no disco r gido.

```
User Confirmation Requested
Last Chance!  Are you SURE you want to continue the installation?

If you're running this on a disk with data you wish to save then WE
STRONGLY ENCOURAGE YOU TO MAKE PROPER BACKUPS before proceeding!

We can take no responsibility for lost disk contents!

[ Yes ]    No
```

Selecione [Yes] e aperte **Enter** para continuar.

O tempo de instala o ir  variar de acordo com o conjunto de distribui o escolhido, o tipo de m dia e a velocidade do computador. Haver  uma s rie de mensagens sendo apresentadas na tela indicando o estado do processo.

A instala o estar  terminada quando a seguinte mensagem for apresentada:

Message

Congratulations! You now have FreeBSD installed on your system.

We will now move on to the final configuration questions.
For any option you do not wish to configure, simply select No.

If you wish to re-enter this utility after the system is up, you may do so by typing: /stand/sysinstall .

[OK]

[Press enter to continue]

Aperte **Enter** para seguir para as configurações de pós instalação.

Selecionando [No] e apertando **Enter** irá abortar a instalação, de forma que modificação alguma será realizada em seu sistema. A mensagem à seguir aparecerá:

Message

Installation complete with some errors. You may wish to scroll through the debugging messages on VTY1 with the scroll-lock feature. You can also choose "No" at the next prompt and go back into the installation menus to retry whichever operations have failed.

[OK]

Essa mensagem é apresentada porque nada foi instalado. Apertar **Enter** fará o programa

2.9. Pós-Instalação

Após a instalação bem sucedida do sistema operacional, é iniciada a configuração de uma série de opções. Cada opção pode ser configurado revisitando as opções de configuração antes de iniciar o recém instalado sistema FreeBSD, ou com o sistema já instalado chamando o comando /stand/sysinstall e selecionando a guia Configure.

2.9.1. Configuração dos dispositivos de rede

Caso tenha configurado PPP para instalação FTP anteriormente, esta tela não estará disponível, mas pode ser configurada posteriormente como será descrito.

Para obter informações detalhadas sobre Redes Locais, e aobre como configurar o FreeBSD para agir como roteador/*gateway*, refira-se ao capítulo de [Advanced Networking](#).

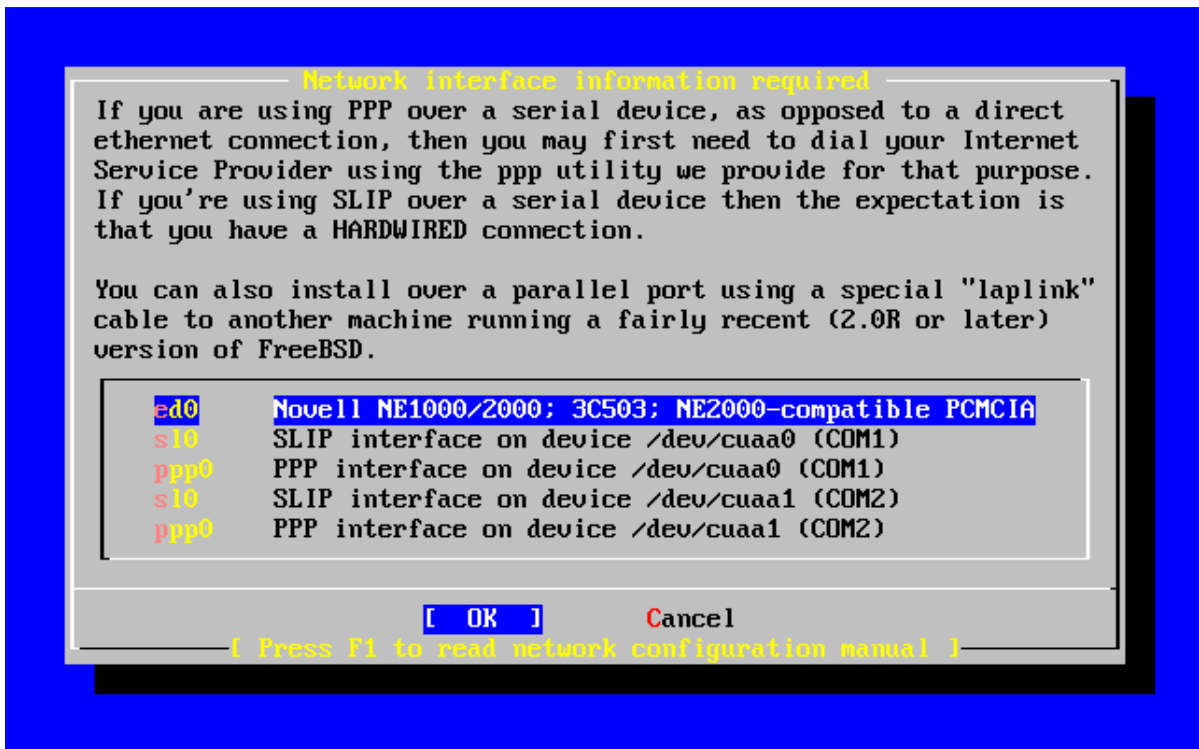
User Confirmation Requested

Would you like to configure any Ethernet or SLIP/PPP network devices?

[Yes] No

Para configurar um dispositivo de rede, selecione a opção [Yes] e aperte **Enter**. Ou então selecione [No] para continuar.

Figura 2-31. Selecionando dispositivos de rede



Selecione a interface à ser configurado com as teclas indicativas e aperte **Enter**.

```
User Confirmation Requested
Do you want to try IPv6 configuration of the interface?

Yes   [ No ]
```

Nesta rede local privada, o protocolo de endereçamento de Internet atual (IPv4) é suficiente, e selecionar a opção [No] e apertar **Enter** satisfaz a necessidade.

Caso esteja conectado à uma rede IPv6 com um servidor RA, selecione a opção [Yes] e aperte **Enter**. Levará alguns segundos a busca por servidores RA.

```
User Confirmation Requested
Do you want to try DHCP configuration of the interface?

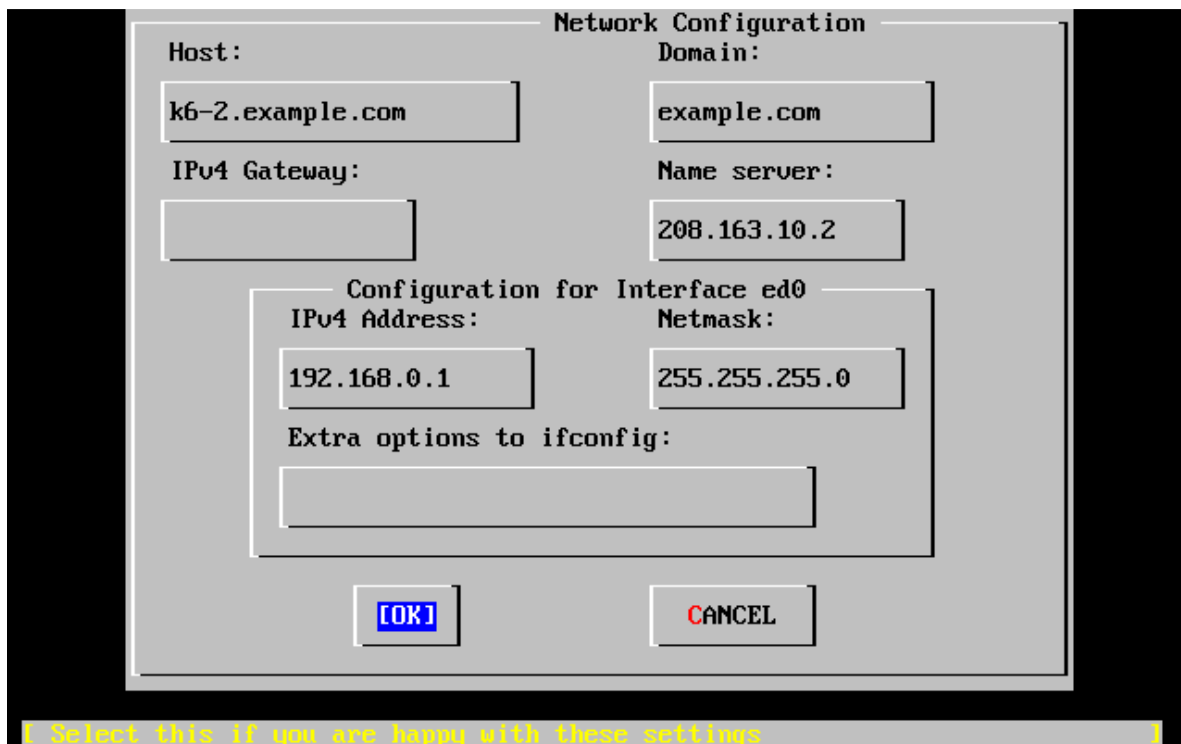
Yes   [ No ]
```

Caso DHCP (Protocolo de Configuração Dinâmica de Estações) não seja necessário, selecione a opção [No] com as teclas indicativas e aperte **Enter**.

Caso selecione [Yes] o programa **dhclient** será executado, e se a busca pelo servidor DHCP tiver sucesso, a configuração da rede será automaticamente preenchida. Refira-se à [Seção 19.10](#) para obter maiores informações.

A próxima tela de configuração da rede apresenta a configuração do dispositivo de rede em um sistema que fará papel de servidor de ligação (*gateway*) em uma rede local.

Figura 2-32. Ajustando configuração de rede para ed0



Utilize a tecla **Tab** para selecionar os campos de informações e preencha os dados de forma apropriada:

Host

O nome completamente qualificado da estação, como por exemplo `k6-2.example.com` neste caso.

Domínio

O nome do domínio onde sua estação está localizada, como `example.com` neste caso.

Gateway IPv4

Endereço IP da estação que encaminhará pacotes para um destino não local. É necessário preencher este campo se a máquina em questão for um cliente de rede. *Mantenha este campo em branco* se a própria estação for o *gateway* para a Internet, à partir de sua rede. O *gateway* IPv4 também é chamado de *gateway* padrão ou rota padrão.

Servidor de nomes

Endereço IP de seu servidor DNS local. Não há servidor de nomes nesta rede local, de forma que o endereço IP do servidor DNS do provedor (208.163.10.2) foi utilizado.

Endereço IPv4

O endereço IP desta interface é 192.168.0.1

Netmask

O bloco de endereçamento utilizado nesta rede é um bloco de classe C (192.168.0.0 - 192.168.255.255). A máscara de rede padrão para classe C é (255.255.255.0).

Opções extra para o ifconfig

Qualquer opção específica de uma interface de rede que possa ser definida com `ifconfig` pode ser adicionada aqui. Neste caso não havia nenhuma opção necessária.

Use a tecla **Tab** para selecionar [OK] quando terminar, e aperte **Enter**.

```
User Confirmation Requested
Would you like to Bring Up the ed0 interface right now?

[ Yes ] No
```

Escolhendo [Yes] e apertando **Enter** a máquina estará colocada na rede e pronta para uso. Entretanto, esta opção só estará plenamente operacional após um *reboot*.

2.9.2. Configuração do *gateway*

```
User Confirmation Requested
Do you want this machine to function as a network gateway?

[ Yes ] No
```

Se a máquina em questão for atuar como *gateway* para a rede local e encaminhar pacotes entre outras redes, escolha [Yes] e aperte **Enter**. Se a máquina for apenas um nó da rede, escolha [No] e aperte **Enter** para continuar.

2.9.3. Configuração de Serviços de Internet

```
User Confirmation Requested
Do you want to configure inetd and the network services that it
provides?

Yes [ No ]
```

Se selecionar [No] vários serviços populares como **telnetd** não serão habilitados. Isso significa que usuários remotos não poderão acessar esta máquina via **telnet**. Os usuários locais continuarão podendo acessar estações remotas via **telnet**.

Estes serviços podem ser habilitados após a instalação, editando o arquivo `/etc/inetd.conf` com seu editor de texto favorito. Veja [Seção 19.14.1](#) para obter mais informações.

Selecione [Yes] caso deseje configurar estes serviços durante a instalação. Uma informação adicional será apresentada:

User Confirmation Requested

The Internet Super Server (inetd) allows a number of simple Internet services to be enabled, including finger, ftp and telnetd. Enabling these services may increase risk of security problems by increasing the exposure of your system.

With this in mind, do you wish to enable inetd?

[Yes] No

Selecione [Yes] para continuar.

User Confirmation Requested

inetd(8) relies on its configuration file, `/etc/inetd.conf`, to determine which of its Internet services will be available. The default FreeBSD `inetd.conf(5)` leaves all services disabled by default, so they must be specifically enabled in the configuration file before they will function, even once `inetd(8)` is enabled. Note that services for IPv6 must be separately enabled from IPv4 services.

Select [Yes] now to invoke an editor on `/etc/inetd.conf`, or [No] to use the current settings.

[Yes] No

Ao escolher [Yes] você poderá adicionar serviços que serão automaticamente iniciados, apagando os caracteres # ao início de cada linha.

Figura 2-33. Editando o arquivo `inetd.conf`

```
^[ (escape) menu    ^y search prompt    ^k delete line      ^p prev li          ^g prev page
^o ascii code       ^x search            ^l undelete line    ^n next li          ^v next page
^u end of file      ^a begin of line    ^w delete word      ^b back 1 char
^t begin of file    ^e end of line      ^r restore word     ^f forward 1 char
^c command          ^d delete char      ^j undelete char    ^z next word
L: 1 C: 1 =====
Sep 22 23:42:04 k6-2 login: ROOT LOGIN (root) ON ttyv121:13:33 obrien Exp $
#
# Internet server configuration database
#
# Define *both* IPv4 and IPv6 entries for dual-stack support.
# To disable a service, comment it out by prefixing the line with '#'.
# To enable a service, remove the '#' at the beginning of the line.
#
#ftp    stream  tcp    nowait  root    /usr/libexec/lukemftpd  ftpd -l -r
#ftp    stream  tcp    nowait  root    /usr/libexec/ftpd       ftpd -l
#ftp    stream  tcp6   nowait  root    /usr/libexec/ftpd       ftpd -l
#telnet stream  tcp    nowait  root    /usr/libexec/telnetd   telnetd
#telnet stream  tcp6   nowait  root    /usr/libexec/telnetd   telnetd
#shell  stream  tcp    nowait  root    /usr/libexec/rshd      rshd
#shell  stream  tcp6   nowait  root    /usr/libexec/rshd      rshd
#login  stream  tcp    nowait  root    /usr/libexec/rlogind   rlogind
#login  stream  tcp6   nowait  root    /usr/libexec/rlogind   rlogind
#finger stream  tcp    nowait/3/10 nobody /usr/libexec/fingerd   fingerd -s
file "/etc/inetd.conf", 119 lines
```

Após adicionar os serviços desejados, aperte **Esc** para visualizar um novo menu que permite sair do editor e salvar as modificações.

2.9.4. FTP Anônimo

```
                User Confirmation Requested
Do you want to have anonymous FTP access to this machine?

                Yes      [ No ]
```

2.9.4.1. Negar FTP anônimo

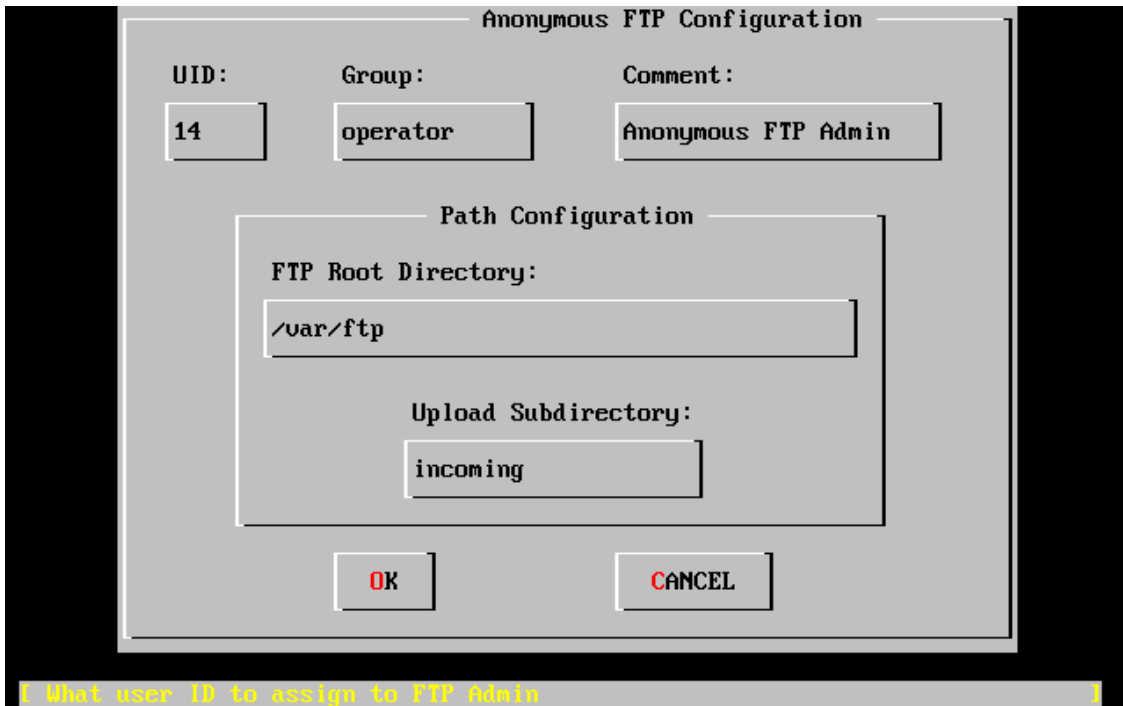
Selecionar a opção padrão, [No] e apertar **Enter** ainda permitirá que usuários tenham contas autenticadas por senhas e utilizar FTP para acessar a estação.

2.9.4.2. Permitir FTP anônimo

Qualquer pessoa poderá acessar a estação, se você optar por permitir conexões FTP anônimas. As implicações de segurança devem ser consideradas antes de habilitar esta opção. Para mais informações sobre segurança, veja [Capítulo 10](#).

Para permitir FTP anônimo, use as flechas indicativas para selecionar [Yes] e aperte **Enter**. A tela seguinte (ou similar) irá apresentar:

Figura 2-34. Configuração padrão para FTP anônimo



Apertar **F1** apresentará a tela de ajuda:

This screen allows you to configure the anonymous FTP user.

The following configuration values are editable:

UID: The user ID you wish to assign to the anonymous FTP user.
All files uploaded will be owned by this ID.

Group: Which group you wish the anonymous FTP user to be in.

Comment: String describing this user in /etc/passwd

FTP Root Directory:

Where files available for anonymous FTP will be kept.

Upload subdirectory:

Where files uploaded by anonymous FTP users will go.

O diretório de FTP raiz será colocado sob /var por padrão. Caso você ainda não tenha bastante experiência, para antecipar suas necessidades de utilização do FTP, o diretório /usr pode ser utilizado como base para o Diretório Raiz de FTP, ficando /usr/ftp.

Quando estiver satisfeito com os valores, aperte **Enter** para continuar.

```
User Confirmation Requested
Create a welcome message file for anonymous FTP users?
```

```
[ Yes ] No
```

Ao selecionar [Yes] e apertar **Enter**, um editor de texto será automaticamente iniciado, permitindo que você edite a mensagem de boas-vindas ao FTP.

Figura 2-35. Editando a mensagem de boas-vindas do FTP

```
^_ [ (escape) menu ^y search prompt ^k delete line ^p prev line ^g prev page
^o ascii code ^x search ^l undelete line ^n next line ^v next page
^u end of file ^a begin of line ^w delete word ^b back char ^z next word
^t begin of file ^e end of line ^r restore word ^f forward char
^c command ^d delete char ^j undelete char ESC-Enter: exit
=====
Your welcome message here.
=====
file "/var/ftp/etc/ftpmotd", 1 lines, read only
```

Este é o editor de texto `ee`. Use as instruções para modificar a mensagem ou modifique-a posteriormente com o editor de sua escolha. Preste atenção no nome/localização do arquivo, na base da tela do editor de textos.

Aperte **Esc** e um menu surgirá com a opção padrão definida como a) leave editor. Aperte **Enter** para sair e continuar, e aperte **Enter** novamente para salvar as alterações, caso tenha feito alguma.

2.9.5. Sistema de arquivos de rede

O sistema de arquivos de rede (*Network File System* - NFS) permite compartilhar arquivos através de uma rede. Uma estação pode ser configurada como servidor, como cliente, ou ambos. Por gentileza, refira-se à [Seção 19.6](#) para obter mais informações.

2.9.5.1. Servidor NFS

```
User Confirmation Requested
Do you want to configure this machine as an NFS server?

Yes [ No ]
```

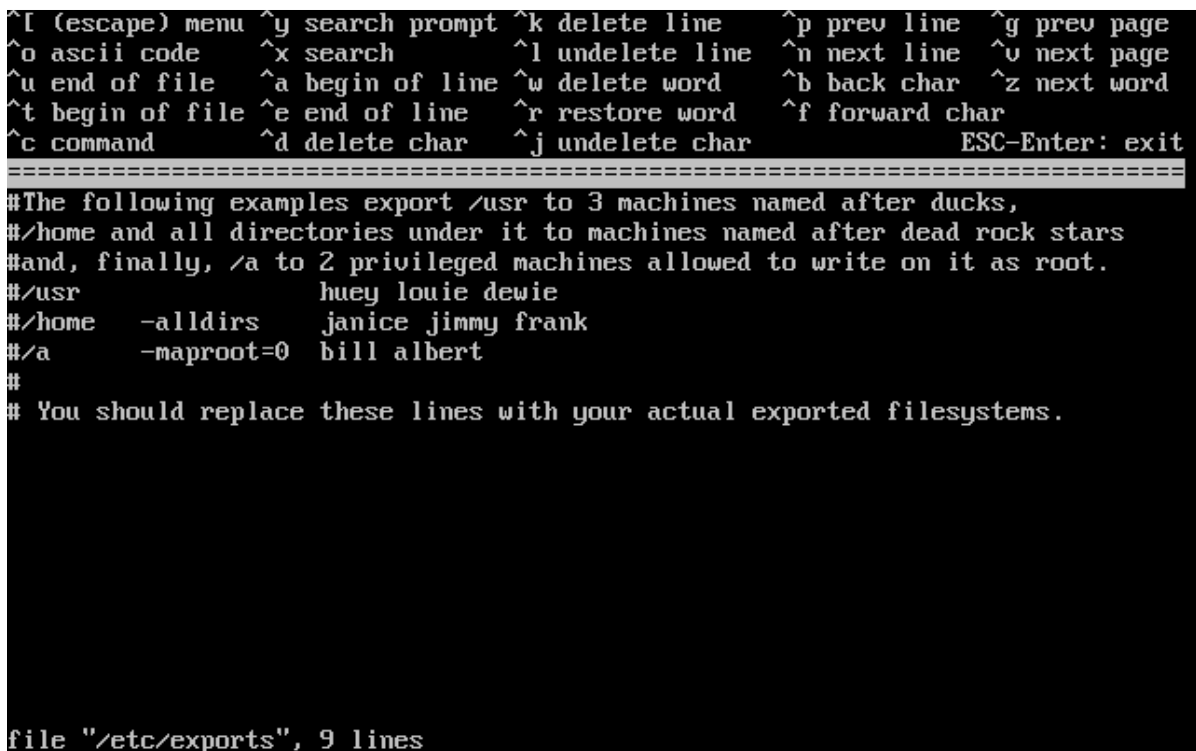
Se não houver necessidades de um servidor de sistema de arquivos de rede, selecione [No] e aperte **Enter**.

Caso [Yes] seja escolhido, uma mensagem será apresentada, indicando que o arquivo `exports` deve ser criado.

```
Message
Operating as an NFS server means that you must first configure an
/etc/exports file to indicate which hosts are allowed certain kinds of
access to your local filesystems.
Press [Enter] now to invoke an editor on /etc/exports
      [ OK ]
```

Aperte **Enter** para continuar. Um editor de texto será iniciado, permitindo a criação do arquivo `exports` bem como sua edição.

Figura 2-36. Editando o arquivo `exports`



```
^_ [ (escape) menu ^y search prompt ^k delete line ^p prev line ^g prev page
^o ascii code ^x search ^l undelete line ^n next line ^v next page
^u end of file ^a begin of line ^w delete word ^b back char ^z next word
^t begin of file ^e end of line ^r restore word ^f forward char
^c command ^d delete char ^j undelete char ESC-Enter: exit
=====
#The following examples export /usr to 3 machines named after ducks,
#/home and all directories under it to machines named after dead rock stars
#and, finally, /a to 2 privileged machines allowed to write on it as root.
#/usr          huey louie dewie
#/home -alldirs  janice jimmy frank
#/a          -maproot=0  bill albert
#
# You should replace these lines with your actual exported filesystems.

file "/etc/exports", 9 lines
```

Utilize as instruções em questão para adicionar os sistemas de arquivos exportados no momento, ou posteriormente, com um editor de texto de sua escolha. Note a localização /nome do arquivo na base da tela do editor de textos.

Aperte **Esc** e um menu irá surgir na opção padrão, a) leave editor. Aperte **Enter** para sair e continuar.

2.9.5.2. Cliente NFS

O cliente NFS permite que sua estação acesse servidores NFS.

```
User Confirmation Requested
Do you want to configure this machine as an NFS client?

Yes   [ No ]
```

Com as teclas de setas, escolha [Yes] ou [No] como preferir, e aperte **Enter**.

2.9.6. Perfil de Segurança

Um "Perfil de Segurança" (*security profile*) é um conjunto de opções de segurança que tem o objetivo de tentar atingir o índice desejado de segurança e conveniência, iniciando e desligando certos programas e configurações. Quanto maior o perfil de segurança, menos programas serão iniciados por padrão. Este é um dos princípios básicos de segurança: não execute nada a não ser o necessário.

Por gentileza, perceba que o perfil de segurança não passa de uma configuração padrão. Todos programas podem ser habilitados e desabilitados após a instalação, bastando adicionar as linhas adequadas ao arquivo `/etc/rc.conf`. Para obter mais informações, por gentileza refira-se à página de manual [rc.conf\(5\)](#).

A tabela a seguir descreve o que cada perfil de segurança faz. As colunas são as escolhas para seus perfís, e as linhas são os programas ou funções que o perfil habilita ou desabilita.

Tabela 2-4. Perfís de segurança disponíveis

	Extremo	Moderado
sendmail(8)	NÃO	SIM
sshd(8)	NÃO	SIM
portmap(8)	NÃO	TALVEZ [a]
Servidor NFS	NÃO	SIM
securelevel(8)	SIM [b]	NO

Notas:

- A opção é habilitada apenas se a estação foi configurada como servidor ou cliente NFS anteriormente durante a instalação.
- Se você escolher um perfil de segurança que define o nível de segurança (*securelevel*), em "Alto (High)" ou "Extremo (Extreme)" deve ficar atento às implicações. Por gentileza, leia a página de manual [init\(8\)](#) e preste atenção especial aos significados dos níveis de segurança, ou você terá alguns problemas posteriormente!

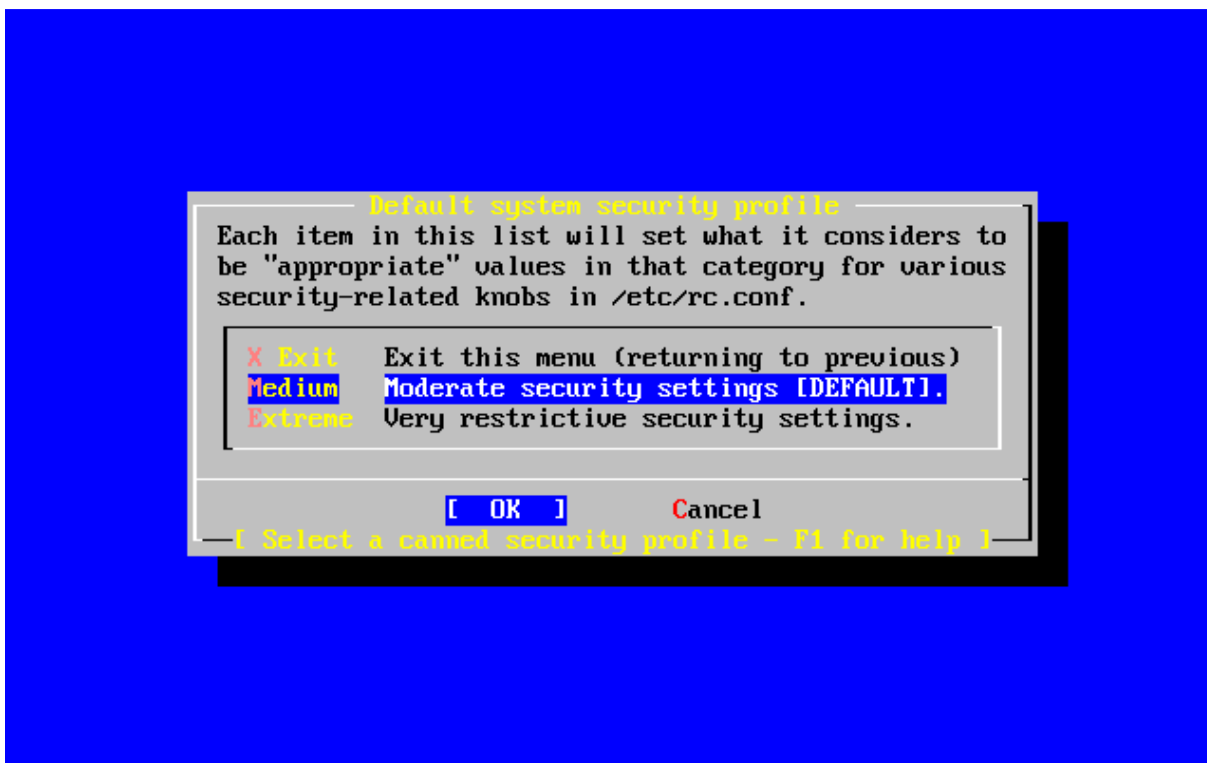
```
User Confirmation Requested
Do you want to select a default security profile for this host
(select
No for "medium" security)?

[ Yes ] No
```

Selecionar [No] e apertar **Enter** definirá o nível de segurança em médio.

Selecionar [Yes] e apertar **Enter** permitirá a seleção de um perfil de segurança diferente.

Figura 2-37. Opções do Perfil de Segurança



Aperte **F1** para apresentar a ajuda. Aperte **Enter** para voltar ao menu de seleção.

Utilize as teclas indicativas para escolher a opção média (Medium) a não ser que esteja certo de outro nível que atenda suas necessidades. Com o [OK] destacado, aperte **Enter**.

Uma mensagem de confirmação apropriada será apresentada, dependendo de que definição de segurança fora escolhida.

Message

Moderate security settings have been selected.

Sendmail and SSHd have been enabled, securelevels are disabled, and NFS server setting have been left intact. PLEASE NOTE that this still does not save you from having to properly secure your system in other ways or exercise due diligence in your administration, this simply picks a standard set of out-of-box defaults to start with.

To change any of these settings later, edit /etc/rc.conf

[OK]

Message

Extreme security settings have been selected.

Sendmail, SSHd, and NFS services have been disabled, and securelevels have been enabled.

PLEASE NOTE that this still does not save you from having to properly secure your system in other ways or exercise due diligence in your administration, this simply picks a more secure set of out-of-box defaults to start with.

To change any of these settings later, edit /etc/rc.conf

[OK]

Aperte **Enter** para continuar com a configuração da pós-instalação.

AtençãoO perfil de segurança não é "Tiro e Queda"! Mesmo se as definições extremas forem utilizadas, é necessário manter-se atualizado com os problemas de segurança, lendo as [listas de discussão](#) apropriadas, utilizando boas senhas e mailing geralmente aderindo à boas práticas de segurança. A opção apenas define um índice de configuração conveniente já na instalação.

2.9.7. Configurações do console do sistema

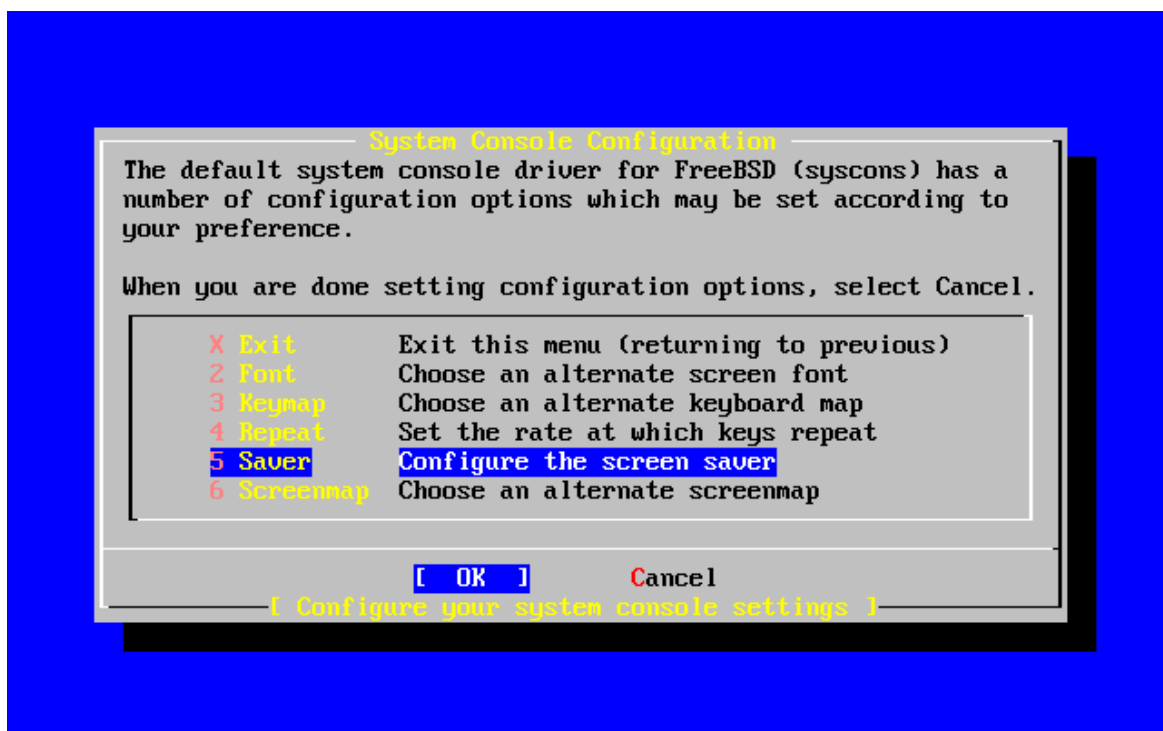
Existem várias opções disponíveis para a customização do console do sistema.

```
User Confirmation Requested
Would you like to customize your system console settings?

[ Yes ] No
```

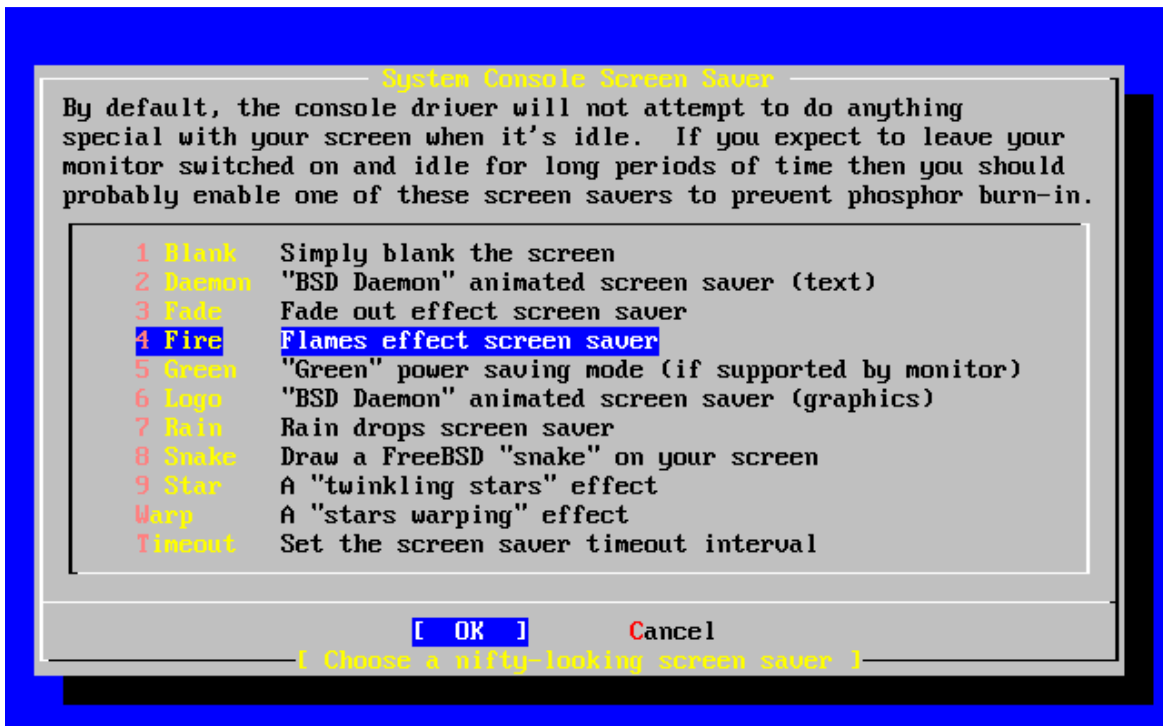
Para visualizar e configurar tais opções, selecione [Yes] e aperte **Enter**.

Figura 2-38. Opções de configuração do console do sistema



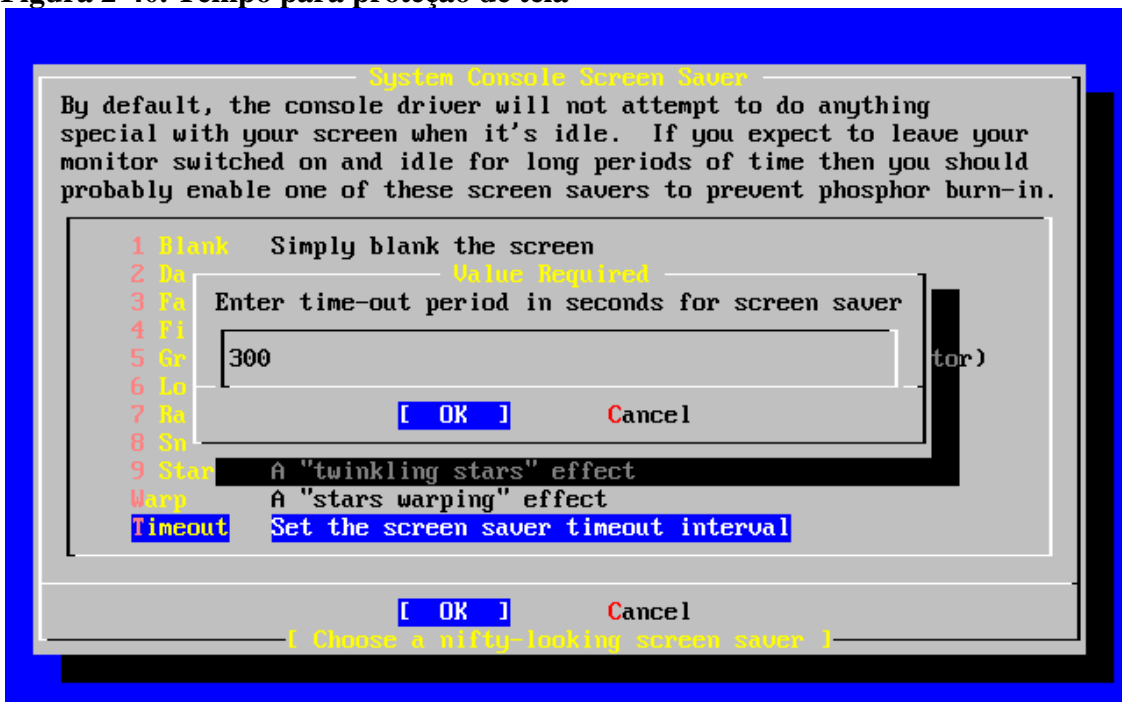
Uma das opções normalmente utilizadas é a proteção de tela. Use as setas para selecionar Saver e aperte **Enter**.

Figura 2-39. Opções da proteção de tela



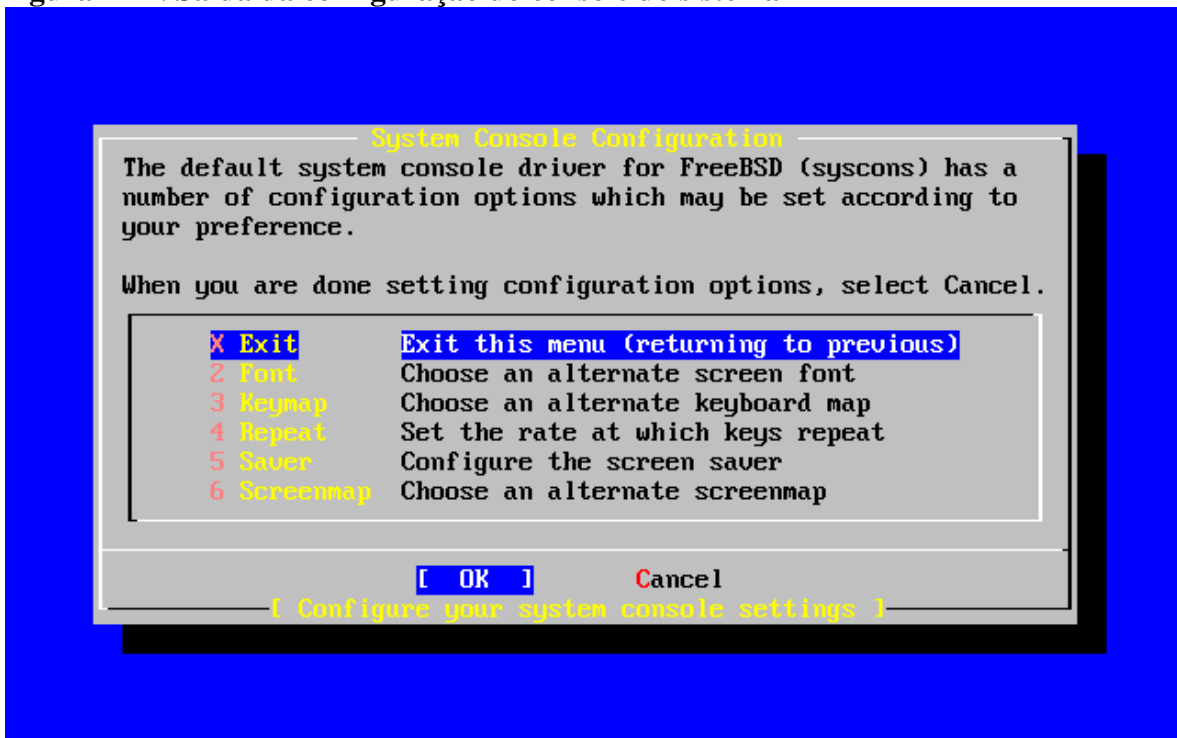
Escolha o protetor de tela desejado usando as setas, e aperte **Enter**. O menu de configuração do console do sistema irá resurgir. O intervalo temporal padrão é 300 segundos. Para modificá-lo selecione Saver novamente. No menu de opções do protetor de tela selecione Timeout e aperte **Enter**. Um menu irá surgir:

Figura 2-40. Tempo para proteção de tela



O valor pode ser modificado, em seguida aperte [OK] e **Enter** para voltar ao menu de opções do protetor de tela.

Figura 2-41. Saida da configuração do console do sistema



Selecione Exit e aperte **Enter** para continuar ao menude configuração de pós-instalação.

2.9.8. Definindo Zona Temporal (Time Zone)

Definir a zona temporal de sua estação possibilitará que ela corrija automaticamente seus horários para os padrões regionais, e realizar funções de horário de forma apropriada.

O exemplo apresentado é para uma estação localizada na faixa de horário do Leste dos Estados Unidos. Sua opção vai variar de acordo com sua localização geográfica.

```
User Confirmation Requested
Would you like to set this machine's time zone now?

[ Yes ]   No
```

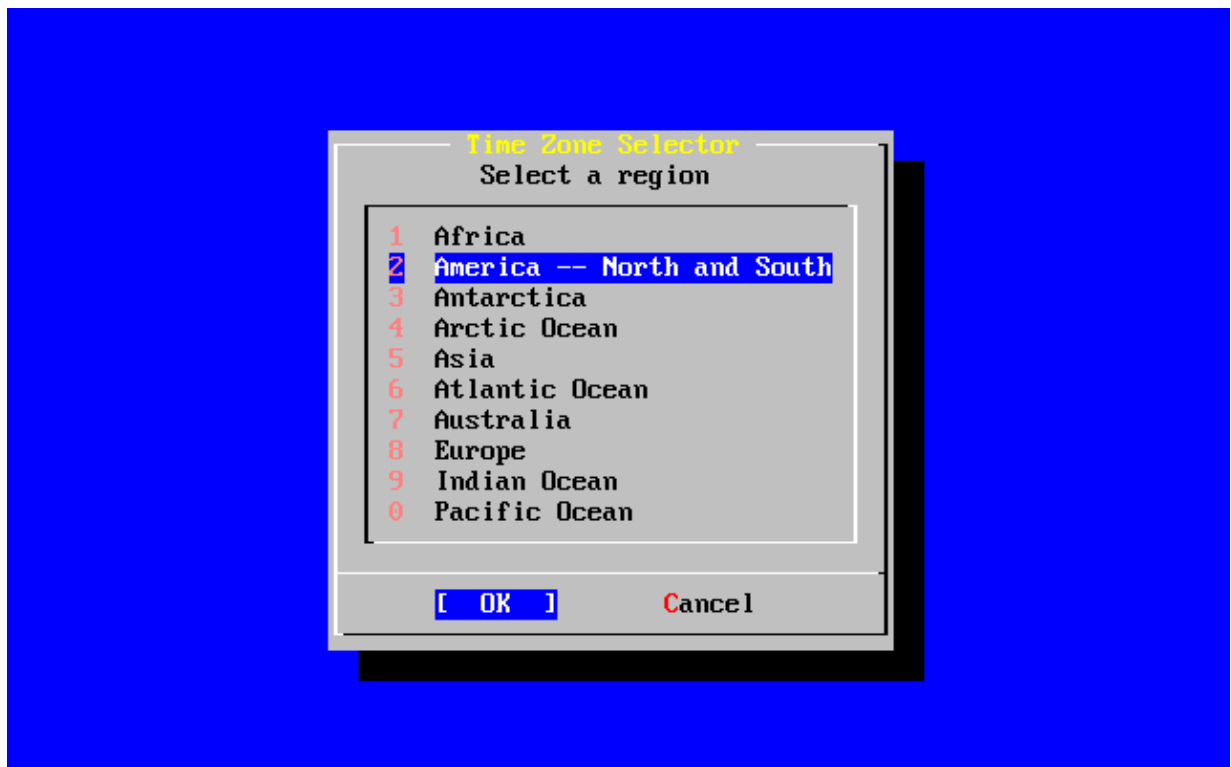
Selecione [Yes] e aperte **Enter** para selecionar o fuso-horário.

```
User Confirmation Requested
Is this machine's CMOS clock set to UTC?  If it is set to local time
or you don't know, please choose NO here!

Yes    [ No ]
```

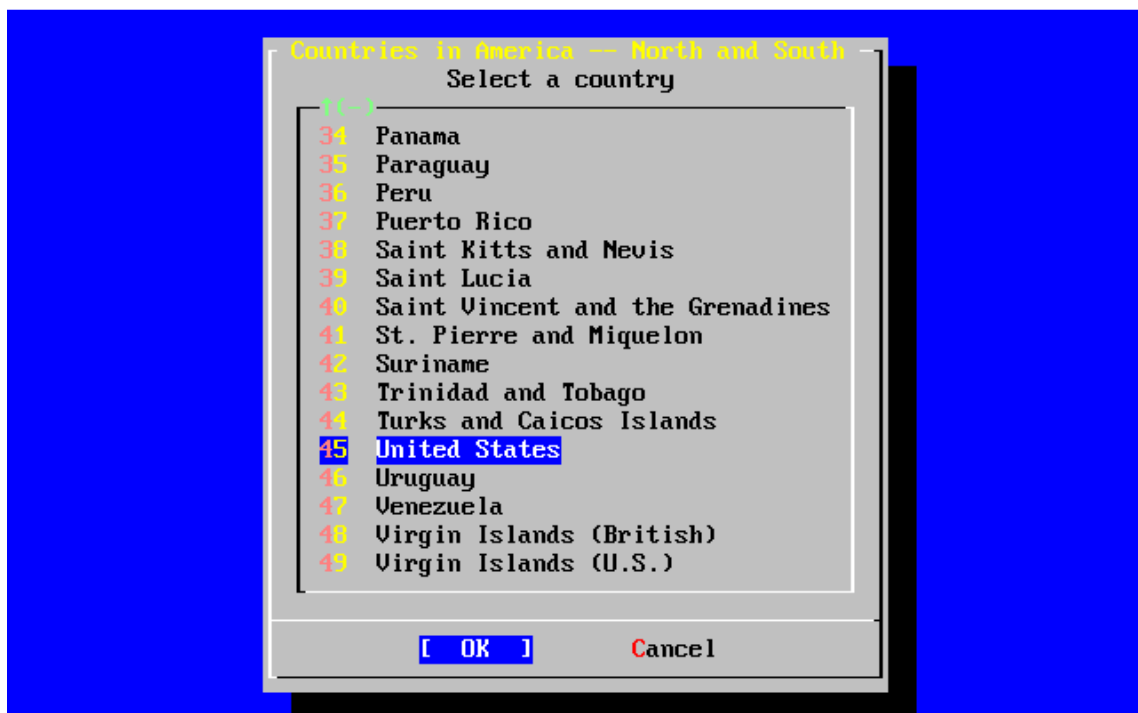
Selecione [Yes] ou [No] de acordo com a forma como o relógio a sua máquina está configurado, e aperte **Enter**.

Figura 2-42. Selecione sua região



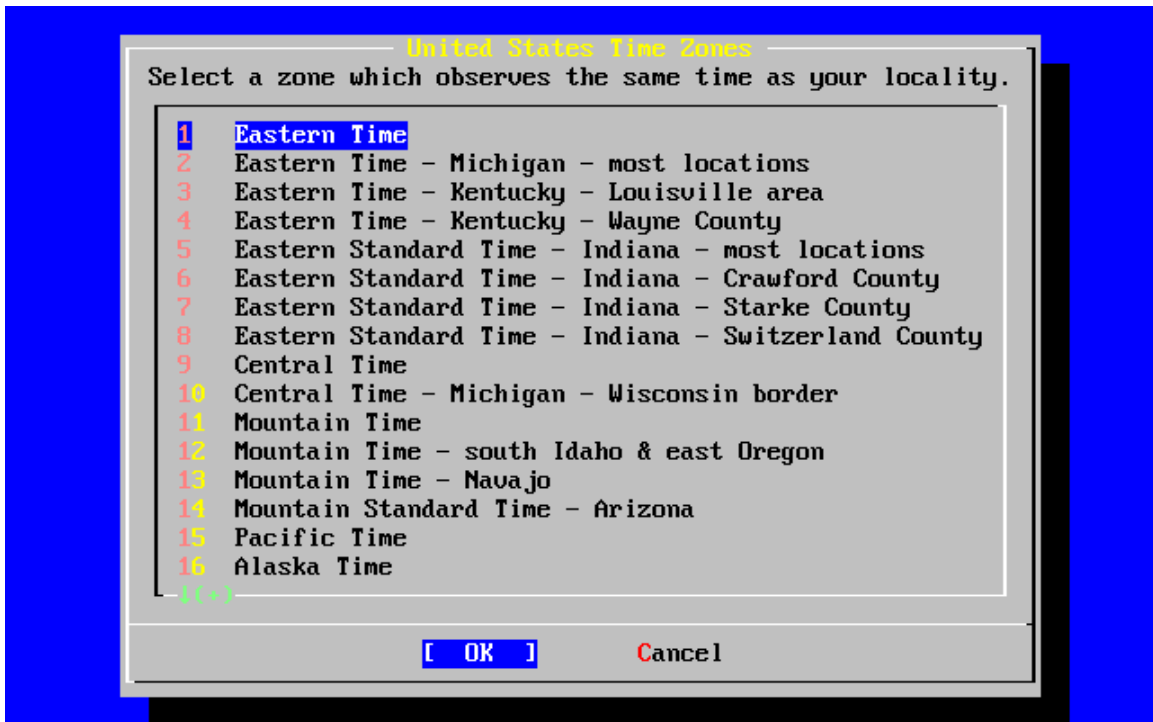
A região apropriada é selecionada usando as setas e apertando **Enter**.

Figura 2-43. Selecione seu país



Selecione o país apropriado com as teclas de setas e aperte **Enter**.

Figura 2-44. Seleção o Fuso-Horário



A zona temporal apropriada pode ser selecionada com as teclas indicativas e apertando **Enter**.

```
Confirmation
Does the abbreviation 'EDT' look reasonable?

[ Yes ] No
```

Confirme a abreviação para a zona temporal em questão se estiver correta. Aperte **Enter** para continuar com a configuração pós-instalação.

2.9.9. Compatibilidade Linux

```
User Confirmation Requested
Would you like to enable Linux binary compatibility?

[ Yes ] No
```

Selecionando [Yes] e apertando **Enter** permitirá a execução de aplicações Linux no FreeBSD. A instalação adicionará os pacotes apropriados para viabilizar a compatibilidade Linux.

Se você estiver instalando o sistema por FTP, a estação necessitará estar conectada à Internet. As vezes um sítio FTP remoto não terá todas distribuições como a compatibilidade binária Linux. Esta compatibilidade pode ser instalada posteriormente, se desejado.

2.9.10. Configurações do Mouse

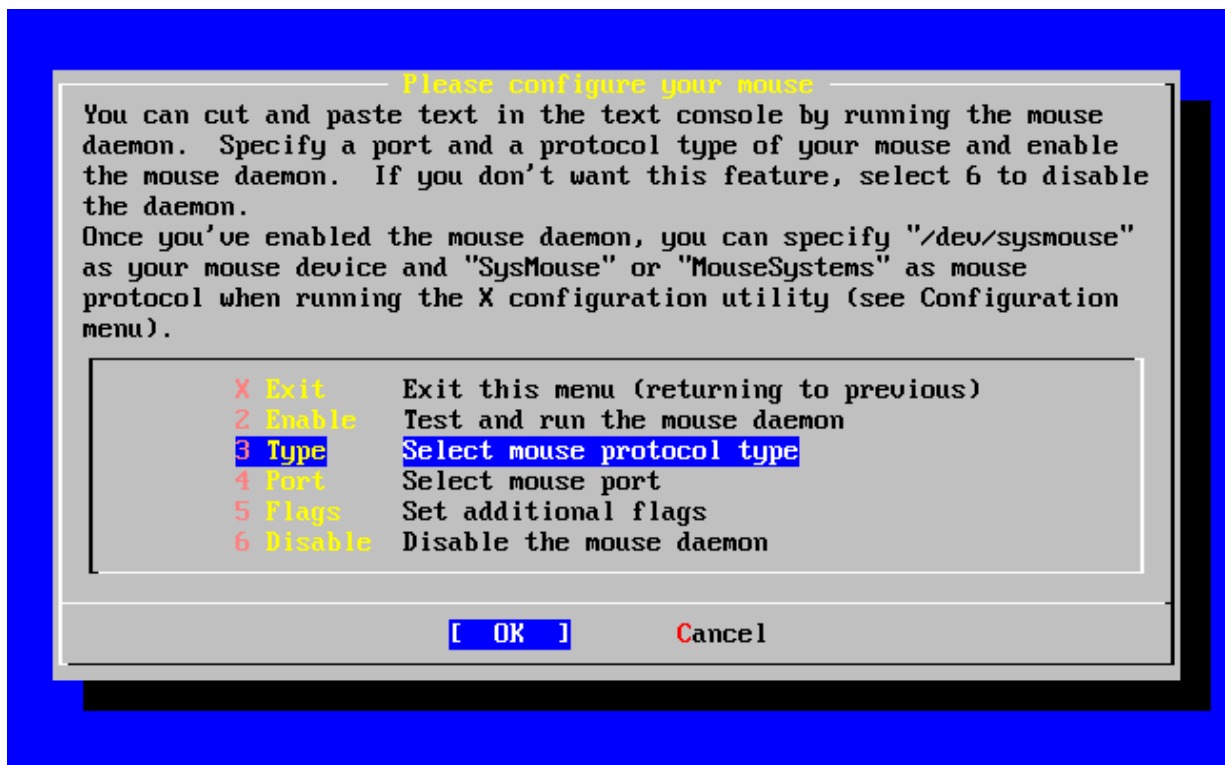
Esta opção permitirá que o usuário copie e cole texto no console e aplicações, com *mouse* de 3 botões. Se você estiver utilizando um mouse de 2 botões, refira-se à página de manual [moused\(8\)](#), após a instalação, para obter detalhes sobre como emular o terceiro botão. Este exemplo aborda um mouse não-USB (como um mouse PS/2 ou serial - COM):

```
User Confirmation Requested
Does this system have a non-USB mouse attached to it?

[ Yes ]   No
```

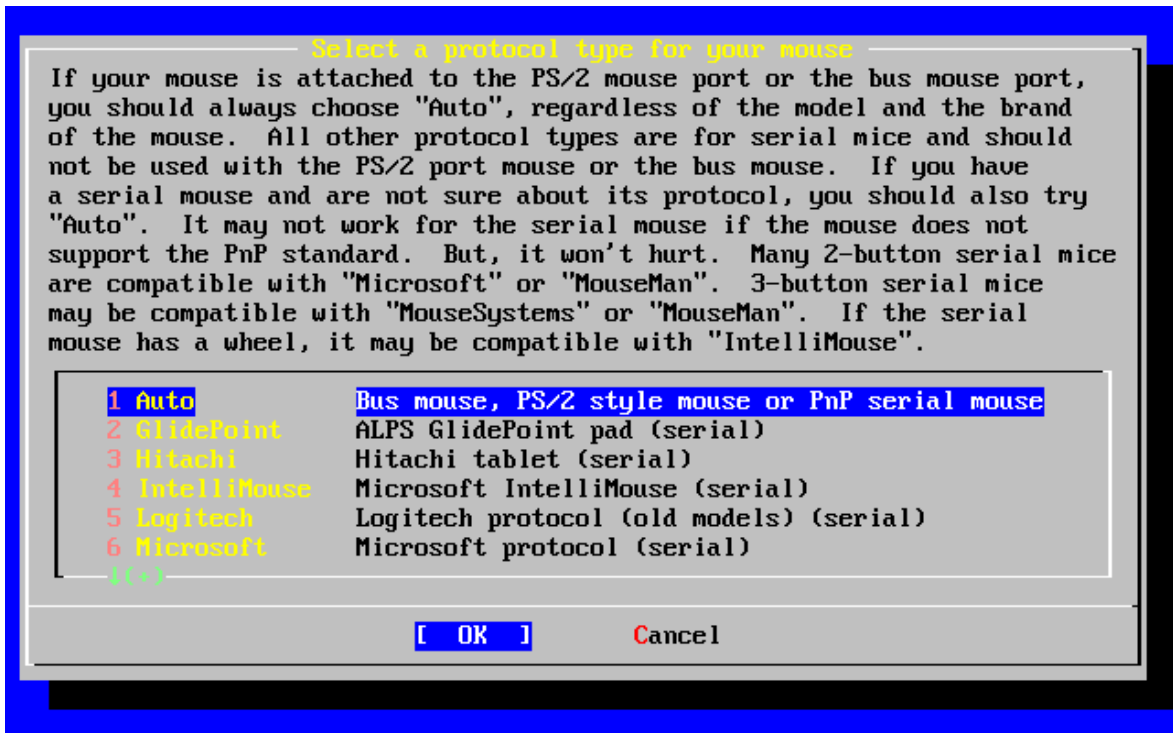
Selecione [Yes] para configurar um mouse não-USB ou [No] para um que seja USB, em seguida aperte **Enter**.

Figura 2-45. Selecione o tipo do protocolo de mouse



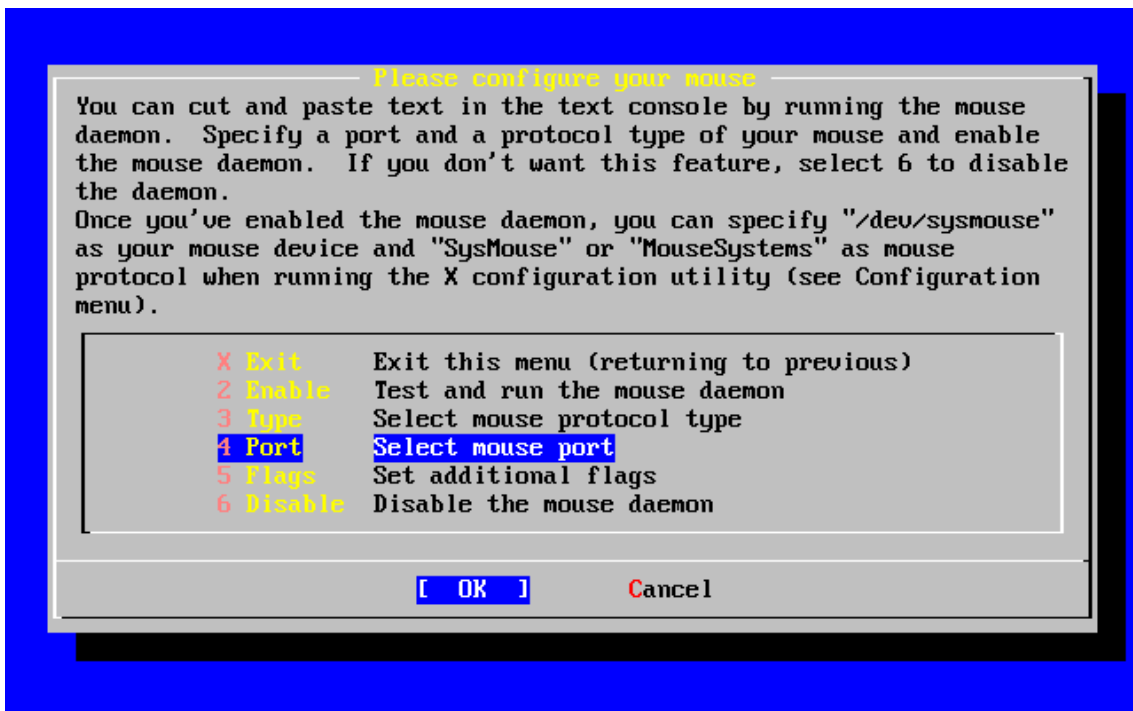
Utilize as setas para escolher Type e aperte **Enter**.

Figura 2-46. Definindo o protocolo do mouse



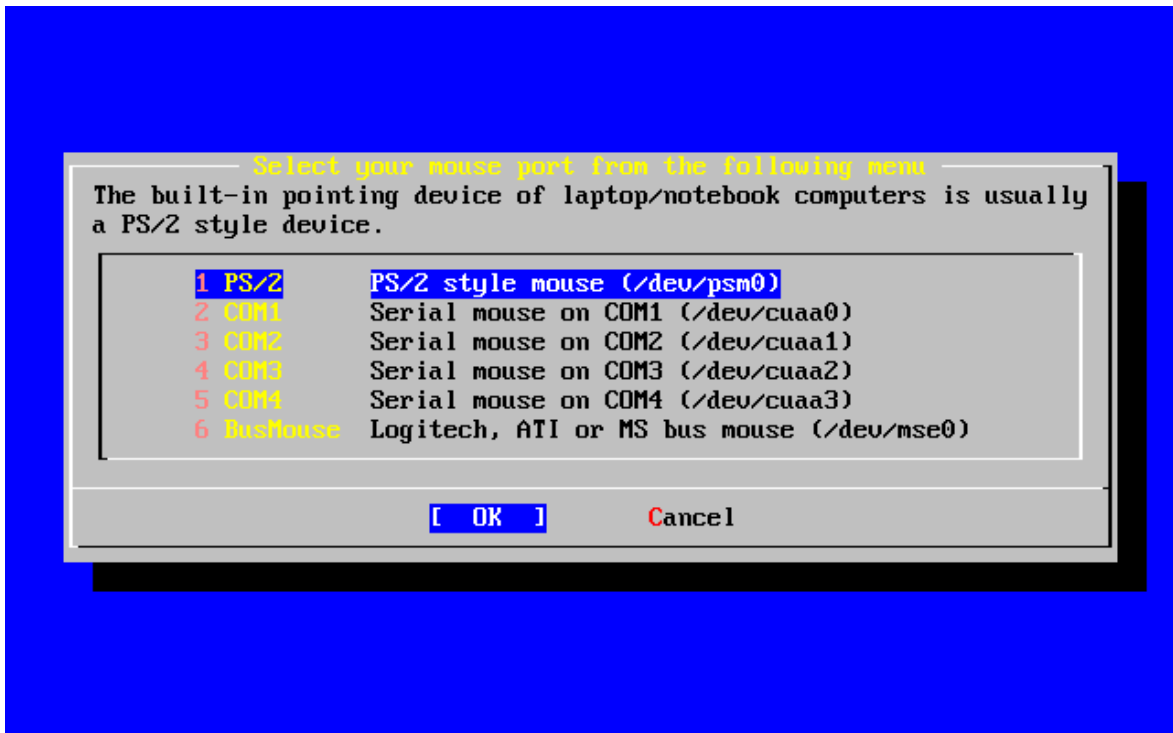
O mouse utilizado neste exemplo é um PS/2, então o padrão, Auto é bastante apropriado. Para modificar o protocolo utilize as setas para escolher outra opção. Garanta que [OK] está marcado e aperte **Enter** para sair deste menu.

Figura 2-47. Configure a porta do mouse



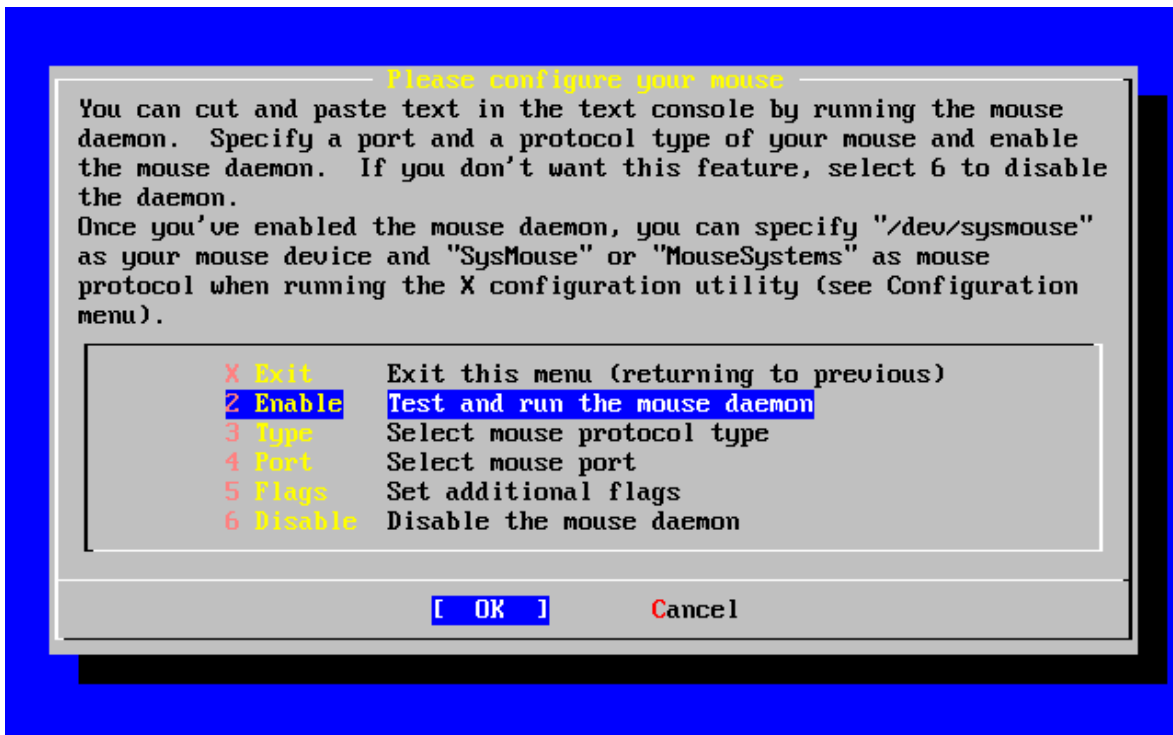
Utilize as flechinhas para escolher Port e aperte **Enter**.

Figura 2-48. Definindo a porta do mouse



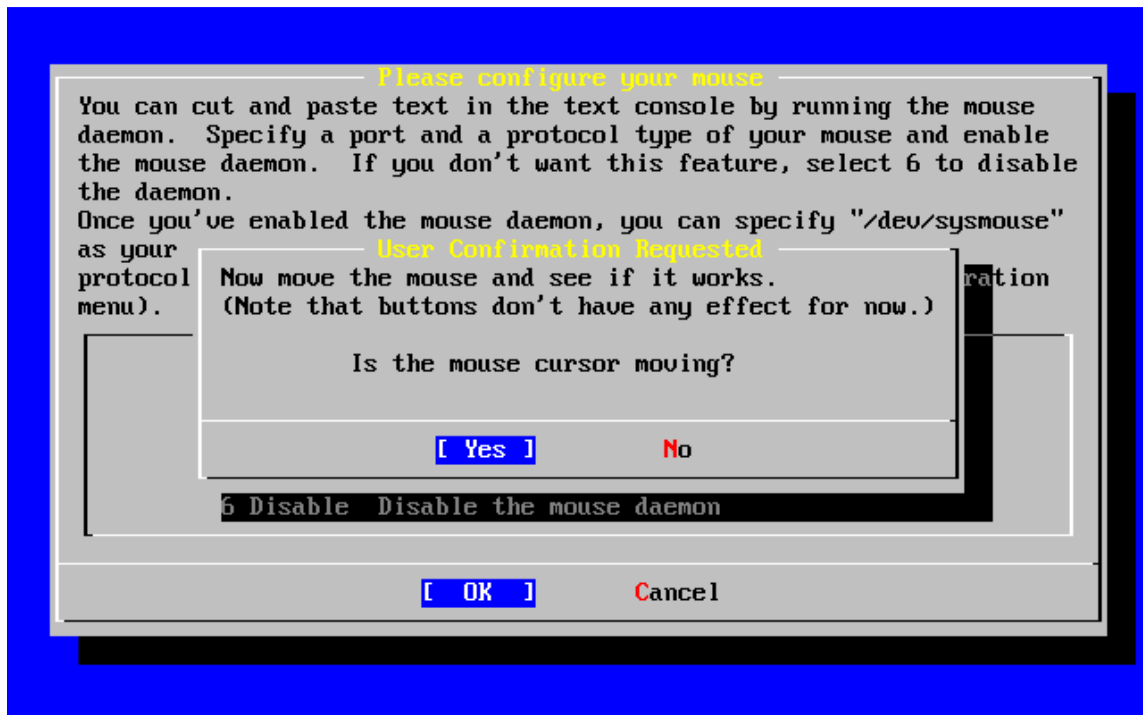
Este sistema possui um mouse PS/2, de forma que PS/2 é a opção mais apropriada. Modifique a porta com as setas e aperte **Enter** caso necessário.

Figura 2-49. Habilite o *daemon* do mouse



Por último, use as setas e escolha Enable, em seguida aperte **Enter** para habilitar e testar o *daemon* do mouse.

Figura 2-50. Teste o *daemon* do mouse



Movimente o mouse pela tela e verifique se o cursor responde apropriadamente. Se estiver funcional, selecione Yes e tecle **Enter**. Do contrário, o mouse não está corretamente configurado -- selecione No e tente utilizar outras opções de configuração.

Selecione Exit com as setas e aperte **Enter** para voltar ao menu de configuração.

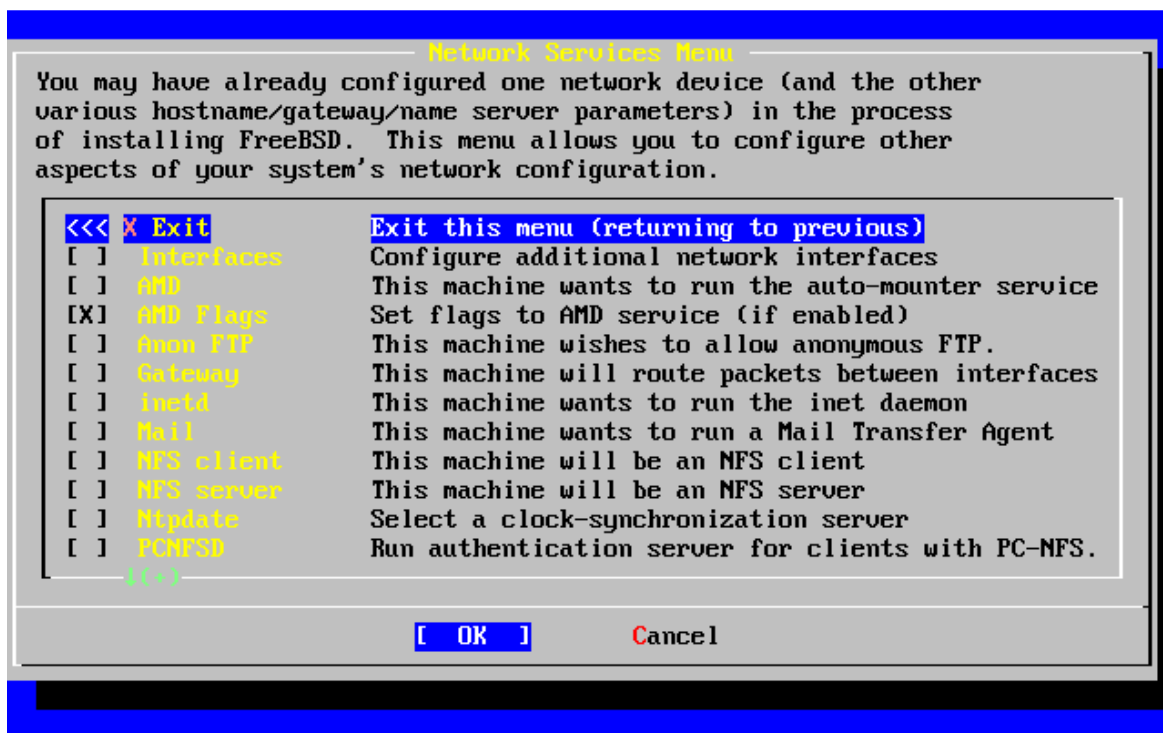
2.9.11. Configurando serviços adicionais de rede

Configurar serviços de rede pode ser uma tarefa árdua à novos usuários, se eles carecem de conhecimento nessa área. Rede, incluindo a Internet, é crítico à todo sistema operacional moderno, incluindo FreeBSD; resultado isso é que é bastante útil entender as capacidades de rede do FreeBSD de forma extensa. Fazer esta configuração durante a instalação garantirá que os usuários obtenham alguma compreensão dos vários serviços de rede que lhes estão disponíveis.

Serviços de rede são programas que aceitam entrada de dados de qualquer parte na rede. Todo esforço é feito para garantir que estes programas não causarão situações "danosas" ao sistema. Infelizmente, programadores não são perfeitos e em dado momento pode acontecer de existirem falhas nos serviços de rede que sejam vulneráveis e que possibilitem que usuários mal intencionados façam coisas ruins. É importante que você habilite apenas os serviços de rede que você sabe que precisa. Se houver dúvida, é melhor não habilitar o serviço até descobrir que precisa dele. É sempre possível habilitar o serviço mais tarde, re-executando o **sysinstall** ou utilizando as funcionalidades disponíveis no arquivo `/etc/rc.conf`.

Ao selecionar a opção "Networking" um menu similar ao seguinte, será apresentado:

Figura 2-51. Configuração de alto-nível da rede



A primeira opção, Interfaces, foi previamente coberta durante a seção de configuração dos dispositivos de rede; embora esta opção possa seguramente ser ignorada.

Selecionar a opção *AMD* inicia o suporte ao utilitário de montagem automática do BSD. Normalmente esta opção é usada em conjunto com o protocolo NFS (veja abaixo) para automatizar a montagem remota de sistema de arquivos. Nenhuma configuração especial é necessária.

O próximo da linha são as opções do AMD (*AMD flags*) Ao selecionar esta opção, um menu irá surgir pedindo para você preencher informações específicas sobre as opções AMD desejadas. O menu já contém algumas opções padrão:

```
-a /.amd_mnt -l syslog /host /etc/amd.map /net /etc/amd.map
```

A opção *-a* define a localização padrão de montagem, que é definido aqui como sendo */.amd_mnt*. A opção *-l* especifica o arquivo de log padrão; contudo, quando se define *syslogd*, toda atividade de log será atribuída à esta aplicação. O diretório */host* é utilizado para montar um sistema de arquivos exportado à partir de uma estação remota, enquanto o diretório */net* é usado para montar um sistema de arquivos exportado com base em um endereço IP. O arquivo */etc/amd.map* define as opções padrão para as exportações AMD.

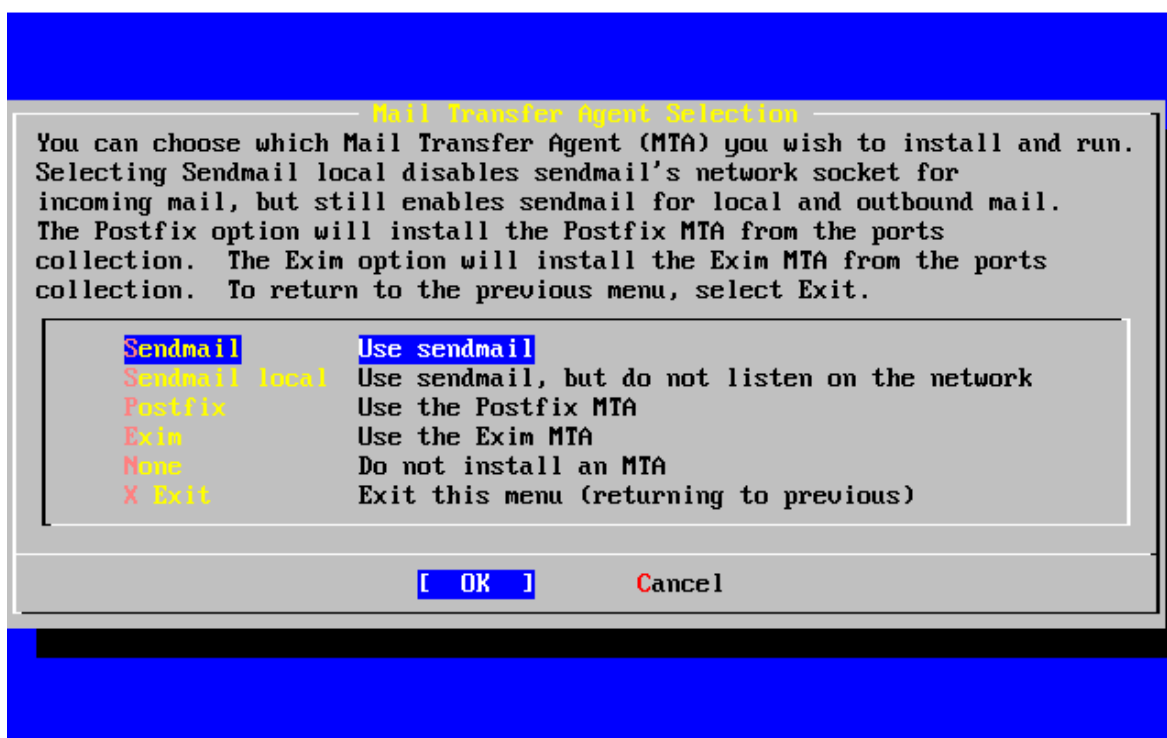
O FTP anônimo permite conexões anônimas à um servidor FTP. Fique atento aos riscos de segurança envolvidos com esta opção. Outro menu será apresentado para explicar os riscos de segurança e configurações em questão.

O menu de configuração *Gateway* definirá a estação como um servidor de ligação, como previamente explicado. Esta opção pode ser utilizada para desabilitar a opção de *gateway* caso ela tenha sido habilitada acidentalmente durante o processo de instalação.

A opção *Inetd* pode ser usada para configurar ou completamente desabilitar o *daemon inetd(8)* foi discutido anteriormente.

A opção *Mail* é utilizada para configurar o agente de transferência de correio (MTA - *Mail Transfer Agent*) padrão do sistema. Selecionar esta opção o levará ao seguinte menu:

Figura 2-52. Selecione o MTA padrão



Aqui você pode escolher qual MTA instalar e definir como padrão. Um MTA nada mais é do que um servidor de correio que entrega mensagens à usuários no sistema local e na Internet.

Ao selecionar **Sendmail** o popular servidor **Sendmail** será instalado, que é o padrão no FreeBSD. A opção *Sendmail local* define que o **Sendmail** continuará sendo o MTA padrão, mas irá desabilitar a funcionalidade de receber mensagens vindas da Internet. As outras opções aqui, **Postfix** e **Exim** agem de forma similar ao **Sendmail**. Ambos entregam mensagens de correio eletrônico; contudo, alguns usuários os preferem como alternativas ao MTA **Sendmail**.

Após escolher um MTA ou escolher não iniciar nenhum, o menu de configuração da rede aparecerá com a próxima opção, o cliente NFS.

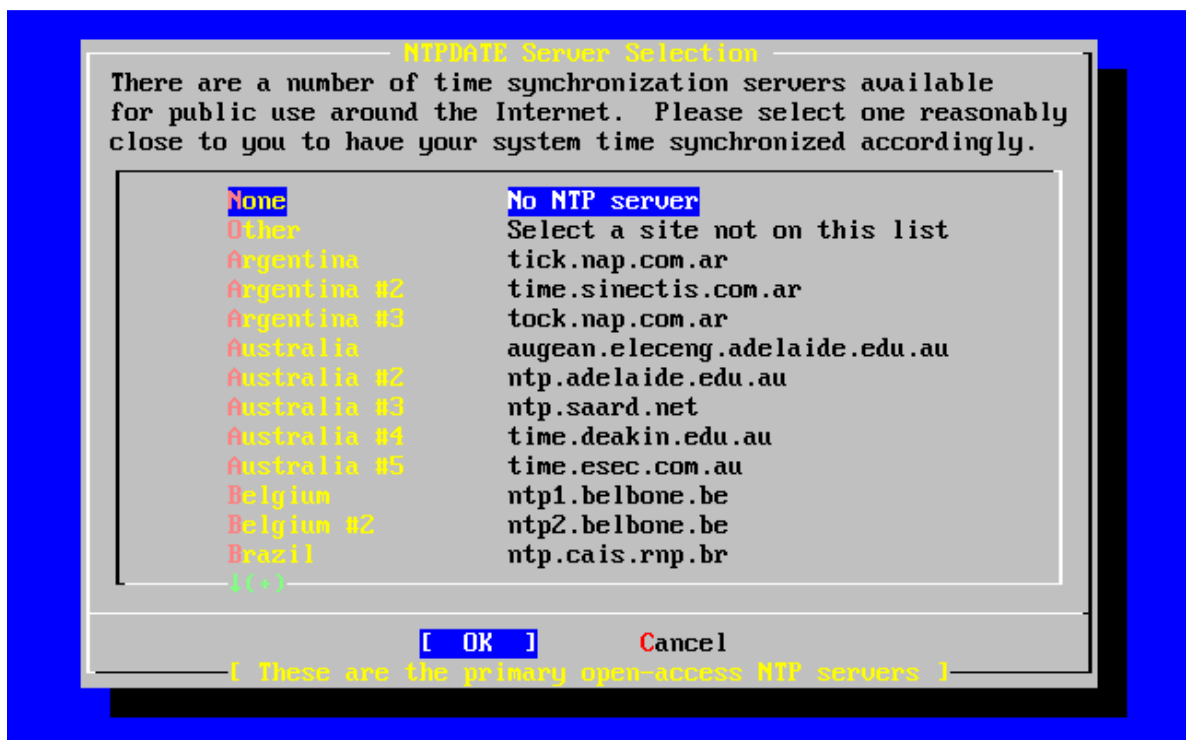
O cliente NFS configurará o sistema de forma que ele possa se comunicar com um servidor através de NFS. Um servidor NFS torna sistemas de arquivos disponíveis à

outras estações na rede através do protocolo NFS . Se a máquina que você está trabalhando for uma máquina isolada, mantenha esta opção desativada. O sistema poderá solicitar maiores configurações mais tarde; veja [Seção 19.6](#) para obter mais informações sobre configurações cliente e servidor.

Abaixo desta opção está a opção de servidor NFS que permite que o sistema seja ajustado como tal. As informações necessárias para iniciar as chamadas remotas de procedimentos, RPC, são adicionadas com esta opção. O RPC é utilizado para coordenar as conexões entre estações e programas.

Na próxima linha está a opção *Ntpdate* que trata da sincronização de hora. Ao selecioná-la, um menu como o seguinte será apresentado:

Figura 2-53. Configuração Ntpdate

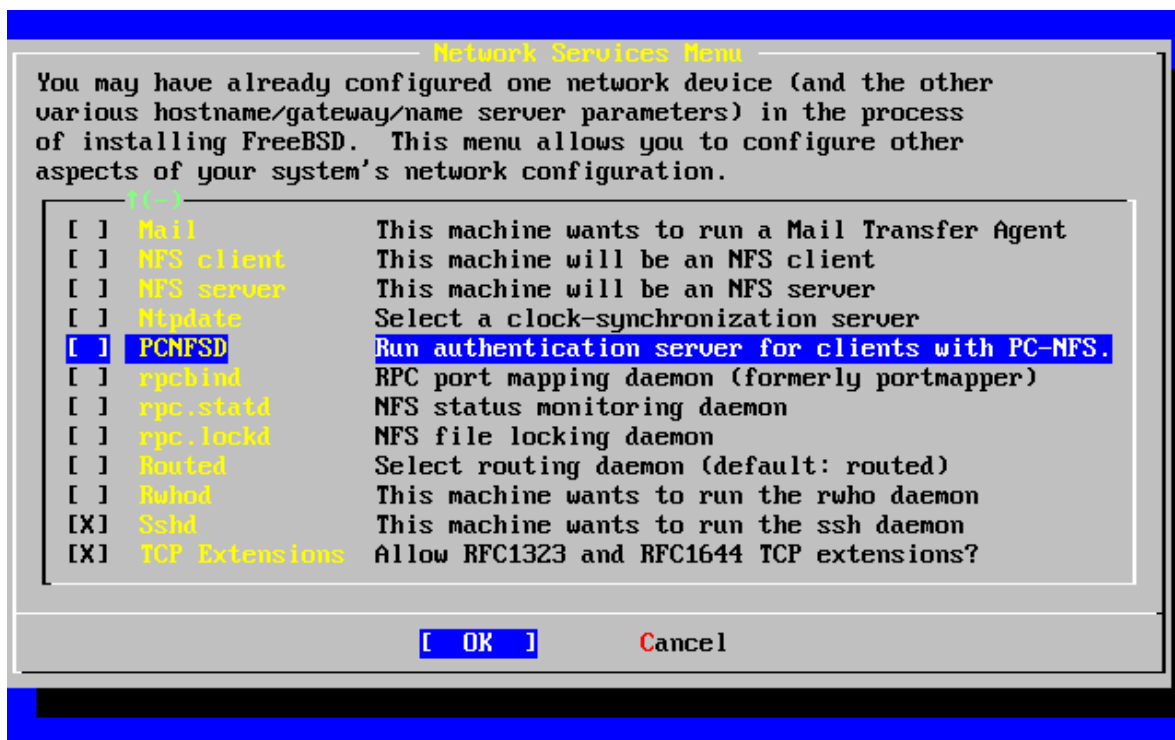


Neste menu, selecione o servidor que for mais próximo de sua localização. Um servidor mais próximo tornará a sincronia de horários mais precisa, pois um servidor distante pode provocar latência na conexão.

A opção seguinte é PCNFSD. Selecioná-la instalará o pacote `net/pcnfsd` à partir da coleção de *ports*. É uma aplicação útil que oferece serviços de autenticação NFS em sistemas que são incapazes de oferecer tais serviços por si só, como o sistema operacional DOS da Microsoft.

Agora será necessário pular um pouco a tela para visualizar as outras opções:

Figura 2-54. Configuração da rede em baixo nível



Os utilitários [rpcbind\(8\)](#), [rpc.statd\(8\)](#), e [rpc.lockd\(8\)](#) são todos utilizados para chamadas remotas de procedimento (RPC). O utilitário `rpcbind.8` gerencia a comunicação entre servidores e clientes NFS, e é necessário para que os servidores NFS operem de forma correta. O *daemon* `rpc.statd` interage com o `rpc.statd` em outra estação, para prover monitoração do estado da conexão. O estado relatado normalmente é mantido no arquivo `/var/db/statd.status`. A opção final aqui listada é `rpc.lockd` que, ao ser selecionada, provê serviço de restrição de acesso aos arquivos. Normalmente ele é utilizado em conjunto com o `rpc.statd` para monitorar quais estações estão requisitando acesso simultaneo aos arquivos e com que frequência os requisita. Enquanto estas últimas duas opções são sensacionais para depuração, não são necessárias para o correto funcionamento de clientes e servidores NFS .

Um pouco mais abaixo na lista de opções, está o item *Routed*, que na verdade não passa de um *daemon* de roteamento. O [routed\(8\)](#) gerencia tabela de roteamentos na rede, descobre roteadores *multicast* e distribui uma cópia de sua tabela de roteamento à qualquer estação fisicamente conectada à rede, que tenha função de *gateway* (veja as páginas de manual [icmp\(4\)](#) e [udp\(4\)](#)). Ao ser selecionada, um menu será apresentado solicitando a localização padrão para o utilitário. A localização padrão já está definida, e pode ser selecionada com a tecla **Enter**. Em seguida você verá um outro menu, desta feita solicitando opções extras à serem passadas ao `routed`. A opção padrão é `-q` e já deve estar disponível na tela.

Na linha seguinte está a opção *Rwhod* que inicia a aplicação [rwhod\(8\)](#) durante a inicialização do sistema. O utilitário `rwhod` envia mensagens do sistema em *broadcast* para toda a rede, periodicamente, ou adquire tais informações quando operando em modo "consumidor". Mais informações podem ser encontradas nas páginas de manual [ruptime\(1\)](#) e [rwho\(1\)](#).

A próxima opção na lista é o *daemon* [sshd\(8\)](#). Trata-se do servidor de terminal seguro do **OpenSSH** e é altamente recomendável em substituição aos tradicionais servidores `telnet` e `FTP`. O servidor `sshd` é utilizado para estabelecer uma ligação segura entre uma estação e outra, por meio de conexões criptografadas.

Finalmente, a opção de Extensões TCP. Ela habilita as extensões TCP definidas na RFC 1323 e RFC 1644. Em várias estações estas extensões podem aumentar a velocidade das conexões, mas em outras pode causar interrupção da conexão. Estas extensões não são recomendadas para servidores, mas podem trazer bons benefícios em estações cliente.

Agora que você configurou os serviços de rede, pode pular até o item mais alto e escolher a opção *Exit* e continuar para a próxima seção de configuração.

2.9.12. Configuração do servidor X

Para utilizar uma interface gráfica com o usuário, como o **KDE**, **GNOME**, ou vários outros, o servidor X necessitará ser configurado.

Nota: Para utilizar o **XFree86**TM como usuário não `root` será necessário ter o pacote [x11/wrapper](#) instalado. Ele é instalado por padrão à partir do FreeBSD 4.7. Em versões anteriores ao 4.7, ele deve ser adicionado por meio do menu de seleção de pacotes.

Para ver se sua placa de vídeo é suportada, verifique o sítio WWW do [XFree86](#).

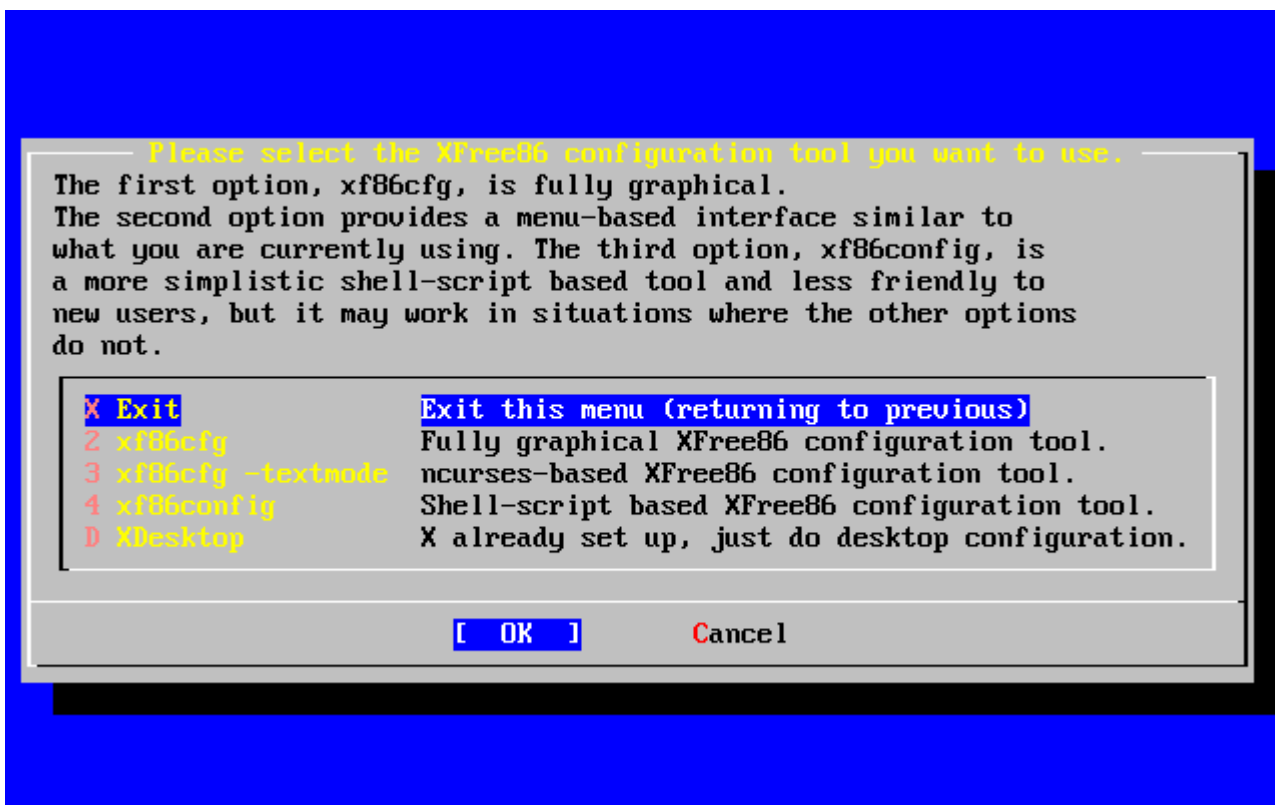
```
User Confirmation Requested
Would you like to configure your X server at this time?

[ Yes ]  No
```

Atenção É necessário conhecer as especificações do seu monitor e informações sobre sua placa de vídeo. Danos no equipamento podem eventualmente ocorrer caso as configurações estejam incorretas. Caso você não tenha estas informações, selecione [No] e faça a configuração de vídeo depois que a instalação for concluída, quando tiver estas informações em mãos. Pode utilizar o `/stand/sysinstall`, selecionando a opção `Configure` e depois disso, `XFree86`, para retornar à este processo de configuração a qualquer momento depois do sistema instalado. Configuração imprópria do servidor X neste momento pode causar instabilidade na máquina. Normalmente é aconselhável configurar o servidor X uma vez que a instalação esteja terminada.

Caso tenha as informações sobre o monitor e placa de rede, escolha [Yes] e aperte **Enter** para proceder à configuração do servidor X.

Figura 2-55. Menu de seleção do método de configuração



Existem diversas formas de configurar o servidor X. Use as teclas indicativas e selecione um dos métodos disponíveis no `sysinstall`, e aperte **Enter**. Leia todas as instruções cuidadosamente.

Os métodos `xf86cfg` e `xf86cfg -textmode` podem fazer a tela ficar escura e demorar alguns poucos segundos para iniciar. Seja paciente.

À seguir será apresentada uma ilustração da ferramenta de configuração `xf86config`. As opções de configuração que você fizer dependerão do equipamento disponível na sua máquina, de forma que suas escolhas certamente serão diferentes das apresentadas:

```

Message
You have configured and been running the mouse daemon.
Choose "/dev/sysmouse" as the mouse port and "SysMouse" or
"MouseSystems" as the mouse protocol in the X configuration utility.

[ OK ]

[ Press enter to continue ]

```

Indica que a configuração prévia do *daemon* do mouse foi detectada. Aperte **Enter** para continuar.

Ao iniciar o `xf86config`, uma breve introdução será apresentada:

```

This program will create a basic XF86Config file, based on menu
selections you

```

make.

The XF86Config file usually resides in /usr/X11R6/etc/X11 or /etc/X11. A sample XF86Config file is supplied with XFree86; it is configured for a standard VGA card and monitor with 640x480 resolution. This program will ask for a pathname when it is ready to write the file.

You can either take the sample XF86Config as a base and edit it for your configuration, or let this program produce a base XF86Config file for your configuration and fine-tune it.

Before continuing with this program, make sure you know what video card you have, and preferably also the chipset it uses and the amount of video memory on your video card. SuperProbe may be able to help with this.

Press enter to continue, or ctrl-c to abort.

Apertando **Enter, a configuração do mouse será iniciada. Garanta que as instruções sejam seguidas e use o ítem ``Mouse Systems" como protocolo de mouse e /dev/sysmouse como porta para o mesmo, se o seu dispositivo de mouse for PS/2 como no modelo ilustrado anteriormente.**

First specify a mouse protocol type. Choose one from the following list:

1. Microsoft compatible (2-button protocol)
2. Mouse Systems (3-button protocol) & FreeBSD moused protocol
3. Bus Mouse
4. PS/2 Mouse
5. Logitech Mouse (serial, old type, Logitech protocol)
6. Logitech MouseMan (Microsoft compatible)
7. MM Series
8. MM HitTablet
9. Microsoft IntelliMouse

If you have a two-button mouse, it is most likely of type 1, and if you have a three-button mouse, it can probably support both protocol 1 and 2. There are two main varieties of the latter type: mice with a switch to select the protocol, and mice that default to 1 and require a button to be held at boot-time to select protocol 2. Some mice can be convinced to do 2 by sending a special sequence to the serial port (see the ClearDTR/ClearRTS options).

Enter a protocol number: 2

You have selected a Mouse Systems protocol mouse. If your mouse is normally

in Microsoft-compatible mode, enabling the ClearDTR and ClearRTS options may cause it to switch to Mouse Systems mode when the server starts.

Please answer the following question with either 'y' or 'n'.
Do you want to enable ClearDTR and ClearRTS? n

You have selected a three-button mouse protocol. It is recommended that you do not enable Emulate3Buttons, unless the third button doesn't work.

Please answer the following question with either 'y' or 'n'.
Do you want to enable Emulate3Buttons? y

Now give the full device name that the mouse is connected to, for example /dev/tty00. Just pressing enter will use the default, /dev/mouse. On FreeBSD, the default is /dev/sysmouse.

Mouse device: /dev/sysmouse

O teclado deve ser configurado no próximo ítem. Um modelo genérico de 1001 teclas é apresentado como modelo ilustrativo. Qualquer nome pode ser usado como variante, ou simplesmente aperte **Enter** para aceitar o valor padrão.

Please select one of the following keyboard types that is the better description of your keyboard. If nothing really matches, choose 1 (Generic 101-key PC)

- 1 Generic 101-key PC
- 2 Generic 102-key (Intl) PC
- 3 Generic 104-key PC
- 4 Generic 105-key (Intl) PC
- 5 Dell 101-key PC
- 6 Everex STEPnote
- 7 Keytronic FlexPro
- 8 Microsoft Natural
- 9 Northgate OmniKey 101
- 10 Winbook Model XP5
- 11 Japanese 106-key
- 12 PC-98xx Series
- 13 Brazilian ABNT2
- 14 HP Internet
- 15 Logitech iTouch
- 16 Logitech Cordless Desktop Pro
- 17 Logitech Internet Keyboard
- 18 Logitech Internet Navigator Keyboard
- 19 Compaq Internet
- 20 Microsoft Natural Pro
- 21 Genius Comfy KB-16M
- 22 IBM Rapid Access
- 23 IBM Rapid Access II
- 24 Chicony Internet Keyboard
- 25 Dell Internet Keyboard

Enter a number to choose the keyboard.

1

Please select the layout corresponding to your keyboard

- 1 U.S. English
- 2 U.S. English w/ ISO9995-3
- 3 U.S. English w/ deadkeys
- 4 Albanian
- 5 Arabic
- 6 Armenian
- 7 Azerbaidjani
- 8 Belarusian
- 9 Belgian
- 10 Bengali
- 11 Brazilian
- 12 Bulgarian
- 13 Burmese
- 14 Canadian
- 15 Croatian
- 16 Czech
- 17 Czech (qwerty)
- 18 Danish

Enter a number to choose the country.
Press enter for the next page

1

Please enter a variant name for 'us' layout. Or just press enter
for default variant

us

Please answer the following question with either 'y' or 'n'.
Do you want to select additional XKB options (group switcher,
group indicator, etc.)? n

Em seguida, vamos para a configuração do monitor. Jamais exceda as taxa de frequência de seu equipamento. Danos poderão ocorrer. Caso tenha alguma dúvida, realize esta configuração depois que tiver estas informações em mãos.

Now we want to set the specifications of the monitor. The two critical parameters are the vertical refresh rate, which is the rate at which the whole screen is refreshed, and most importantly the horizontal sync rate, which is the rate at which scanlines are displayed.

The valid range for horizontal sync and vertical sync should be documented in the manual of your monitor. If in doubt, check the monitor database `/usr/X11R6/lib/X11/doc/Monitors` to see if your monitor is there.

Press enter to continue, or ctrl-c to abort.

You must indicate the horizontal sync range of your monitor. You can either select one of the predefined ranges below that correspond to industry-standard monitor types, or give a specific range.

It is VERY IMPORTANT that you do not specify a monitor type with a horizontal sync range that is beyond the capabilities of your monitor. If in doubt, choose a conservative setting.

```
hsync in kHz; monitor type with characteristic modes
1 31.5; Standard VGA, 640x480 @ 60 Hz
2 31.5 - 35.1; Super VGA, 800x600 @ 56 Hz
3 31.5, 35.5; 8514 Compatible, 1024x768 @ 87 Hz interlaced (no
800x600)
4 31.5, 35.15, 35.5; Super VGA, 1024x768 @ 87 Hz interlaced, 800x600
@ 56 Hz
5 31.5 - 37.9; Extended Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz
6 31.5 - 48.5; Non-Interlaced SVGA, 1024x768 @ 60 Hz, 800x600 @ 72 Hz
7 31.5 - 57.0; High Frequency SVGA, 1024x768 @ 70 Hz
8 31.5 - 64.3; Monitor that can do 1280x1024 @ 60 Hz
9 31.5 - 79.0; Monitor that can do 1280x1024 @ 74 Hz
10 31.5 - 82.0; Monitor that can do 1280x1024 @ 76 Hz
11 Enter your own horizontal sync range
```

Enter your choice (1-11): 6

You must indicate the vertical sync range of your monitor. You can either select one of the predefined ranges below that correspond to industry-standard monitor types, or give a specific range. For interlaced modes, the number that counts is the high one (e.g. 87 Hz rather than 43 Hz).

```
1 50-70
2 50-90
3 50-100
4 40-150
5 Enter your own vertical sync range
```

Enter your choice: 2

You must now enter a few identification/description strings, namely an identifier, a vendor name, and a model name. Just pressing enter will fill in default names.

The strings are free-form, spaces are allowed.
Enter an identifier for your monitor definition: Hitachi

A seleção de uma placa de rede à partir de uma lista de equipamentos disponíveis, é a próxima opção. Se você passar da sua placa na lista, continue e aperte **Enter** e a lista começará desde o princípio novamente. Apenas um trecho da lista será apresentada:

Now we must configure video card specific settings. At this point you can choose to make a selection out of a database of video card definitions.

Because there can be variation in Ramdacs and clock generators even between cards of the same model, it is not sensible to blindly copy the settings (e.g. a Device section). For this reason, after you make a selection, you will still be asked about the components of the card, with the settings from the chosen database entry presented as a strong hint.

The database entries include information about the chipset, what driver to run, the Ramdac and ClockChip, and comments that will be included in the Device section. However, a lot of definitions only hint about what driver to run (based on the chipset the card uses) and are untested.

If you can't find your card in the database, there's nothing to worry about. You should only choose a database entry that is exactly the same model as your card; choosing one that looks similar is just a bad idea (e.g. a GemStone Snail 64 may be as different from a GemStone Snail 64+ in terms of hardware as can be).

Do you want to look at the card database? y

288	Matrox Millennium G200 8MB	mgag200
289	Matrox Millennium G200 SD 16MB	mgag200
290	Matrox Millennium G200 SD 4MB	mgag200
291	Matrox Millennium G200 SD 8MB	mgag200
292	Matrox Millennium G400	mgag400
293	Matrox Millennium II 16MB	mga2164w
294	Matrox Millennium II 4MB	mga2164w
295	Matrox Millennium II 8MB	mga2164w
296	Matrox Mystique	mgal064sg
297	Matrox Mystique G200 16MB	mgag200
298	Matrox Mystique G200 4MB	mgag200
299	Matrox Mystique G200 8MB	mgag200
300	Matrox Productiva G100 4MB	mgag100
301	Matrox Productiva G100 8MB	mgag100
302	MediaGX	mediagx
303	MediaVision Proaxcel 128	ET6000
304	Mirage Z-128	ET6000
305	Miro CRYSTAL VRX	Verite 1000

Enter a number to choose the corresponding card definition.
Press enter for the next page, q to continue configuration.

288

Your selected card definition:

Identifier: Matrox Millennium G200 8MB
Chipset: mgag200
Driver: mga
Do NOT probe clocks or use any Clocks line.

Press enter to continue, or ctrl-c to abort.

Now you must give information about your video card. This will be used for the "Device" section of your video card in XF86Config.

You must indicate how much video memory you have. It is probably a good idea to use the same approximate amount as that detected by the server you intend to use. If you encounter problems that are due to the used server not supporting the amount memory you have (e.g. ATI Mach64 is limited to 1024K with the SVGA server), specify the maximum amount supported by the server.

How much video memory do you have on your video card:

- 1 256K
- 2 512K
- 3 1024K
- 4 2048K
- 5 4096K
- 6 Other

Enter your choice: 6

Amount of video memory in Kbytes: 8192

You must now enter a few identification/description strings, namely an identifier, a vendor name, and a model name. Just pressing enter will fill in default names (possibly from a card definition).

Your card definition is Matrox Millennium G200 8MB.

The strings are free-form, spaces are allowed.

Enter an identifier for your video card definition:

Em seguida, os modos de vídeo são definidos para as resoluções desejadas. Normalmente as variações úteis são 640x480, 800x600 e 1024x768, mas todos dependem da placa de vídeo os suportar, bem o monitor, e principalmente, se for adequado ao conforto visual. Ao selecionar a escala de cores, escolha o modo mais alto que sua placa suportar.

For each depth, a list of modes (resolutions) is defined. The default resolution that the server will start-up with will be the first listed mode that can be supported by the monitor and card. Currently it is set to:

"640x480" "800x600" "1024x768" "1280x1024" for 8-bit
"640x480" "800x600" "1024x768" "1280x1024" for 16-bit
"640x480" "800x600" "1024x768" "1280x1024" for 24-bit

Modes that cannot be supported due to monitor or clock constraints will

be automatically skipped by the server.

- 1 Change the modes for 8-bit (256 colors)
- 2 Change the modes for 16-bit (32K/64K colors)
- 3 Change the modes for 24-bit (24-bit color)
- 4 The modes are OK, continue.

Enter your choice: 2

Select modes from the following list:

- 1 "640x400"
- 2 "640x480"
- 3 "800x600"
- 4 "1024x768"
- 5 "1280x1024"
- 6 "320x200"
- 7 "320x240"
- 8 "400x300"
- 9 "1152x864"
- a "1600x1200"
- b "1800x1400"
- c "512x384"

Please type the digits corresponding to the modes that you want to select.

For example, 432 selects "1024x768" "800x600" "640x480", with a default mode of 1024x768.

Which modes? 432

You can have a virtual screen (desktop), which is screen area that is larger than the physical screen and which is panned by moving the mouse to the edge of the screen. If you don't want virtual desktop at a certain resolution, you cannot have modes listed that are larger. Each color depth can have a differently-sized virtual screen

Please answer the following question with either 'y' or 'n'.
Do you want a virtual screen that is larger than the physical screen?
n

For each depth, a list of modes (resolutions) is defined. The default resolution that the server will start-up with will be the first listed mode that can be supported by the monitor and card.
Currently it is set to:

"640x480" "800x600" "1024x768" "1280x1024" for 8-bit
"1024x768" "800x600" "640x480" for 16-bit
"640x480" "800x600" "1024x768" "1280x1024" for 24-bit

Modes that cannot be supported due to monitor or clock constraints will be automatically skipped by the server.

- 1 Change the modes for 8-bit (256 colors)

- 2 Change the modes for 16-bit (32K/64K colors)
- 3 Change the modes for 24-bit (24-bit color)
- 4 The modes are OK, continue.

Enter your choice: 4

Please specify which color depth you want to use by default:

- 1 1 bit (monochrome)
- 2 4 bits (16 colors)
- 3 8 bits (256 colors)
- 4 16 bits (65536 colors)
- 5 24 bits (16 million colors)

Enter a number to choose the default depth.

4

Finalmente, a configuração deve ser salva. Garanta que `/etc/XF86Config` é a localização padrão na hora de salvar seu arquivo de configuração.

I am going to write the XF86Config file now. Make sure you don't accidentally overwrite a previously configured one.

Shall I write it to `/etc/X11/XF86Config`? y

Caso a configuração falhe, você pode tentar configurar a interface gráfica novamente, selecionando [Yes] quando a seguinte mensagem surgir:

```
User Confirmation Requested
The XFree86 configuration process seems to have
failed. Would you like to try again?
```

```
[ Yes ]      No
```

Caso tenha dificuldades com a configuração do **XFree86**, selecione [No] e aperte **Enter** e continue o processo de instalação. Após a instalação, utilize o `xf86cfg -textmode` ou `xf86config` para acessar a lista de utilitários para configuração em modo texto, lembrando que você deve estar logado como usuário `root`. Existe um método adicional de configuração do **XFree86** descrito no [Capítulo 5](#). Se sua escolha for não configurar o **XFree86** agora, o menu seguinte será a seleção de pacotes.

Por padrão, o servidor gráfico pode ser interrompido a qualquer momento, com a sequência de teclas **Ctrl+Alt+Backspace**. Se algo errado acontecer você pode utilizar esta opção para evitar que seu equipamento seja danificado.

As configurações padrão que permitem a alternância entre motods de vídeo pode ser utilizada, com a sequência de teclas **Ctrl+Alt++** ou **Ctrl+Alt+-**.

Após a instalação, o monitor pode ser ajustado em sua altura, largura, ou centralização, por meio do **xvidtune** depois que o **XFree86** estiver instalado.

Existem advertências que são apresentadas com configurações impróprias puderem danificar seu equipamento. Considere-as. Não faça nada se tiver dúvidas sobre a integridade de seu *hardware*. Ao invés disso, use os controles do monitor para ajustar a visualização do X Window. Podem haver diferenças de tamanho no monitor ao alternar entre a interface gráfica e a interface de modo texto, mas isso ainda é melhor do que causar danos à seu equipamento.

Leia a página de manual do [xvidtune\(1\)](#) antes de realizar qualquer ajuste.

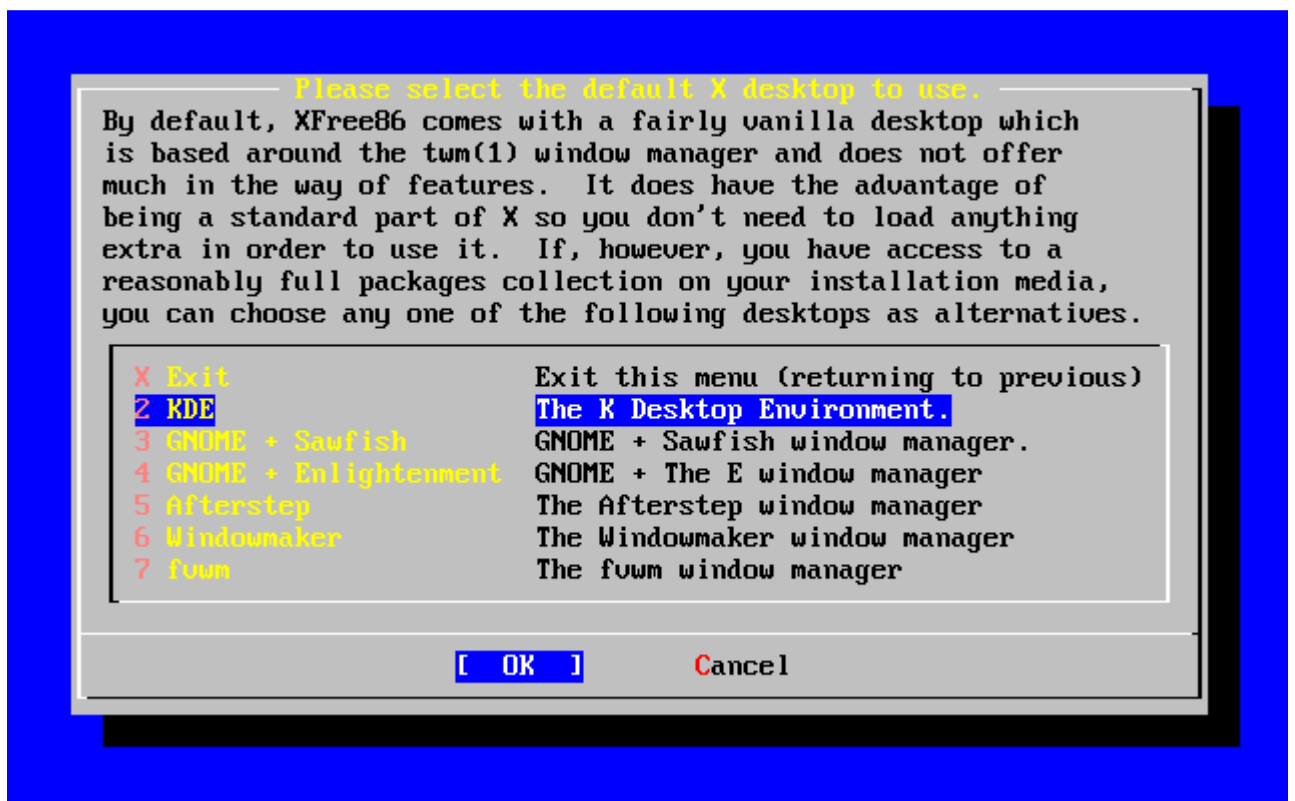
Após a configuração com sucesso do **XFree86** a próxima etapa será a escolha de um gerenciador de janelas padrão para seu *desktop*.

2.9.13. Escolha o *Desktop X* padrão

Existe uma variedade de gerenciadores de janelas disponíveis. Eles variam de ambientes extremamente básicos à ambientes completos de trabalho, com um leque enorme de aplicações. Algumas opções necessitam de espaço mínimo em disco, e pouca memória, enquanto outros com mais funcionalidades tem necessidade de mais recursos. A melhor maneira de definir qual se encaixa melhor com você é experimentar diferentes opções. Eles estão disponíveis na coleção de *ports* ou como pacotes, e podem ser adicionados após a instalação.

Você pode escolher entre um dos mais populares ambientes de trabalho e instalar/configurar como desktop padrão. Com isso você poderá inicia-lo logo após a instalação.

Figura 2-56. Escolha o *desktop* padrão



Use as teclas indicativas para selecionar o seu *desktop* preferido, e aperte **Enter**. A instalação do *desktop* escolhido ocorrerá.

2.9.14. Instalação de pacotes

Pacotes são binários pré-compilados, e são uma maneira conveniente de instalar aplicações.

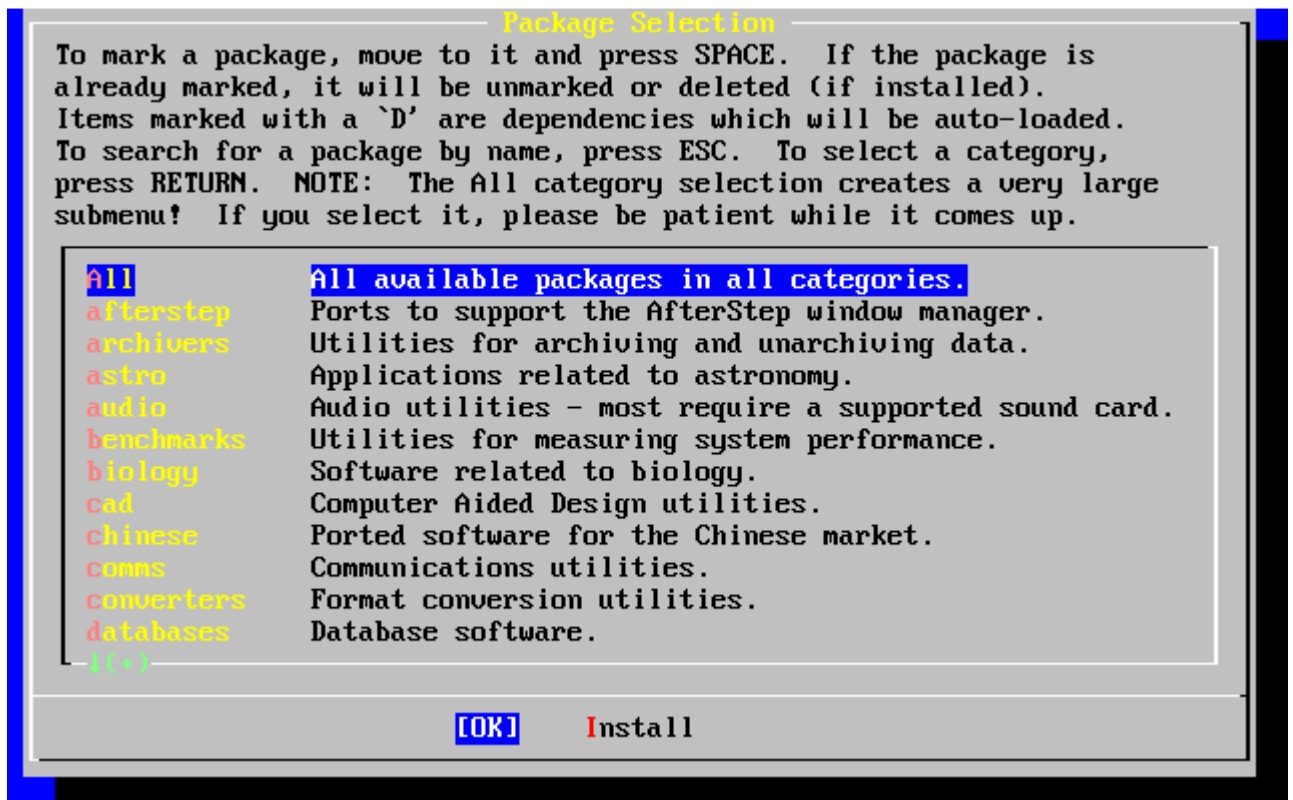
A instalação de um pacote será apresentada com o propósito de ilustração. Pacotes adicionais podem ser adicionados também neste momento, se desejado. Após a instalação, o `/stand/sysinstall` pode ser usado para a instalação de pacotes adicionais.

```
User Confirmation Requested
The FreeBSD package collection is a collection of hundreds of
ready-to-run applications, from text editors to games to WEB servers
and more. Would you like to browse the collection now?

[ Yes ] No
```

Selecionar [Yes] e apertar **Enter** o levará às telas de seleção de pacotes:

Figura 2-57. Selecione a categoria do pacote

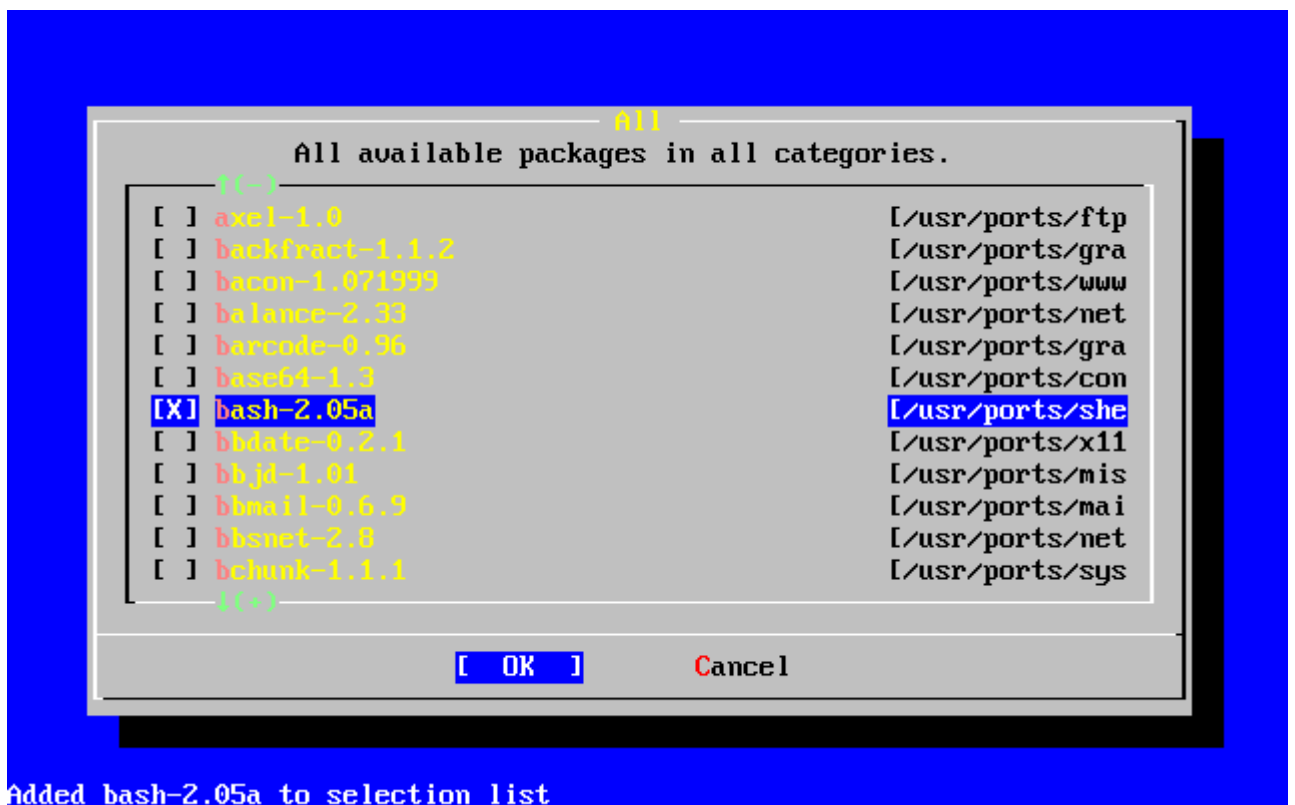


Apenas os pacotes disponíveis na mídia de instalação em uso estarão disponíveis para serem adicionados a qualquer momento.

Todos pacotes disponíveis serão apresentados sob a categoria All, que você pode selecionar agora, ou então selecione uma categoria particular. Selecione a opção desejado com as teclas indicativas e aperte **Enter**.

Um menu será apresentado, mostrando todos os pacotes disponíveis para a categoria selecionada:

Figura 2-58. Seleção de pacotes



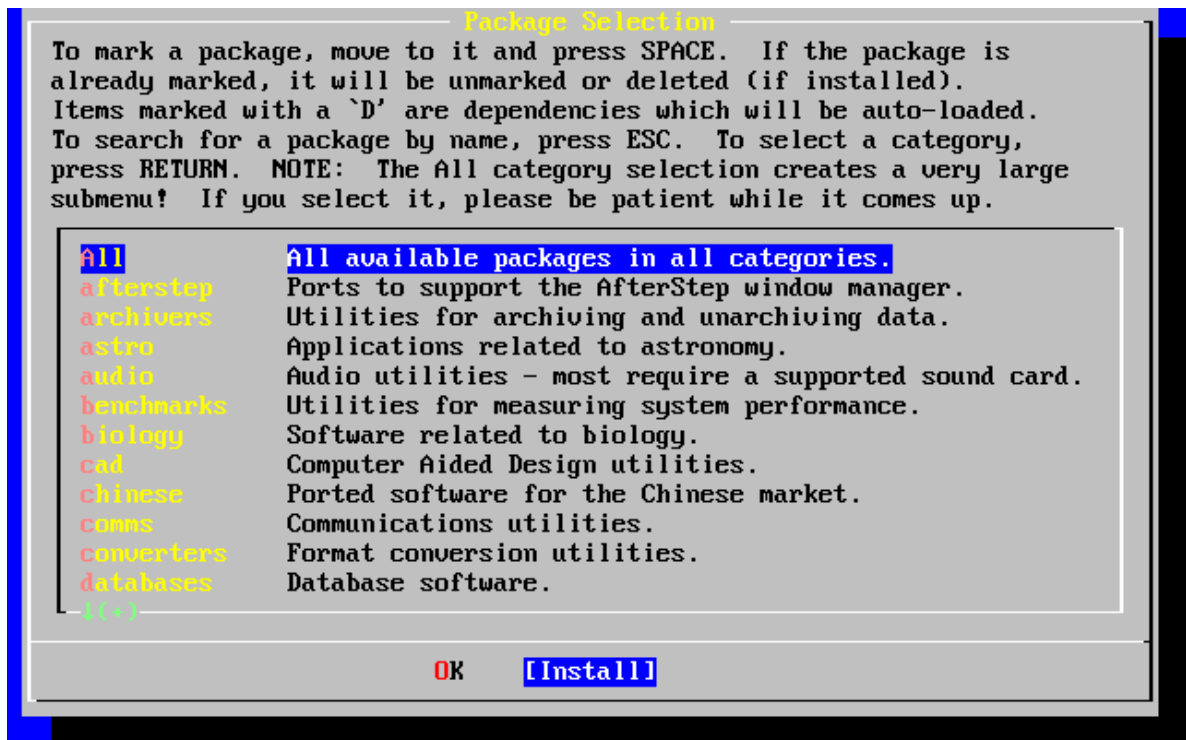
O interpretador de comandos **bash** está selecionado neste exemplo. Selecione quantos pacotes desejar, bastando para tal marcar o ítem em questão e apertar a tecla **Space**. Uma breve descrição de cada pacote aparecerá na diagonal inferior esquerda da tela.

A tecla **Tab** pode ser utilizada para alternar o cursor entre o último pacote selecionado, a opção [OK] e a opção [Cancel].

Quando terminar de marcar os pacotes para instalação, aperte **Tab** apenas uma vez para passar à opção [OK] e aperte **Enter** para voltar ao menu de seleção de pacote.

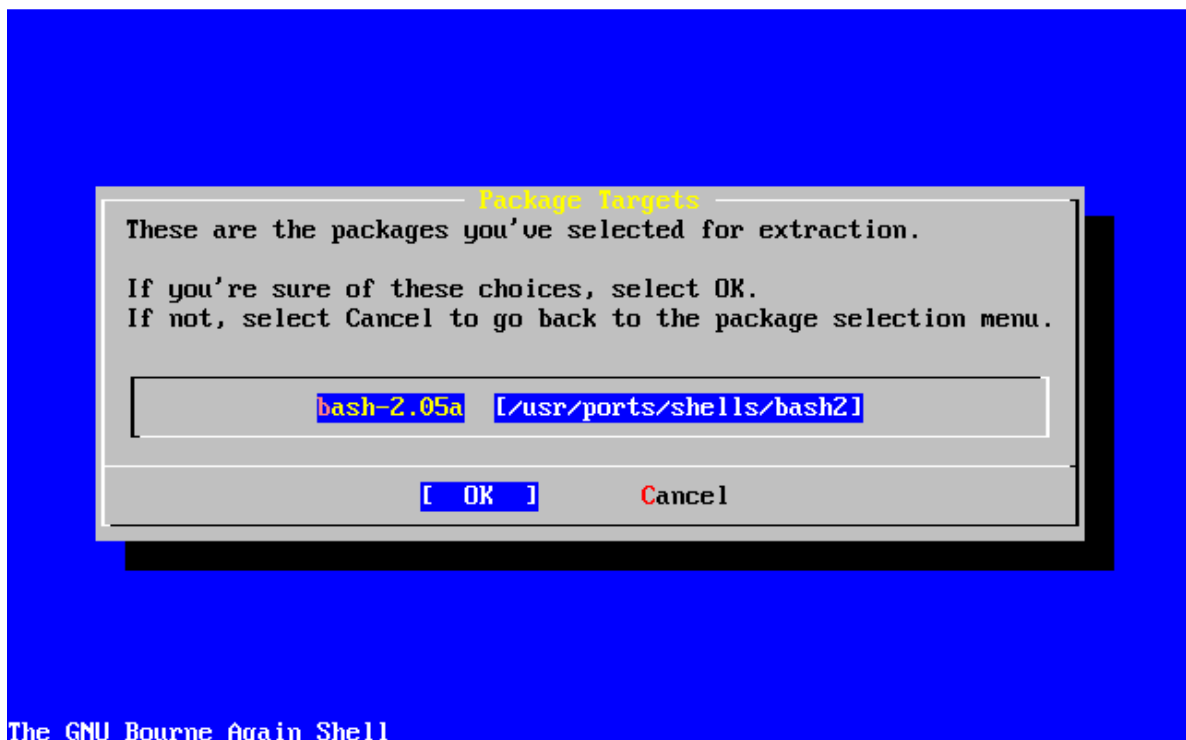
As teclas de setas esquerda e direita também serve para alternar entre [OK] e [Cancel]. Este método pode ser utilizado para selecionar [OK] e em seguida apertar **Enter** para voltar ao menu de seleção de pacotes.

Figura 2-59. Instalação de pacotes



Use a tecla **Tab** e as setas para selecionar [Install] e aperte **Enter**. Será necessário confirmar que você deseja instalar os pacotes escolhidos:

Figura 2-60. Confirmar instalação de pacotes



A instalação dos pacotes será iniciada tão logo [OK] seja selecionado e o **Enter** seja apertado. Mensagens de instalação serão apresentadas até que a mesma esteja terminada. Preste atenção se houverem mensagens de erro.

A configuração final continua assim que os pacotes forem instalados. Se você não selecionou pacote algum, e deseja voltar à configuração final, selecione Install de qualquer forma.

2.9.15. Adicionar Usuários e Grupos

Você deve adicionar ao menos um usuário durante a instalação, de forma que você possa utilizar o sistema sem se logar como superusuário, `root`. A partição raiz normalmente é pequena, e utilizar aplicações como usuário `root` pode rapidamente consumir todo seu espaço em disco. O problema maior é que como superusuário você pode danificar seu sistema sem querer, como comentado na mensagem a seguir:

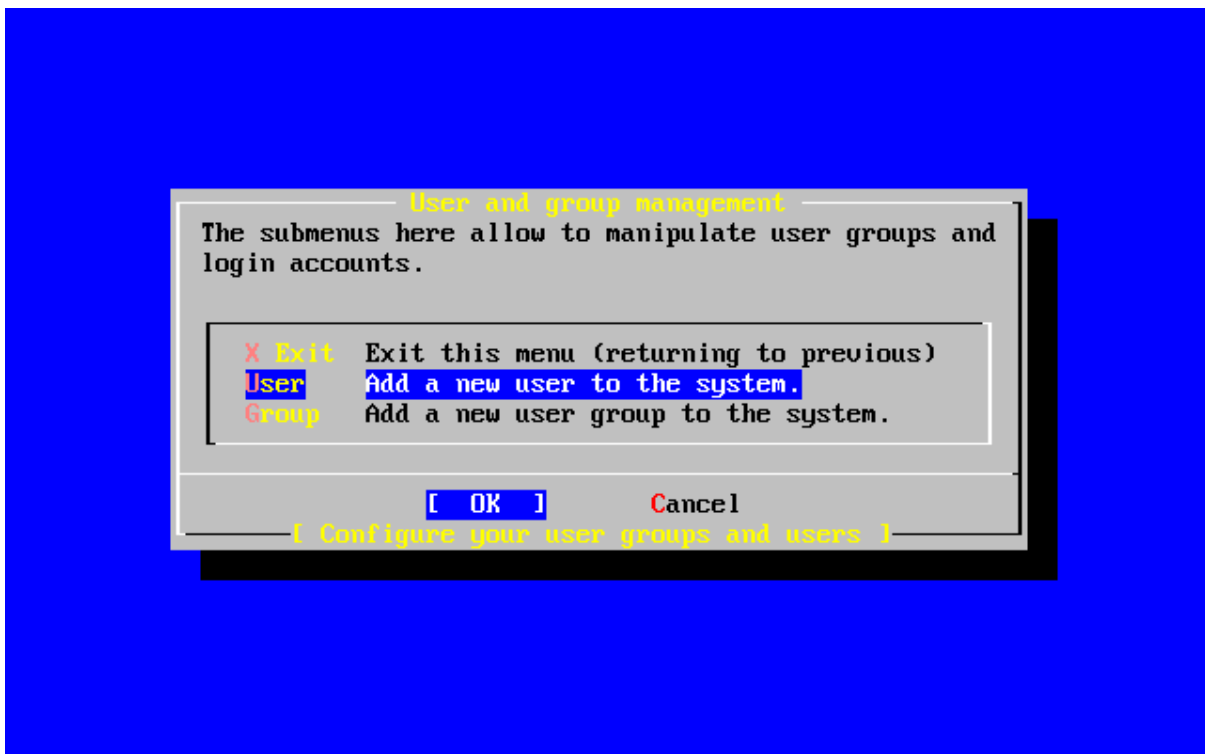
```

      User Confirmation Requested
Would you like to add any initial user accounts to the system?
Adding
at least one account for yourself at this stage is suggested since
working as the "root" user is dangerous (it is easy to do things
which
adversely affect the entire system).

      [ Yes ]   No
```

Selecione [Yes] e aperte **Enter** para continuar com a adição de usuário.

Figura 2-61. Escolha "User"



Escolha a opção User com as setas e aperte **Enter**.

Figura 2-62. Adicionando informações do usuário

The screenshot shows a terminal window with a dialog box titled "User and Group Management". Inside the dialog, there is a sub-dialog titled "Add a new user". The sub-dialog has several input fields: "Login ID:" with the value "rpratt", "UID:" with the value "1001", "Group:" which is empty, "Password:" with the value "*****", "Full name:" with the value "Randy Pratt", "Member groups:" with the value "wheel", "Home directory:" with the value "/home/rpratt", and "Login shell:" with the value "/usr/local/bin/bash". At the bottom of the sub-dialog are two buttons: "OK" and "CANCEL". Below the sub-dialog, a yellow bar contains the text "Select this if you are happy with these settings".

As descrições a seguir aparecerão na base da tela, conforme os ítems vão sendo selecionados com o **Tab**, de forma a auxiliar o usuário no momento de adicionar as informações necessárias:

Login ID

O nome de *login* para o novo usuário (imperativo).

UID

Identificação numerica para o novo usuário (mantenha o campo em branco para definição automática).

Group

O nome do grupo que o usuário fará parte (mantenha o campo em branco para uma definição automática).

Password

A senha para o novo usuário (defina este campo com cuidado!).

Full name

Nome completo do novo usuário (comentário).

Member groups

Grupos auxiliares a que este usuário pertencerá (terá os mesmos direitos definidos pra este grupo).

Home directory

Diretório *home* (mantenha em branco para o padrão).

Login shell

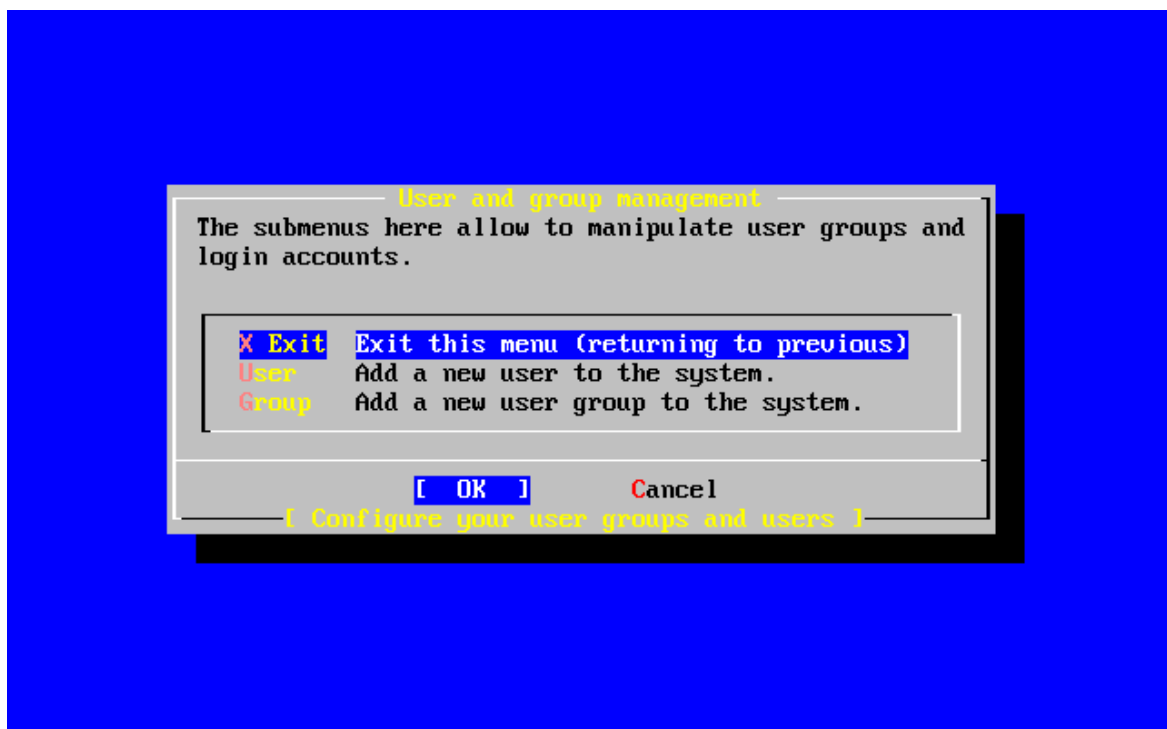
o interpretador de comandos do usuário (deixe em branco para usar o padrão, por exemplo `/bin/sh`).

O interpretador de comandos (*shell*) foi modificado de `/bin/sh` para `/usr/local/bin/bash` para usar o interpretador **bash** que foi previamente instalado como pacote. Não tente usar um interpretador de comandos que ainda não existe no sistema ou você não conseguirá se logar. O interpretador de comando mais utilizado no mundo BSD é o *C shell*, que pode ser compreendido como `/bin/tcsh`.

O usuário também foi adicionado ao grupo `wheel` de forma que ele possa se tornar superusuário e assumir os privilégios do `root`.

Quando estiver satisfeito, aperte [OK] e o menu de gerenciamento de usuário e grupo será apresentado novamente:

Figura 2-63. Sair do gerenciamento de usuário e grupo



Pode-se adicionar grupos neste momento, caso haja necessidades específicas. De outra forma, isso pode ser feito através do `/stand/sysinstall` depois que a instalação estiver completa.

Quando terminar de adicionar os usuários, selecione Exit com as setas e aperte **Enter**.

2.9.16. Defina a senha de `root`

```
Message
Now you must set the system manager's password.
This is the password you'll use to log in as "root".

[ OK ]

[ Press enter to continue ]
```

Aperte **Enter** para definir a senha de `root`.

A senha deverá ser digitada corretamente duas vezes. É desnecessário dizer, mas garanta que você não esquecerá sua senha.

```
Changing local password for root.
New password :
Retype new password :
```

A instalação continuará depois que a senha for definida corretamente.

2.9.17. Saindo da instalação

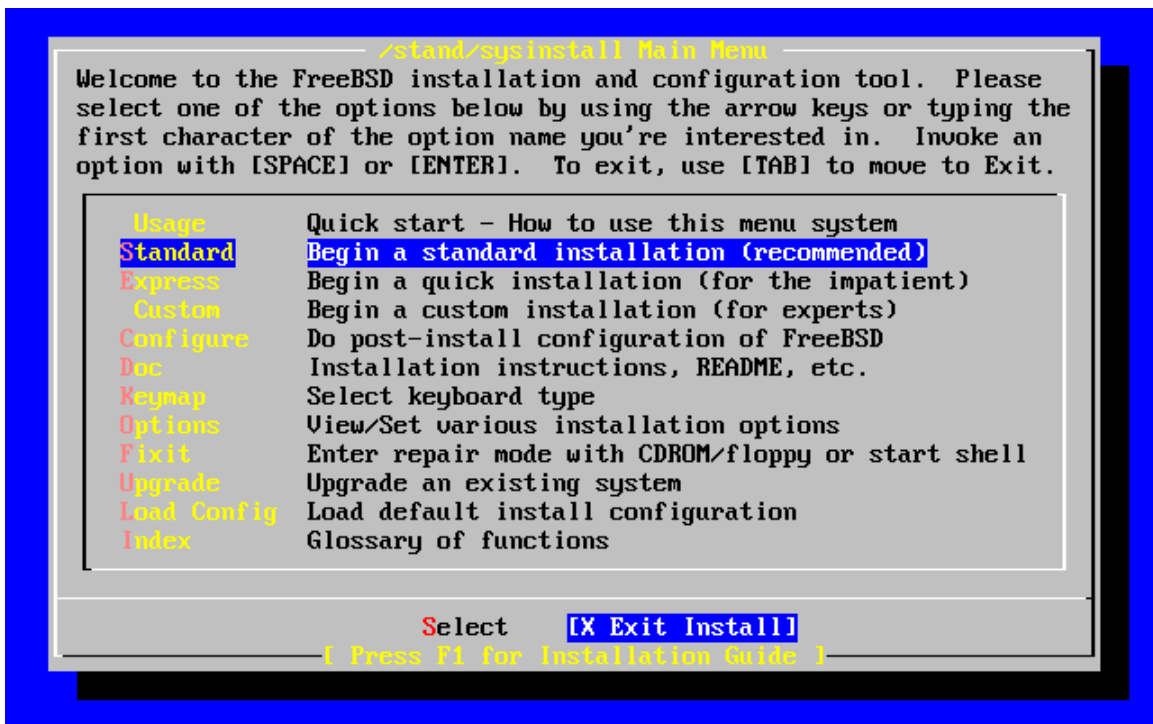
Caso queira configurar algum dispositivo de rede adicional ou qualquer outra configuração, você pode fazê-lo agora ou após a instalação, através do `/stand/sysinstall`.

```
User Confirmation Requested
Visit the general configuration menu for a chance to set any last
options?

Yes [ No ]
```

Selecione [No] com a tecla indicativa e aperte **Enter** para voltar ao menu principal de instalação.

Figura 2-64. Sair da instalação



Escolha [X Exit Install] com as teclas indicativas e aperte **Enter**. Será necessário confirmar que deseja sair da instalação:

```

      User Confirmation Requested
Are you sure you wish to exit?  The system will reboot (be sure to
remove any floppies from the drives).
```

```

      [ Yes ]   No
```

Escolha [Yes] e remova o disquete caso você tenha usado-o como mídia de inicialização. O dispositivo de CDROM permanecerá travado até que o sistema comece reinicializar. Depois o dispositivo de CDROM será destravado e o disco poderá ser retirado (seja rápido).

O sistema irá reiniciar, preste atenção a qualquer mensagem de erro que possa aparecer.

2.9.18. Inicialização do FreeBSD

2.9.18.1. Inicialização do FreeBSD em i386™

Se tudo correr bem, você verá uma série de mensagens na tela, e em dado momento chegará ao *prompt* de *login*. Você pode acompanhar o conteúdo das mensagens apertando **Scroll-Lock** e usando as teclas **PgUp** e **PgDn**. Apertando **Scroll-Lock** a tela será destravada e retornará à *prompt*.

A mensagem completa não será apresentada (por limitação de *buffer*) mas pode ser acompanhada por linha de comando depois que você se logar, digitando `dmesgno` terminal.

Logue-se com o usuário e senha definidos durante a instalação do sistema (rpratt, neste exemplo). Evite logar-se como root se não for necessário.

Mensagens típicas de inicialização (informação sobre a versão do sistema que está sendo inicializada foi omitida):

```
Copyright (c) 1992-2002 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993,
1994
    The Regents of the University of California.  All rights reserved.

Timecounter "i8254" frequency 1193182 Hz
CPU: AMD-K6(tm) 3D processor (300.68-MHz 586-class CPU)
  Origin = "AuthenticAMD" Id = 0x580 Stepping = 0
  Features=0x8001bf<FPU,VME,DE,PSE,TSC,MSR,MCE,CX8,MMX>
  AMD Features=0x80000800<SYSCALL,3DNow!>
real memory = 268435456 (262144K bytes)
config> di sn0
config> di lnc0
config> di le0
config> di ie0
config> di fe0
config> di cs0
config> di bt0
config> di aic0
config> di aha0
config> di adv0
config> q
avail memory = 256311296 (250304K bytes)
Preloaded elf kernel "kernel" at 0xc0491000.
Preloaded userconfig_script "/boot/kernel.conf" at 0xc049109c.
md0: Malloc disk
Using $PIR table, 4 entries at 0xc00fde60
npx0: <math processor> on motherboard
npx0: INT 16 interface
pcib0: <Host to PCI bridge> on motherboard
pci0: <PCI bus> on pcib0
pcib1: <VIA 82C598MVP (Apollo MVP3) PCI-PCI (AGP) bridge> at device
1.0 on pci0
pci1: <PCI bus> on pcib1
pci1: <Matrox MGA G200 AGP graphics accelerator> at 0.0 irq 11
isab0: <VIA 82C586 PCI-ISA bridge> at device 7.0 on pci0
isa0: <ISA bus> on isab0
atapci0: <VIA 82C586 ATA33 controller> port 0xe000-0xe00f at device
7.1 on pci0
ata0: at 0x1f0 irq 14 on atapci0
ata1: at 0x170 irq 15 on atapci0
uhci0: <VIA 83C572 USB controller> port 0xe400-0xe41f irq 10 at device
7.2 on pci0
usb0: <VIA 83C572 USB controller> on uhci0
usb0: USB revision 1.0
uhub0: VIA UHCI root hub, class 9/0, rev 1.00/1.00, addr 1
uhub0: 2 ports with 2 removable, self powered
chip1: <VIA 82C586B ACPI interface> at device 7.3 on pci0
ed0: <NE2000 PCI Ethernet (RealTek 8029)> port 0xe800-0xe81f irq 9 at
device 10.0 on pci0
ed0: address 52:54:05:de:73:1b, type NE2000 (16 bit)
isa0: too many dependant configs (8)
isa0: unexpected small tag 14
```

```
fdc0: <NEC 72065B or clone> at port 0x3f0-0x3f5,0x3f7 irq 6 drq 2 on isa0
fdc0: FIFO enabled, 8 bytes threshold
fd0: <1440-KB 3.5" drive> on fdc0 drive 0
atkbdc0: <keyboard controller (i8042)> at port 0x60-0x64 on isa0
atkbd0: <AT Keyboard> flags 0x1 irq 1 on atkbdc0
kbd0 at atkbd0
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: model Generic PS/2 mouse, device ID 0
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff on isa0
sc0: <System console> at flags 0x1 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
sio0 at port 0x3f8-0x3ff irq 4 flags 0x10 on isa0
sio0: type 16550A
sio1 at port 0x2f8-0x2ff irq 3 on isa0
sio1: type 16550A
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
ppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/15 bytes threshold
ppbus0: IEEE1284 device found /NIBBLE
Probing for PnP devices on ppbus0:
plip0: <PLIP network interface> on ppbus0
lpt0: <Printer> on ppbus0
lpt0: Interrupt-driven port
ppi0: <Parallel I/O> on ppbus0
ad0: 8063MB <IBM-DHEA-38451> [16383/16/63] at ata0-master using UDMA33
ad2: 8063MB <IBM-DHEA-38451> [16383/16/63] at ata1-master using UDMA33
acd0: CDROM <DELTA OTC-H101/ST3 F/W by OIPD> at ata0-slave using PIO4
Mounting root from ufs:/dev/ad0sla
swapon: adding /dev/ad0slb as swap device
Automatic boot in progress...
/dev/ad0sla: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0sla: clean, 48752 free (552 frags, 6025 blocks, 0.9% fragmentation)
/dev/ad0slf: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0slf: clean, 128997 free (21 frags, 16122 blocks, 0.0% fragmentation)
/dev/ad0slg: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0slg: clean, 3036299 free (43175 frags, 374073 blocks, 1.3% fragmentation)
/dev/ad0sle: filesystem CLEAN; SKIPPING CHECKS
/dev/ad0sle: clean, 128193 free (17 frags, 16022 blocks, 0.0% fragmentation)
Doing initial network setup: hostname.
ed0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet 192.168.0.1 netmask 0xfffff00 broadcast 192.168.0.255
    inet6 fe80::5054::5ff::fede:731b%ed0 prefixlen 64 tentative
scopeid 0x1
    ether 52:54:05:de:73:1b
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x8
    inet6 ::1 prefixlen 128
    inet 127.0.0.1 netmask 0xff000000
Additional routing options: IP gateway=YES TCP keepalive=YES
routing daemons:.
additional daemons: syslogd.
Doing additional network setup:.
Starting final network daemons: creating ssh RSA host key
Generating public/private rsal key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
```

```
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
cd:76:89:16:69:0e:d0:6e:f8:66:d0:07:26:3c:7e:2d root@k6-2.example.com
creating ssh DSA host key
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
f9:a1:a9:47:c4:ad:f9:8d:52:b8:b8:ff:8c:ad:2d:e6 root@k6-2.example.com.
setting ELF ldconfig path: /usr/lib /usr/lib/compat /usr/X11R6/lib
/usr/local/lib
a.out ldconfig path: /usr/lib/aout /usr/lib/compat/aout
/usr/X11R6/lib/aout
starting standard daemons: inetd cron sshd usbd sendmail.
Initial rc.i386 initialization:.
rc.i386 configuring syscons: blank_time screensaver moused.
Additional ABI support: linux.
Local package initialization:.
Additional TCP options:.
```

```
FreeBSD/i386 (k6-2.example.com) (ttyv0)
```

```
login: rpratt
Password:
```

A geração das chaves RSA e DSA podem levar algum tempo em estações lentas. Isso acontece apenas na primeira inicialização de um sistema recém instalado. Inicializações subsequentes serão mais rápidas.

Se o servidor X foi configurado e um ambiente de trabalho padrão foi definido, poderá ser iniciado digitando `startx` na linha de comando.

2.9.18.2. Inicialização do FreeBSD em Alpha

Uma vez terminados os procedimentos de instalação, será possível iniciar o FreeBSD ao digitar algo, no terminal SRM, algo como:

```
>>>BOOT DKC0
```

Isso instrui o sistema interno a iniciar um disco específico Para garantir que o FreeBSD seja inicializado automaticamente no futuro, utilize estes comandos:

```
>>> SET BOOT_OSFLAGS A
>>> SET BOOT_FILE ''
>>> SET BOOTDEF_DEV DKC0
>>> SET AUTO_ACTION BOOT
```

As mensagens de inicialização serão similares (mas não idênticas) à aquelas produzidas pelo processo de *boot* no FreeBSD em i386™.

2.9.19. Desligando o FreeBSD

É importante desligar o sistema operacional de forma apropriada. Não desligue a energia simplesmente. Primeiro, se torne superusuário digitando `su` na linha de comando e entrando com a senha de `root`. Isso só funcionará se o usuário for membro

do grupo `wheel`. Do contrário, logue-se como `root` e use o comando `shutdown -h now`.

```
The operating system has halted.  
Please press any key to reboot.
```

É seguro desligar o computador depois que o comando de encerramento for dado, e a mensagem ``Please press any key to reboot" aparecer. Se alguma tecla for pressionada ao invés do botão de energia do computador, o sistema reiniciará.

Ainda seria possível utilizar a combinação de teclas **Ctrl+Alt+Del** para reiniciar o sistema; contudo isso não é recomendado durante a operação normal do sistema.

2.10. Hardware Suportado

O FreeBSD atualmente roda em uma variedade de PCs com processadores ``x86" Intel, AMD, Cyrix ou NexGeon com uma grande variedade de barramentos ISA, VLB, EISA e PCI, bem como em uma série de máquinas baseadas em processadores Compaq alpha. Ainda dispõe de suporte à diversas controladoras SCSI, IDE ou ESDI, placas PCMCIA, dispositivos USB e placas seriais. O FreeBSD suporta também o barramento de microcanal da IBM (MCA).

Uma lista de equipamentos suportados acompanha o FreeBSD em cada versão, no documento ``Notas de Hardware" do FreeBSD. Este documento normalmente pode ser encontrado no arquivo chamado `HARDWARE.TXT`, na raiz do CDROM do FreeBSD ou algum sítio FTP, ou ainda no menu de documentação do **sysinstall**. O documento lista, para cada arquitetura, os equipamentos que são reconhecidamente suportados em cada versão do FreeBSD. Copias da lista de equipamentos suportados para cada versão e arquitetura do sistema podem ainda ser encontradas na página de

2.11. Resolução de problemas

A seção à seguir cobre a resolução básica de de problemas durante a instalação do FreeBSD, como os problemas que normalmente os usuários reportam. Existem ainda algumas questões e respostas para as pessoas que desejam iniciar o FreeBSD em conjunto com MS-DOS®.

2.11.1. O que fazer se algo der errado

Devido a várias limitações da arquitetura de PC, é impossível que o reconhecimento de dispositivos seja 100% confiável, contudo, existem algumas coisas que podem ser feitas quando ela falhar.

Verifique o documento Notas de Hardware da sua versão do FreeBSD e garanta que seu equipamento é suportado.

Se seu equipamento for oficialmente suportado e ainda assim você encontrar incoerência em seu controle ou qualquer outro problema, resete seu computador, e quando a opção de configuração visual do kernel for dada, escolha-a. Isso permitirá que você defina informações de seu equipamento para o sistema. O *kernel* no disco de inicialização é configurado de forma a oferecer o maior suporte possível a inúmeros equipamentos, sempre com pré-definição de recursos no padrão do fabricante, em termos de IRQs e endereços de E/S, bem como canais DMA. Se seu equipamento foi reconfigurado, será necessário utilizar o editor de configurações para informar ao FreeBSD como os recursos serão controlados.

É ainda possível identificar um dispositivo não presente que conflita com algum equipamento que já é reconhecido. Neste caso, a configuração dos *drivers* conflitantes devem ser desabilitadas.

Nota: Alguns problemas de instalação podem ser evitados ou aliviados atualizando o *firmware* de vários componentes de *hardware*, notavelmente na maioria dos casos, a placa mãe. O *firmware* da placa mãe também costuma ser chamado de BIOS e a maioria das placas-mãe ou fabricantes de computadores tem um sítio WWW onde as informações e arquivos para atualizações podem ser encontradas.

A maioria dos fabricantes aconselham fortemente que a atualização da BIOS da placa mãe não seja realizada a não ser que exista um bom motivo, como a atualização crítica de inúmeros itens. O processo de atualização *pode* dar errado, causando danos permanentes ao *chip* da BIOS.

Atenção Não desabilite qualquer dispositivo que venha a ser utilizado durante a instalação, como a saída padrão (tela -- `sc0`). Se a instalação falhar de forma misteriosa depois que você saiu do editor de configuração, você provavelmente removeu ou modificou algo que não deveria. Reinicie o sistema e tente novamente.

No modo de configuração, você pode:

- Listar dispositivos de controles disponíveis no *kernel*.
- Desabilitar dispositivos de controle para os equipamentos que não estão presentes em seu sistema.
- Modificar IRQs, DRQs e endereços de E/S utilizados pelo dispositivo de controle.

Depois de ajustar o *kernel* para que ele esteja de acordo com a configuração de seu equipamento, aperte `q` para iniciar o sistema com as novas configurações. Uma vez que a instalação esteja terminada, todas modificações que foram feitas se tornarão permanentes, de forma que você não tenha que refazer a configuração toda vez que for iniciar o sistema. Ainda é muito provável que você deseje construir um [kernel customizado](#).

2.11.2. Tratando partições MS-DOS® já existentes

Muitos usuários costumam querer instalar o FreeBSD em um PC já habitado por sistemas operacionais baseados em Microsoft®. Para tais situações, FreeBSD é acompanhado de um utilitário chamado **FIPS**. Este utilitário pode ser encontrado no

diretório `tools` no CD de instalação do sistema, ou baixado dos vários servidores espelhos do FreeBSDs.

O utilitário **FIPS** permite que uma partição MS-DOS já existente seja redimensionada em duas partes, preservando a partição original e permitindo que o sistema seja instalado no segundo trecho livre da partição. Primeiro, você tem que desfragmentar sua partição MS-DOS usando o **Desfragmentador de Discos** do Windows® (vá pelo *Explorer*, clique no seu disco rígido com o botão direito do mouse e escolha a opção de desfragmentar seu disco), ou use o **Norton Disk Tools**. Agora o programa **FIPS** pode ser executado. Ele perguntará as informações necessárias, basta seguir as instruções na tela. Depois você pode reiniciar o computador e instalar FreeBSD na partição livre. Veja o menu de distribuições de instalação (Distributions) para um valor estimado de quanto espaço em disco será necessário para o tipo de instalação que você deseja.

Existe ainda um produto muito bom da PowerQuest (<http://www.powerquest.com>) chamado **PartitionMagic**®. Esta aplicação é muito mais funcional que o **FIPS**, e é altamente recomendado se você planeja adicionar/remover os sistemas com certa frequência. Ele custa dinheiro, portanto se você planeja instalar FreeBSD e mantê-lo instalado, **FIPS** vai te atender muito bem.

2.11.3. Utilizando sistema de arquivos do MS-DOS e do Windows® File Systems

No momento o FreeBSD não suporta sistema de arquivos comprimidos com a aplicação **Double Space**™. Portanto o sistema de arquivos deve ser descomprimido antes que o FreeBSD possa acessá-lo. Isso pode ser feito com o **Agente de Compressão** localizado no menu Iniciar> Programas > Ferramentas do sistema.

FreeBSD suporta sistemas de arquivos MS-DOS. É necessário usar o comando [mount_msdos\(8\)](#) (que no FreeBSD 5.X se tornou o [mount_msdosfs\(8\)](#)) com os parâmetros usuais. A utilização mais comum deste comando é:

```
# mount_msdos /dev/ad0s1 /mnt
```

Neste exemplo, o sistema de arquivos MS-DOS está localizado na primeira partição do disco rígido primário. Sua situação pode ser distinta, verifique a saída do comando `dmesg`, e do `mount`. Eles devem produzir informações o bastante para te dar idéia sobre a disposição do particionamento.

Nota: Sistemas de arquivos MS-DOS estendidos normalmente são mapeados depois das partições do FreeBSD. Em outras palavras, o número da partição podem ser maiores dos que os utilizados pelo FreeBSD. Por padrão a primeira partição MS-DOS se for `/dev/ad0s1` e a partição FreeBSD for `/dev/ad0s2` a partição estendida MS-DOS será `/dev/ad0s3`. Para alguns, isto pode ser confuso no início.

Partições NTFS podem ser montados de forma similar, usando o comando [mount_ntfs\(8\)](#).

2.11.4. Perguntas e respostas para usuários de Alpha

Esta seção responde algumas perguntas normalmente realizadas sobre a instalação do FreeBSD em sistemas Alpha.

2.11.4.1. [Posso iniciar o sistema à partir do ARC ou do console da BIOS do Alpha?](#)

2.11.4.2. [Socorro, não tenho espaço disponível! Terei que apagar todos os meus dados antes?](#)

2.11.4.3. [Posso montar meu sistema de arquivos do Tru64 da Compaq ou do VMS?](#)

2.11.4.1. Posso iniciar o sistema à partir do ARC ou do console da BIOS do Alpha?

Não. FreeBSD, assim como o Tru64 da Compaq e o VMS, só podem ser iniciados à partir do console SRM.

2.11.4.2. Socorro, não tenho espaço disponível! Terei que apagar todos os meus dados antes?

Infelizmente, sim.

2.11.4.3. Posso montar meu sistema de arquivos do Tru64 da Compaq ou do VMS?

Não, não no momento.

2.12. Guia avançado de instalação

Contribuição de Valentino Vaschetto.

Essa seção descreve as formas de instalar o FreeBSD em casos excepcionais.

2.12.1. Instalando FreeBSD em um sistema sem monitor e sem teclado

Esse tipo de instalação é chamada "instalação decapitada" ("headless install", em inglês) porque a estação onde você quer instalar o FreeBSD não tem um monitor, ou nem sequer tem um dispositivo de saída VGA. Você pergunta, como isso pode ser possível? Usando um console serial. Um console serial basicamente utiliza uma outra máquina para a função de tela principal e teclado de um sistema. Para isso, basta seguir os passos para criar um disquete de instalação, explicado no [Seção 2.2.7](#).

Para modificar estes disquetes e iniciar o sistema em um console serial, siga os seguintes passos:

1. **Habilitando os disquetes de inicialização em um console serial**

Se você iniciar o sistema a partir dos disquetes que você acabou de criar, ele irá entrar no modo normal de instalação. Nós queremos que o FreeBSD seja iniciado em um console serial, para prosseguirmos com nossa instalação. Para isso, monte o disquete do `kern.flp` no seu FreeBSD usando o comando [mount\(8\)](#).

```
# mount /dev/fd0 /floppy
```

Agora que seu disquete está montado, entre no diretório em questão:

```
# cd /floppy
```

É aqui que você deve configurar o disquete à iniciar o processo de instalação por meio de um console serial. É necessário criar um arquivo chamado `boot.config` contendo a linha `/boot/loader -h`. Tudo que este comando faz é passar uma opção ao processo de inicialização através do carregador do sistema, informando que ele deve carregar os componentes em questão, em um console serial.

```
# echo "/boot/loader -h" > boot.config
```

Agora que o disquete está corretamente configurado, basta desmontar o disquete usando o comando [umount\(8\)](#)

```
# cd /  
# umount /mnt
```

Agora você pode retirar o disquete do dispositivo.

2. Conectando seu cabo de modem nulo (*Null Modem Cable*)

Agora é necessário conectar um cabo de modem nulo entre as duas estações. Conecte o cabo nas portas seriais das duas máquinas. *Um cabo serial normal não funcionará aqui*, é necessário um cabo de modem nulo, pois ele tem alguns fios cruzados internamente.

3. Iniciando a instalação

Agora é hora de prosseguir com a instalação. Coloque o disquete gerado com a imagem `kern.flp` no dispositivo de disquetes da estação decapitada e ligue a máquina.

4. Conectando-se à sua estação decapitada

Agora você deve se conectar à máquina com o comando [cu\(1\)](#):

```
# cu -l /dev/cuaa0
```

É isso! Agora você está conectado à estação decapitada através de sua sessão `cu`. Ele pedirá para que você substitua o disco atual pelo `mfsroot.flp`, e depois disso

possibilitará a seleção do tipo de terminal que deve ser utilizado. Escolhe o console colorido do FreeBSD e continue com sua instalação!

2.13. Preparando sua própria mídia de instalação

Nota: Para evitar redundâncias, a expressão "disco do FreeBSD" neste contexto, refere-se ao CDROM ou DVD do FreeBSD que você adquiriu ou criou sozinho.

Pode haver algumas situações onde será preciso criar sua própria mídia de instalação ou origem. Pode ser mídia física como fita, ou uma origem de onde o `sysinstall` possa acessar todos os arquivos, como um sítio FTP local, ou uma partição MS-DOS®.

Por exemplo:

- Você tem inúmeras máquinas conectadas à sua rede local, e apenas um disco do FreeBSD. Você pretende criar um sítio FTP local com o conteúdo do disco do FreeBSD, e depois permitir que estas máquinas usem o sítio FTP em questão ao invés de se conectarem na Internet.
- Você tem um disco do FreeBSD, e o FreeBSD não reconhece seu dispositivo de CD/DVD, mas o MS-DOS/Windows® reconhece. Você quer copiar os arquivos de instalação do FreeBSD para uma partição DOS no mesmo computador e então instalar o FreeBSD à partir destes arquivos.
- O computador onde você pretende instalar o FreeBSD não tem um dispositivo de CD/DVD nem está na rede, mas você pode se conectar por cabo paralelo ou serial através de um "Laplink", em um computador que tem esses dispositivos.
- Você quer criar uma fita que pode ser utilizada para a instalação do FreeBSD.

2.13.1. Criando um CD de instalação

Como parte integrante de cada versão, o projeto FreeBSD disponibiliza duas imagens de CD ("imagens ISO"). Estas imagens podem ser gravadas para gerar o disco de instalação do FreeBSD. Se você tem um gravador de CD e largura de banda suficiente, esta é a maneira mais fácil de instalar o FreeBSD.

1. Baixe a imagem ISO correta

As imagens ISO de cada versão podem ser baixadas à partir de `ftp://ftp.FreeBSD.org/pub/FreeBSD/ISO-IMAGES-arquitetura/versão` ou do servidor espelho mais próximo. Substitua as expressões *arquitetura* e *versão* com os valores apropriados.

Este diretório normalmente terá as seguintes imagens:

Tabela 2-5. Nomes e significados das imagens ISO do FreeBSD

Nome do arquivo	Conteúdo
<code>versão-mini.iso</code>	Tudo que você precisa para instalar o FreeBSD.
<code>versão-disc1.iso</code>	Tudo que você precisa para instalar o FreeBSD, acrescido de quantos pacotes adicionais de terceiros couberem no disco.
<code>versão-disc2.iso</code>	Um ``sistema de arquivos vivo (Live)"', que costuma ser usado em conjunto com a opção de recuperação (``Repair'') do sysinstall . Uma cópia da árvore CVS do FreeBSD, e quantos pacotes de terceiros couberem no disco.

Você *deve* baixar uma das duas, ou a imagem mini ISO ou o disco um. Não baixe ambos, afinal a imagem do disco um contém tudo que o mini ISO oferece, e um pouco mais.

Use o mini ISO se o seu acesso à Internet é barato. Ele permitirá que você instale o FreeBSD, e todos os pacotes de terceiros você pode baixar e instalar usando o sistema de *ports* e *packages* (veja [Capítulo 4](#)) conforme necessário.

Use a imagem do disco um, se você pretende obter uma seleção razoável de pacotes de terceiros no mesmo disco.

As imagens adicionais de disco são úteis, mas não essenciais, especialmente se você tiver acesso rápido à Internet.

2. Grave os CDs

Você deve gravar as imagens de CD no disco. Se você pretende fazer isso em outro sistema FreeBSD, veja as informações em [Seção 12.5](#) para obter mais informações (em particular, [Seção 12.5.3](#) e [Seção 12.5.4](#)).

Se você pretende fazê-lo em outra plataforma, então será preciso usar os utilitários existentes para controlar seu gravador de CD na plataforma em questão. As imagens disponíveis estão no formato ISO padrão, que a maioria das aplicações para gravação de CD suportam.

2.13.2. Criando um sítio FTP local com um disco do FreeBSD

Os discos do FreeBSD são dispostos da mesma forma como no sítio FTP. Isso torna a criação de um sítio FTP local muito fácil, para poder utilizá-lo em outras máquinas na sua rede ao instalar o FreeBSD.

1. No servidor FreeBSD onde o sítio FTP ficará hospedado, garanta que o CDROM do FreeBSD está no dispositivo, e montado no ponto `/cdrom`.
2. `# mount /cdrom`

3. Crie uma conta para FTP anônimo no `/etc/passwd`. Para isso, edite o `/etc/passwd` com o comando `vipw(8)` e adicione esta linha:
4. `ftp:*:99:99::0:0:FTP:/cdrom:/nonexistent`
5. Garanta que o serviço de FTP está habilitado no `/etc/inetd.conf`.

Qualquer pessoa com conectividade em rede para sua máquina poderá escolher o tipo de mídia FTP e digitar `ftp://sua máquina` depois de selecionar a opção ``Other" no menu de sítios FTP, durante a instalação.

AtençãoEssa abordagem está OK para qualquer máquina na sua rede local, e que esteja protegida por seu *firewall*. Oferecer serviços de FTP para outras estações na Internet (e não apenas na sua rede local) expõe seu computador à atenção de usuários mal intencionados, ou apenas indesejados. Recomendamos fortemente que você siga boas práticas de segurança para fazer isso.

2.13.3. Criando disquetes de instalação

Se você tem que instalar o sistema a partir de um disquete (o que nós *não* aconselhamos), ou por questões de equipamentos não suportados ou simplesmente porque você insiste em fazer as coisas da maneira difícil, primeiro você deve preparar alguns disquetes para a instalação.

No mínimo, você vai precisar de quantos disquetes forem necessários, de 1.44 MB ou 1.2 MB, para armazenar o diretório `bin` (distribuição binária). Se você estiver fazendo os disquetes à partir do DOS, então eles *DEVEM* estar formatados com o comando `FORMAT` do MS-DOS. Use o *Explorer* para formatar os discos (clique com o botão direito sobre o dispositivo `A:` e escolha ``Formatar").

Não confie em disquetes pré-formatados de fábrica. Formate-os novamente, apenas para garantir. Muitos problemas já foram relatados e descobriu-se depois que o motivo era apenas mídias formatadas de maneira imprópria, e é por isso que estamos lembrando você deste detalhe.

Se você estiver criando os disquetes em outra máquina, formatar ainda é uma boa idéia, contudo não será necessário colocar um sistema de arquivos DOS em cada disquete. Você pode usar os comandos `disklabel` e `newfs` para definir UFS como sistema de arquivos, seguindo a sequência de comandos (para um disquete de 3.5", 1.44 MB) ilustrada a seguir:

```
# fdformat -f 1440 fd0.1440
# disklabel -w -r fd0.1440 floppy3
# newfs -t 2 -u 18 -l 1 -i 65536 /dev/fd0
```

Nota: Use `fd0.1200` e `floppy5` para discos de 5.25", 1.2 MB.

Depois você pode monta-los e escrever nos mesmos, como qualquer outro sistema de arquivos.

Depois de ter formatado dos disquetes, será necessário copiar os arquivos para eles. Os arquivos de distribuição são divididos em blocos com tamanhos definidos

convenientemente para que cinco deles caibam em um disquete convencional de 1.44 MB. Utilize todos seus disquetes, empacotando quantos arquivos couberem em cada um deles, até que você tenha todas as distribuições desejáveis em disquete, sempre da mesma forma. Cada distribuição deve estar em um subdiretório do disquete, por exemplo: `a:\bin\bin.aa`, `a:\bin\bin.ab`, e assim por diante.

Uma vez que você tenha chegado à tela de seleção de mídia durante o processo de instalação, escolha `select ``Floppy``` e você será indagado sobre o restante.

2.13.4. Instalando a partir de uma partição MS-DOS®

Para preparar uma instalação à partir de uma partição MS-DOS, copie os arquivos da distribuição no diretório chamado `freebsd` no diretório raiz da partição. Por exemplo, `c:\freebsd`. A estrutura de diretório do CD ou do sítio FTP devem ser parcialmente reproduzidas neste diretório, então sugerimos fortemente a utilização do comando `xcopy` do MS-DOS, se estiver fazendo a cópia a partir do CD. Por exemplo, para preparar uma instalação mínima do FreeBSD:

```
C:\> md c:\freebsd
C:\> xcopy e:\bin c:\freebsd\bin\ /s
C:\> xcopy e:\manpages c:\freebsd\manpages\ /s
```

Assumindo que `c:` é onde existe espaço livre e `E:` é onde seu CDROM está inserido.

Se você não tem um dispositivo de CD, você pode baixar a distribuição à partir de ftp.FreeBSD.org. Cada distribuição está em seu próprio diretório; por exemplo, a distribuição `base` pode ser encontrada no diretório [5.3/base/](http://ftp.FreeBSD.org/5.3/base/).

Nota: Nas versões 4.X do FreeBSD e anteriores, a distribuição ```base``` era chamada de ```bin```. Ajuste os comandos e URLs apresentados acima de acordo com a versão que você estiver utilizando.

Desejando instalar muitas distribuições a partir de uma partição MS-DOS (e tendo espaço livre para isso), instale cada uma sob o diretório `c:\freebsd` -- A distribuição `BIN` é a única necessária para uma instalação mínima.

2.13.5. Criando uma fita de instalação

Instalar o sistema à partir de uma fita provavelmente é o método mais fácil de todos, bem perto de uma instalação FTP ou por CDROM. O programa de instalação espera que os arquivos sejam simplesmente empacotados com o [tar\(1\)](#) e colocados na fita. Depois de obter todos os arquivos da distribuição que você quer instalar, basta empacotá-los com o [tar\(1\)](#) na fita:

```
# cd /freebsd/distdir
# tar cvf /dev/rwt0 dist1 ... dist2
```

Quando você realizar a instalação, deverá garantir que existe espaço suficiente para um diretório temporário (que você poderá escolher) que acomodará *todo* o conteúdo da fita que você criou. Devido à natureza não aleatória de acesso às fitas, este método de

instalação requer um espaço considerável para armazenamento temporário. Você deve prever a necessidade de tanto espaço temporário quanto a quantidade de dados gravados na fita.

Nota: Ao iniciar a instalação, a fita deve estar no dispositivo *antes* de iniciar o processo de boot à partir do disquete. A varredura por dispositivos pode, do contrário, falhar em procura-lo.

2.13.6. Antes de instalar via rede

Existem três tipos de instalação por rede disponíveis. Porta serial (PPP ou SLIP), porta paralela (PLIP (cabo *laplink*)), ou rede *Ethernet* (qualquer placa de rede padrão, incluindo alguns dispositivos PCMCIA).

O suporte SLIP é um tanto primitivo e limitado primariamente à redes cabeadas, como um cabo serial ligado entre um *laptop* e outro computador. O *link* deve ser cabeado, pois a instalação SLIP no momento não oferece recursos para discagem; este suporte é oferecido com o utilitário PPP, que deve ser utilizado com preferência sobre SLIP sempre que possível.

Se você estiver usando um modem, então PPP certamente é sua única escolha. Garanta que você tem as informações do seu provedor de serviços em mãos, pois será necessário sabê-las bem no início do processo de instalação.

Se você usa PAP ou CHAP para se conectar ao seu ISP (em outras palavras, se você se conecta ao ISP no Windows sem usar uma rotina), então tudo que você precisa fazer é digitar o comando `dia1` no terminal **ppp**. Do contrário, será necessário saber como discar para seu ISP, usando ps ``comandos AT" específicos para seu modem, pois o discador PPP provê um emulador de terminal bastante simples. Por gentileza, refira-se ao [handbook](#) do PPP em modo usuário e às entradas da [FAQ](#) sobre PPP. Se você tiver problemas, as mensagens de *log* podem ser direcionadas à sua tela, utilizando os comandos `set log local`

Se existir uma conexão cabeada para outra máquina FreeBSD (2.0-RELEASE ou posterior) disponível, você deve considerar também a instalação por meio de um cabo ``laplink " na porta paralela. O tráfego de dados em uma porta paralela é muito maior do que normalmente é possível em uma linha serial (cerca de 50 kbytes/sec), resultando portando em uma instalação mais rápida.

Finalmente, para a instalação por rede mais rápida possível, uma placa de rede *Ethernet* é sempre uma boa escolha! FreeBSD suporta a maioria das placas *Ethernet* disponíveis para PC; uma tabela das placas suportados (e suas configurações usuais) é disponibilizada nas Notas de Hardware de cada versão do FreeBSD. Se você estiver utilizando um dos cartões PCMCIA suportados, garanta também que ela está inserida *antes* que o *laptop* seja ligado! O FreeBSD não suporta, infelizmente, inserção à quente de cartões PCMCIA durante a instalação.

Você precisará também saber o seu endereço IP na rede, e o valor da máscara de rede para sua classe de endereçamento, bem como o nome da sua estação. Se você estiver instalando por meio de uma conexão PPP e não tem um endereço IP estático, não tema,

o endereço IP é atribuído automaticamente pelo seu ISP. Seu administrador de sistemas pode informar que valores usar para uma configuração específica. Se você for fazer referência à outras estações por nome, e não por endereçamento IP, será necessário saber o endereço de um servidor de nomes e também o endereço do seu *gateway* (se você estiver utilizando PPP, é o endereço IP da outra ponta do provedor). Se sua intenção é instalar por FTP por meio de um *proxy* HTTP, será necessário saber o endereço do *proxy*. Se você não faz idéia da resposta para algumas ou a maioria destas questões, então você deveria conversar com seu administrador de sistemas ou seu ISP *antes* de tentar esse tipo de instalação.

2.13.6.1. Antes de instalar por NFS

A instalação NFS é trivial. Basta simplesmente copiar os arquivos da distribuição FreeBSD que você deseja utilizar no servidor NFS e apontar a seleção da mídia NFS para este servidor.

Se o servidor suporta exclusivamente acesso em "portas privilegiadas" (normalmente este é o padrão em estações de trabalho Sun), será necessário definir esta opção no menu "Options" antes que a instalação possa prosseguir.

Se sua placa de rede *Ethernet* é de baixa qualidade e sofre com taxas de transferência baixa, você ainda pode desejar habilitar a opção apropriada neste mesmo menu.

Para uma instalação NFS funcionar, o servidor deve suportar montagem de subdiretórios, por exemplo, se o diretório de sua distribuição FreeBSD 5.3 fica localizado em: `ziggy:/usr/archive/stuff/FreeBSD`, então o servidor `ziggy` deverá permitir montagem direta do `/usr/archive/stuff/FreeBSD`, não apenas do `/usr` ou `/usr/archive/stuff`.

No arquivo `/etc/exports` do FreeBSD, este controle é feito por meio das opções `-alldirs`. Outros servidores NFS podem manter convenções distintas. Se você estiver tendo problemas com permissões negadas ("permission denied") no servidor, é muito provável que você não tenha habilitado estas configurações de forma apropriada.

Capítulo 3. UNIX Básico

Índice

- 3.1. [Sinópsse](#)
- 3.2. [Consoles virtuais e terminais](#)
- 3.3. [Permissões](#)
- 3.4. [Estrutura de Diretórios](#)
- 3.5. [Organização de disco](#)
- 3.6. [Montando e desmontando sistemas de arquivos](#)
- 3.7. [Processos](#)
- 3.8. [Daemons, sinais e controle de processos](#)
- 3.9. [Shells](#)
- 3.10. [Editores de Texto](#)
- 3.11. [Dispositivos e Nós de Dispositivos](#)
- 3.12. [Formatos de binários](#)
- 3.13. [Para mais informações](#)

Reescrito por Chris Shumway.

3.1. Sinópsse

O capítulo a seguir cobrirá os comandos básicos e funcionalidades do sistema operacional FreeBSD. A maioria deste material é relevante à qualquer sistema operacional baseado ou similar ao UNIX®. Sinta-se livre para pular este capítulo se você for familiar com o conteúdo deste material. Se você é um novo usuário FreeBSD, então definitivamente você deve ler todo este capítulo cuidadosamente.

Após a leitura deste capítulo, você saberá:

- Como usar os "consoles virtuais" do FreeBSD.
- Como as permissões UNIX funcionam.
- A disposição padrão dos sistemas de arquivos no FreeBSD.
- Como montar e desmontar sistemas de arquivos.
- O que são *daemons*, processos e sinais.
- O que é uma *shell* (interpretador de comandos) e como modificar seu ambiente padrão de *login*.
- Como utilizar editores de texto básicos.
- O que são dispositivos e nós de dispositivos.
- Que formato binário é usado no FreeBSD.
- Como ler as páginas de manual para obter mais informações.

3.2. Consoles virtuais e terminais

O FreeBSD pode ser utilizado de várias maneiras. Uma delas é através de comandos digitados em um terminal de texto. Muito da flexibilidade e do poder de um sistema operacional UNIX® é disponível de imediato ao alcance de suas mãos, quando se utiliza o FreeBSD desta maneira. Esta seção descreve o que são ``terminais" e ``consoles" e como utiliza-los no FreeBSD.

3.2.1. O Console

Se você não configurou o FreeBSD para iniciar um ambiente gráfico automaticamente durante sua inicialização, o sistema irá apresentar a você uma tela de *login* assim que for iniciado, logo depois que os *scripts* de inicialização forem executados. Você verá algo similar a:

```
Additional ABI support:.
Local package initialization:.
Additional TCP options:.

Fri Sep 20 13:01:06 EEST 2002

FreeBSD/i386 (pc3.example.org) (ttyv0)

login:
```

A mensagem deve ser um pouco diferente em seu sistema, mas algo similar será apresentado. As duas últimas linhas são as que nós estamos especialmente interessandos no momento. Na penúltima linha você pode ler:

```
FreeBSD/i386 (pc3.example.org) (ttyv0)
```

Essa linha oferece algumas informações sobre o sistema que acabou de ser iniciado. Você está olhando para um console ``FreeBSD", rodando em arquitetura Intel x86 ou compatível^[1]. O nome desta máquina (toda máquina UNIX tem um nome) é `pc3.example.org`, e agora você está analisando seu console de sistema -- o terminal `ttyv0`.

Finalmente, a última linha será sempre:

```
login:
```

Este é o local onde você deve digitar seu nome de usuário (``username") para logar-se no FreeBSD. A próxima seção descreve como fazê-lo.

3.2.2. Logando-se no FreeBSD

FreeBSD é um sistema multiusuário e de multiprocessamento. Esta é a descrição formal normalmente dada a um sistema que pode ser usado por pessoas diferentes, que simultaneamente executam uma série de programas na mesma estação.

Todo sistema multiusuário requer uma maneira para fazer distinção entre um "usuário" e os outros. No FreeBSD (e todo sistema operacional similar ou baseado em UNIX) isso é garantido com a obrigatoriedade de todo usuário "logar-se" ao sistema antes de poder executar qualquer programa. Todo usuário tem um nome único (o "username") e uma senha secreta pessoal (o "password"). O FreeBSD solicitará estas duas informações para permitir que um usuário se logue ao sistema e execute qualquer programa.

Assim que o FreeBSD é iniciado e acaba de executar seus *scripts* de inicialização[2], será apresentada a você uma tela que solicitará um usuário válido:

```
login:
```

Para compreensão deste exemplo, vamos assumir que o seu nome de usuário é john. Digite *john* na tela de *login* e aperte **Enter**. Em seguida será apresentada a tela que solicita sua "senha" (o "password"):

```
login: john
Password:
```

Digite agora a senha do usuário john e aperte **Enter**. A senha *não é apresentada!* Não é necessário se preocupar com isso agora. É suficiente dizer que o comportamento é este por razões de segurança.

Se você digitou sua senha corretamente, você deve ter se logado no FreeBSD e está pronto para testar todos os comandos disponíveis.

Você deve ver a MOTD, a mensagem do dia, seguida do terminal de comandos (um caracter #, \$, ou %). Isso indica que você se logou com sucesso no FreeBSD.

3.2.3. Múltiplos Consoles

É legal utilizar um console para executar comandos UNIX, mas o FreeBSD pode executar vários programas de uma vez. Ter apenas um console onde os comandos são digitados pode ser considerado um leve desperdício, quando um sistema operacional como o FreeBSD pode rodar dezenas de programas ao mesmo tempo. É aí que os "consoles virtuais" podem se mostrar muito úteis.

O FreeBSD pode ser configurado para oferecer vários consoles virtuais diferentes. Você pode alternar de um para outro apertando um conjunto de teclas em seu teclado. Cada console tem seu próprio canal de saída, e o FreeBSD cuida de redirecionar as entradas do console para cada console virtual e monitor, e também a saída padrão quando você alterna de um para o próximo console virtual.

Combinações especiais de teclas foram reservada pelo FreeBSD para controlar a alternância de consoles[3]. Você pode usar **Alt-F1**, **Alt-F2**, até **Alt-F8** para alternar entre diferentes consoles virtuais no FreeBSD.

Conforme você estiver alternando entre um console e outro, o FreeBSD se responsabiliza por salvar e restaurar o conteúdo de de cada tela. O resultado é uma "ilusão" de múltiplas telas e teclados "virtuais" disponíveis, de forma que você possa

digitar qualquer comando para o FreeBSD executar. Os programas que são iniciados por você em um console virtual não param de ser executados quando o console não está visível. Eles continuam a ser processados quando acontece alternância de console virtual.

3.2.4. O arquivo `/etc/ttys`

A configuração padrão do FreeBSD iniciará oito consoles virtuais. Contudo, esta configuração não é imutável e pode ser facilmente customizada de forma a possibilitar que o sistema seja iniciado com mais ou menos consoles virtuais. O número e definições para os consoles virtuais estão configuradas no arquivo `/etc/ttys`.

Você pode usar o arquivo `/etc/ttys` para configurar consoles virtuais do FreeBSD. Cada linha descomentada neste arquivo (linhas que não começam com o caracter #) contém ajustes para um único terminal ou console virtual. A versão padrão deste arquivo oferece com o FreeBSD nove consoles virtuais, e habilita oito deles. São as linhas que começam com `ttv`:

```
# name getty                type      status    comments
#
ttv0    "/usr/libexec/getty Pc"   cons25   on       secure
# Virtual terminals
ttv1    "/usr/libexec/getty Pc"   cons25   on       secure
ttv2    "/usr/libexec/getty Pc"   cons25   on       secure
ttv3    "/usr/libexec/getty Pc"   cons25   on       secure
ttv4    "/usr/libexec/getty Pc"   cons25   on       secure
ttv5    "/usr/libexec/getty Pc"   cons25   on       secure
ttv6    "/usr/libexec/getty Pc"   cons25   on       secure
ttv7    "/usr/libexec/getty Pc"   cons25   on       secure
ttv8    "/usr/X11R6/bin/xdm -nodaemon" xterm    off     secure
```

Para obter uma descrição detalhada do significado de cada coluna deste arquivo e todas opções disponíveis para os consoles virtuais, consulte a página de manual [ttys\(5\)](#).

3.2.5. Console de usuário em modo mono-usuário

Uma descrição detalhada sobre o que é o "modo mono-usuário" pode ser encontrada em [Seção 7.6.2](#). Vale a pena notar que existe apenas um console disponível quando se está operando o sistema em modo mono-usuário. Não existem consoles virtuais disponíveis. As configurações do console em modo mono-usuário também podem ser definidas no arquivo `/etc/ttys`. Olha na linha que começa com `console`:

```
# name getty                type      status    comments
#
# If console is marked "insecure", then init will ask for the root
password
# when going to single-user mode.
console none                unknown   off     secure
```

Nota: Como indicam os comentários acima da linha `console`, você pode editar esta linha e modificar a entrada `secure` para `insecure`. Se você o fizer, quando o FreeBSD iniciar em modo mono-usuário, ele continuará pedindo a senha do usuário `root`.

Seja cuidadoso ao modificar esta opção para *insecure*. Se você esquecer um dia a senha de `root`, entrar em modo mono-usuário vai ser um pouco complicado. Ainda é possível, mas será um pouco difícil para os usuários ainda não muito confortáveis com o processo de inicialização do FreeBSD e os programas envolvidos.

Notas

- [1] É isso que *i386* significa. Note que mesmo se você não estiver executando o FreeBSD em um processador Intel 386, esta informação ainda será apresentada como *i386*. Não trata-se do tipo de seu processador, mas sim a "arquitetura" do processador, que é apresentada aqui.
 - [2] *Scripts* de inicialização são programas executados automaticamente quando o FreeBSD é iniciado. Sua principal função é configurar tudo que for necessário para o resto do sistema funcionar, e iniciar todos os serviços que você configurou para serem executados em instância de fundo, (em *background* realizado tarefas úteis).
 - [3] Uma descrição técnica e consideravelmente acurada de todos detalhes sobre o console FreeBSD e dispositivos de teclado pode ser encontrada nas páginas de manuais do [syscons\(4\)](#), [atkbd\(4\)](#), [vidcontrol\(1\)](#) e [kbdcontrol\(1\)](#). Não vamos entrar nestes detalhes aqui, mas o leitor interessado pode sempre consultar as páginas de manual para obter explicações mais detalhadas de como as coisas funcionam.
-

3.3. Permissões

FreeBSD, como decendente direto do UNIX® BSD, se baseia em inúmeros conceitos chaves do UNIX. O primeiro e mais citado, é que o FreeBSD é um sistema operacional multi-usuário. O sistema pode gerenciar inúmeros usuários, todos trabalhando simultaneamente em tarefas completamente distintas. O sistema é responsável pelo compartilhamento, e gerencialmente apropriado de pedidos de acesso aos dispositivos de controle de equipamentos, periféricos, de memória, e tempo de processamento, de forma justa entre os usuários.

Pelo fato do sistema poder suportar múltiplos usuários, tudo que o sistema gerencia é composto de um conjunto de permissões que governam quem pode ler, escrever, e executar recursos. Estas permissões são armazenadas como três octetos divididos em três partes, uma para o dono do arquivo, uma para o grupo à que o arquivo pertence, e uma para todos os outros usuários. Sua representação numérica funcionam assim:

Valor	Permissões	Listagem de diretório
0	Sem leitura, sem escrita, sem execução	---
1	Sem leitura, sem escrita, com execução	--x
2	Sem leitura, com escrita, sem execução	-w-
3	Sem leitura, com escrita, com execução	-wx

Valor	Permissões	Listagem de diretório
4	Com leitura, sem escrita, sem execução	<i>r--</i>
5	Com leitura, sem escrita, com execução	<i>r-x</i>
6	Com leitura, com escrita, sem execução	<i>rw-</i>
7	Com leitura, com escrita, com execução	<i>rwX</i>

A opção *-l* pode ser usada junto ao [ls\(1\)](#) para visualizar listagem detalhada do diretório incluindo uma coluna com informações sobre as permissões de cada arquivo, para o dono, grupo, e outros. Por exemplo, a saída do comando `ls -l` em um diretório qualquer, mostrará:

```
% ls -l
total 530
-rw-r--r-- 1 root wheel      512 Sep 5 12:31 myfile
-rw-r--r-- 1 root wheel      512 Sep 5 12:31 otherfile
-rw-r--r-- 1 root wheel    7680 Sep 5 12:31 email.txt
...
```

Aqui está a forma como a primeira coluna do `ls -l` é dividida:

```
-rw-r--r--
```

O primeiro caracter (o mais à esquerda) indica que se este é um arquivo normal, se é um diretório, um dispositivo especial, um *socket* ou qualquer outro pseudo-dispositivo especial. Nesse caso, o *-* indica que trata-se de um arquivo regular. Os próximos três caracteres, *rw-* neste exemplo, indicam as permissões para o dono do arquivo. Os próximos três caracteres, *r--*, indicam as permissões para o grupo ao qual o arquivo pertence. Os três caracteres finais, *r--*, indicam as permissões para o resto do mundo. Um hífen indica que a permissão está desligada. No caso deste arquivo, o dono pode ler e escrever neste arquivo, o grupo pode ler o arquivo e o resto do mundo pode também apenas ler o arquivo. De acordo com a tabela acima, as permissões para o arquivo seriam *644*, onde cada dígito representa as três partes das permissões do arquivo.

Isso é muito bonito e legal, mas como o sistema controla as permissões nos dispositivos? FreeBSD na verdade trata a maioria dos dispositivos de *hardware* como arquivos onde os programas podem ler, abrir, e escrever dados, como se fosse qualquer outro arquivo. Estes arquivos de dispositivos são armazenados no diretório `/dev`.

Diretórios também são tratados como arquivos comuns. Eles tem permissões de leitura, escrita e execução. O *bit* de execução em um diretório tem uma sutil diferença quando comparado a um arquivo. Quando um diretório é definido como executável, significa que ele pode ser explorado, ou seja, que é possível entrar neste diretório (``cd``). Também significa que dentro do diretório, pode-se acessar arquivos cujos nomes são conhecidos (de acordo com as permissões dos próprios arquivos, é claro).

Em particular, para realizar a listagem de um diretório, é necessário ter permissões de leitura no mesmo, enquanto para apagar um arquivo cujo nome já é conhecido, é necessário ter permissões de escrita e execução no diretório onde está o arquivo.

Existem outros bits de permissões, mas eles são utilizados em circunstâncias especiais como binários como *setuid* e diretórios com opção *sticky*. Se você deseja obter mais informações sobre as permissões de arquivo e como defini-las, dê uma olhada na página de manual do [chmod\(1\)](#).

3.3.1. Permissões simbólicas

Contribuição de Tom Rhodes.

Permissões simbólicas, as vezes se referem à expressões de permissões, ou o uso de caracteres em substituição aos valores octais que atribuem permissões aos arquivos e diretórios. Expressões simbólicas usam a sintaxe (quem) (ação) (permissões), onde os seguintes valores estão disponíveis:

Opção	Letra	Representação
(quem)	u	Usuário
(quem)	g	Grupo dono
(quem)	o	Outros
(quem)	a	Todos ("o mundo")
(ação)	+	Adicionar permissões
(ação)	-	Remover permissões
(ação)	=	Definir permissões explicitamente
(permissões)	r	Leitura
(permissões)	w	Escrita
(permissões)	x	Execução
(permissões)	t	Bit de <i>sticky</i>
(permissões)	s	Define UID ou GID

Estes valores são usados com o comando [chmod\(1\)](#) exatamente como da forma anterior, mas sem letras. Por exemplo, você pode usar o seguinte comando para impedir que outros usuários deixem de acessar determinado *ARQUIVO*:

```
% chmod go= ARQUIVO
```

Pode-se fazer uso de uma lista separado por vírgula, quando se desejar que mais de uma modificação seja realizada no arquivo. Por exemplo, o comando à seguir remove as permissões de escrita para o grupo e ``todo o mundo'', no *ARQUIVO*, e então adiciona permissões de execução à todos:

3.4. Estrutura de Diretórios

Conhecer a hierarquia de diretórios do FreeBSD é fundamental para se obter entendimento total do sistema. O conceito mais importante para se assimilar é o conceito do diretório raiz, o ``/". Esse diretório é montado inicialmente no momento em que o sistema é carregado, e ele contém o sistema base necessário para preparar a entrada em modo de operação multi-usuário do sistema operacional. O diretório raiz contém ainda pontos de montagem para cada outro sistema de arquivos que você queira montar.

Um ponto de montagem é um diretório onde sistemas de arquivos adicionais podem ser acoplados ao sistema de arquivos raiz. Pontos de montagem padrão incluem `/usr`, `/var`, `/mnt`, e `/cdrom`. Esses diretórios normalmente são encontrados como entradas dispostas no arquivo `/etc/fstab`. O `/etc/fstab` é uma tabela com vários sistemas de arquivos e pontos de montagem que o sistema usa como referência. A maioria dos sistemas de arquivos no `/etc/fstab` são montados automaticamente no momento da inicialização, à partir do *script* [rc\(8\)](#) a não ser que eles contenham a opção *noauto* definida. Consulte a página de manual do [fstab\(5\)](#) para obter mais informações sobre o formato do arquivo `/etc/fstab` e seu conteúdo.

Uma descrição completa da hierarquia do sistema de arquivos está disponível na página de manual [hier\(7\)](#). Por hora, uma breve descrição dos diretórios mais comuns será abordada:

Diretório	Descrição
/	Diretório raiz do sistema de arquivo.
/bin/	Utilitários do usuário, fundamentais para os ambientes de multi-usuário e mono-usuário.
/boot/	Arquivos e programas de configuração que são usados durante o processo de inicialização do sistema operacional.
/boot/default/	Arquivos de configuração padrão do sistema de inicialização, veja a página de manual loader.conf(5) .
/dev/	Dispositivos de controle; veja intro(4) .
/etc/	Arquivos e scripts de configuração do sistema.
/etc/default/	Arquivos de configuração padrão do sistema; veja rc(8) .
/etc/mail/	Arquivos de configuração para os agentes de transferência de correio, como o sendmail(8) .
/etc/namedb/	Arquivos de configuração do named; veja named(8) .
/etc/periodic/	Scripts que são executados diariamente, semanalmente e mensalmente por meio do cron(8) ; veja periodic(8) .

Diretório	Descrição
<code>/etc/ppp/</code>	Arquivos de configuração do <code>ppp</code> ; veja ppp(8) .
<code>/mnt/</code>	Diretório vazio normalmente utilizado pelos administradores de sistemas como ponto de montagem temporário.
<code>/proc/</code>	Sistema de arquivos de processos; veja procfs(5) e mount_procfs(8) .
<code>/root/</code>	Diretório home da conta de usuário <code>root</code> .
<code>/sbin/</code>	Utilitários de programas e administração, fundamentais para os ambientes de multi-usuário e mono-usuário.
<code>/stand/</code>	Programas usados no ambiente independente.
<code>/tmp/</code>	Arquivos temporários, normalmente um diretório montado em memória por meio do mfs(8) (o conteúdo do <code>/tmp</code> normalmente <i>NÃO</i> é preservado durante uma reinicialização do sistema).
<code>/usr/</code>	A maioria dos utilitários e aplicações de usuários.
<code>/usr/bin/</code>	Utilitários comuns, ferramentas de programação e aplicações.
<code>/usr/include/</code>	Arquivos C padrão de inclusão.
<code>/usr/lib/</code>	Bibliotecas.
<code>/usr/libdata/</code>	Arquivos de dados de utilitários variados.
<code>/usr/libexec/</code>	Daemons & utilitários do sistema (executados por outros programas).
<code>/usr/local/</code>	Executáveis locais, bibliotecas, etc. Também utilizado como destino padrão para o sistema de Ports. Dentro de <code>/usr/local</code> , a disposição hierárquica geral é definida pelo hier(7) como o definido pelo <code>/usr</code> . Exceto o diretório de páginas de manuais que são diretamente dispostas sob <code>/usr/local</code> ao invés de serem sobre <code>/usr/local/share</code> , e a documentação das aplicações de terceiros, que ficam sob <code>share/doc/port</code> .
<code>/usr/obj/</code>	Arquivo de destino dos objetos binários específicos para cada arquitetura, construídos à partir da árvore <code>/usr/src</code> .
<code>/usr/ports</code>	A Coleção de Ports do FreeBSD (opcional).
<code>/usr/sbin/</code>	Daemons & utilitários do sistema (executados por usuários).
<code>/usr/share/</code>	Arquivos independentes de arquitetura.
<code>/usr/src/</code>	Arquivos de fontes locais e/ou BSD.
<code>/usr/X11R6/</code>	Executáveis da distribuição do X11R6, bibliotecas, etc (opcional).
<code>/var/</code>	Arquivos de log de múltiplos propósitos, arquivos temporários, de transição, e arquivos de bobina (spool).
<code>/var/log/</code>	Arquivos de logs variados do sistema.
<code>/var/mail/</code>	Arquivos de caixa de correio dos usuários.
<code>/var/spool/</code>	Diretórios de spool de impressão e de correio.
<code>/var/tmp/</code>	Arquivos temporários que são mantidos entre reinicializações do sistema.
<code>/var/yp</code>	Mapas NIS.

3.5. Organização de disco

A menor unidade organizacional que o FreeBSD utiliza para definir os arquivos é o nome de arquivo. Nome de arquivos são ``case-sensitive'', ou seja fazem distinção entre maiúsculas e minúscilas, o que significa que `readme.txt` e `README.TXT` são dois arquivos distintos. FreeBSD não utiliza a extensão do arquivo (`.txt`) para determinar se o arquivo é um programa ou um documento, ou ainda qualquer outro tipo de dado.

Arquivos são armazenados em diretórios. Um diretório pode não conter arquivo algum, ou pode conter centenas de arquivos. Um diretório pode ainda conter outros diretórios, permitindo que você construa hierarquias inteiras de diretórios, uns dentro dos outros. Isso torna a organização de seus dados muito mais fácil.

Arquivos e diretórios são identificados por seu nome, seguido por uma barra dianteira, /, seguido de outros nomes de diretórios, conforme necessário. Se você tem o diretório `foo`, e dentro dele o diretório `bar`, que por sua vez contém o arquivo `readme.txt`, então o nome completo, ou o *path* para este arquivo é `foo/bar/readme.txt`.

Diretórios e arquivos são armazenados em um sistema de arquivos. Cada sistema de arquivos contém exatamente um diretório de nível hierárquico interior, chamado de *diretório raíz* para o sistema de arquivos. Este diretório raíz pode conter outros diretórios.

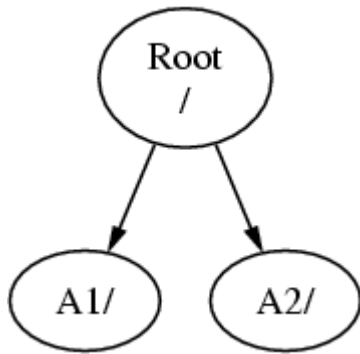
Certamente isso tudo é similar à qualquer outro sistema operacional que você já usou. Existem algumas diferenças básicas; por exemplo, DOS usa a barra invertida, \, para separar qualquer arquivo ou nome de diretório, enquanto o Mac OS® utiliza o `:`.

O FreeBSD não utiliza letras de dispositivos ou qualquer outro tipo de nome para os dispositivos do sistema de arquivos. Você jamais escreveria `c:/foo/bar/readme.txt` no FreeBSD.

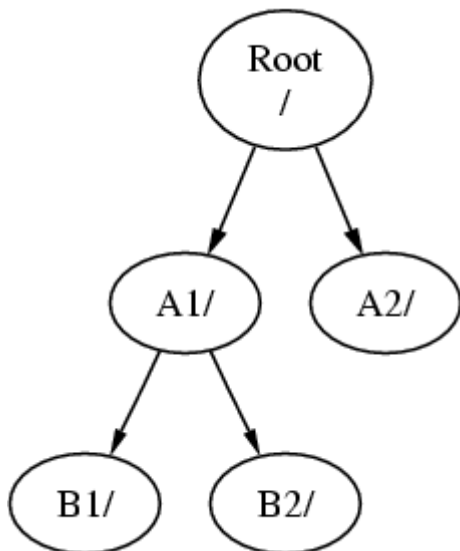
Ao invés disso, um sistema de arquivo é definido como *sistema de arquivos raíz*. O diretório raíz do sistema de arquivo raíz é o `/`. Todo outro sistema de arquivos é montado (*mounted*) sob este sistema de arquivos. Não importa quantos discos existam no seu sistema FreeBSD, cada diretório é apresentado como parte do disco.

Suponha que você tenha três sistemas de arquivos, chamados *A*, *B*, e *C*. Cada sistema de arquivos tem seu próprio diretório raíz, que contém outros dois diretórios chamados *A1*, *A2* (e da mesma forma *B1*, *B2* e *C1*, *C2*).

Assuma *A* como sistema de arquivos raíz. Se você usou o comando `ls` para visualizar o conteúdo deste diretórios, você veria dois subdiretórios, *A1* e *A2*. A árvore de diretórios ficaria assim:

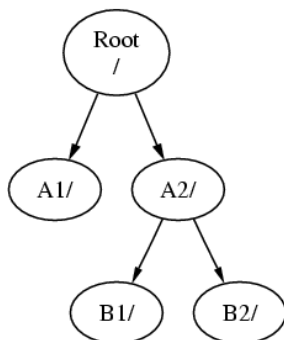


Todo sistema de arquivos deve ser montado em um diretório disposto em outro sistema de arquivos. Então, suponha que você tenha montado o sistema de arquivos *B* no diretório *A1*. O diretório raiz do *B* substitui *A1*, e os diretórios em *B* se tornam apropriadamente disponíveis:



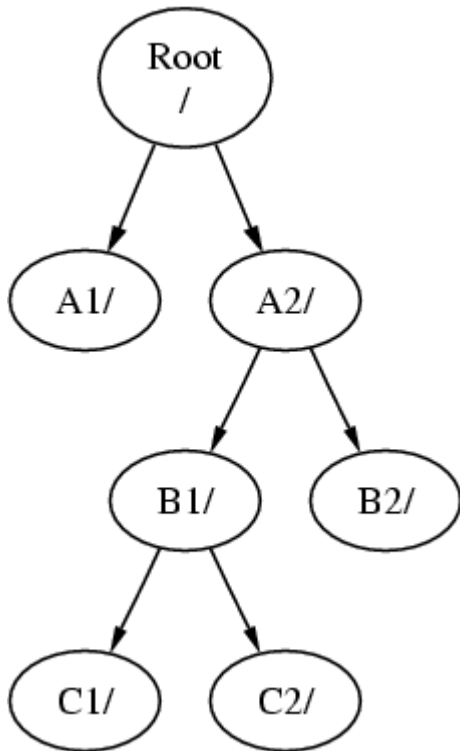
Quaisquer arquivos que estejam nos diretórios *B1* ou *B2* podem ser acessados com o *path* */A1/B1* ou */A1/B2* conforme necessário. Qualquer arquivo que estivesse no */A1* foi temporariamente escondido. Eles reaparecerão se *B* for *desmontado* de *A*.

Se *B* foi montado em *A2* então o diagrama será parecido com:

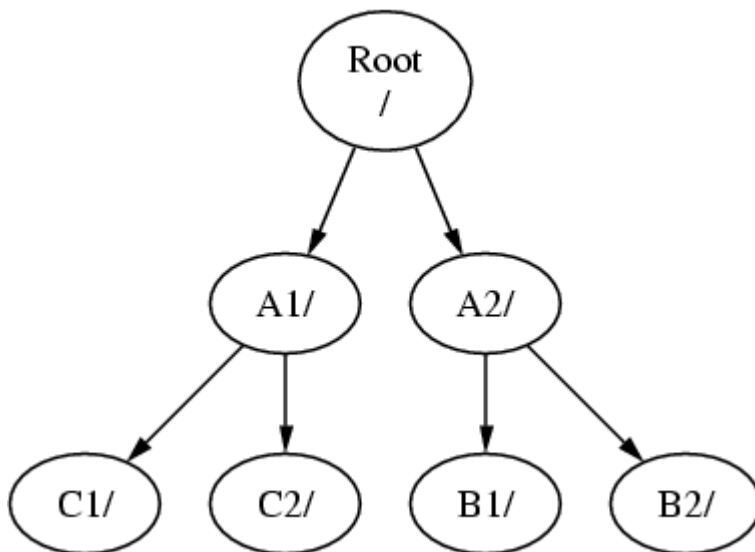


e os *paths* seriam /A2/B1 e /A2/B2 respectivamente.

Os sistemas de arquivos podem ser montados em sobreposição à outros. Prosseguindo com o último exemplo, o sistema de arquivo *C* poderia ser montado sobre o diretório *B1* no sistema de arquivos *B*, levando ao seguinte arranjo:



Ou *C* poderia ser montado diretamente no sistema de arquivos *A*, sob o diretório *A1*:



Se você tem familiaridade com o DOS, a teoria é parecida, mas não idêntica, ao comando `join`.

Normalmente isso não é algo com que você tenha que se preocupar. Tipicamente basta criar os sistemas de arquivos ao instalar o FreeBSD e decidir onde monta-los, e depois nunca mais modifica-los, a não ser que você adicione um novo disco.

É completamente possível ter um sistema de arquivos raíz, e não criar mais nenhum. Existem alguns problemas com esta abordagem e uma grande desvantagem.

Benefícios de sistemas de arquivos múltiplos

- Sistemas de arquivos distintos podem ter *opções de montagem* distintas. Por exemplo, com planejamento cuidadoso, o sistema de arquivos raíz pode ser montado como apenas-leitura, criando um ambiente onde será impossível apagar ou editar arquivos críticos inadvertidamente. Separando sistemas de arquivos que podem ser escritos por todos usuários, como o `/home`, de outros sistemas de arquivos, ainda permite que eles sejam montados com *nosuid*; esta opção previne que os bits *suid/guid* sejam atribuídos em arquivos executáveis, não tendo mais efeito no sistema de arquivos em questão, possivelmente complementando a segurança do ambiente.
- O FreeBSD otimiza a disposição dos arquivos de forma automática, dependendo de como o sistema de arquivos está sendo usado. Portanto um sistema de arquivos que contém muitos arquivos pequenos que são freqüentemente escritos terão regras de otimização distintas daqueles sistemas de arquivos com poucos, mas grandes arquivos. Mantendo um único sistema de arquivos enorme, esta otimização se torna inviável.
- Os sistemas de arquivos do FreeBSD são muito robustos, mesmo em situações de queda de energia. Contudo, uma interrupção de energia em um ponto crítico pode danificar parcialmente a estrutura do sistema de arquivos. Dividir os dados em múltiplos sistemas de arquivos aumenta a possibilidade do sistema não ficar indisponível, tornando mais fácil a restauração de cópias de segurança, caso necessário.

Benefícios de um único sistema de arquivos

- Sistemas de arquivos tem tamanho fixo. Se você criar um sistema de arquivos ao instalar o FreeBSD e definir tamanho específico, posteriormente você pode descobrir que tem a necessidade de uma partição maior. Não é uma tarefa fácil resolver isso, sem fazer cópias de segurança e recriar o sistema de arquivos com o novo tamanho e depois restaurar a cópia.

Importante: A partir do FreeBSD 4.4 o comando [growfs\(8\)](#) passou a ser parte do sistema, o que possibilita que o tamanho dos sistemas de arquivos sejam incrementados na hora, acabando com esta limitação.

Sistemas de arquivos são contidos em partições. Isso não significa a mesma coisa do que a forma de utilização do termo partição anteriormente neste capítulo, por causa da herança UNIX® do FreeBSD. Cada partição é identificada com uma letra que varia de *a* à *h*. Cada partição possui um sistema de arquivos, o que significa que sistemas de arquivos normalmente são identificados ou por seu ponto de montagem na hierarquia do sistema de arquivos, ou pela letra disposta no nome de cada partição.

O FreeBSD ainda usa espaço de disco para o "espaço de troca", ou *swap*. Swap oferece características de *memória virtual* ao FreeBSD. Isso permite que seu computador se comporte como se tivesse mais memória disponível do que na realidade tem. Quando o FreeBSD excede a memória disponível ele move alguns trechos de dados que não estão sendo utilizados para o espaço de troca, a *swap*, e move esses dados de volta pra memória (descarregando alguma outra coisa para liberar recursos) quando necessário.

Algumas partições mantêm algumas convenções associadas.

Partição	Convenção
<i>a</i>	Normalmente contém o sistema de arquivos raiz
<i>b</i>	Normalmente contém a partição de swap (troca)
<i>c</i>	Normalmente tem o mesmo tamanho da partição inclusa. Isso permite que alguns utilitários que precisam trabalhar na partição toda (por exemplo, um identificador de blocos defeituosos) funcionem corretamente na partição <i>c</i> . Você não vai criar um sistema de arquivos de verdade nesta partição.
<i>d</i>	A partição <i>d</i> costumava ter um significado especial associado com ela, contudo agora ele desfez. Hoje em dia algumas ferramentas podem continuar se comportando de forma estranha se indicarmos que ele trabalhe na partição <i>d</i> , de forma que o sysinstall não cria mais esta partição.

Cada partição que contém um sistema de arquivos é armazenada em um tipo especial de partição que o FreeBSD chama de *slice*. Este termo é utilizado pelo FreeBSD para fazer menção ao que, mais uma vez, costumava ser chamado de partição, e isso se deve à base UNIX por trás do sistema FreeBSD. Estas partições ("slices") são numeradas, iniciando do 1, até o 4.

Número de partições seguem o nome do dispositivo adicionado de um prefixo com o carácter *s*, começando do 1. Portanto "*da0s1*" é a primeira *slice* de um disco, mas é possível que existam partições lógicas dentro de cada partição física do tipo apropriado. Estas partições extendidas são numeradas à partir do 5, de forma que "*ad0s5*" seja a primeira partição extendida do primeiro disco IDE. Estes dispositivos são utilizados por sistemas de arquivos que ocupam esse tipo de partição.

Partições, dispositivos físicos, "perigosamente dedicados" e outros dispositivos contém *partições* que costumam ser representadas como letras, começando de *a* até *h*. Estas letras são atribuídas ao nome dos dispositivos, de forma que "*da0a*" indique a partição no primeiro dispositivo, que na realidade é "perigosamente dedicado". "*ad1s3e*" é a quinta partição na primeira *slice* do segundo dispositivo de disco IDE.

Finalmente, cada disco no sistema está identificado. Um nome de disco começa com um código que indica o tipo do disco em questão, e em seguida um número, que indica que disco é esse. Diferente das partições, a numeração dos discos inicia em 0. Informações usuais que você encontrará listados em [Tabela 3-1](#).

Ao se referir a uma partição, o FreeBSD requer que você nomeie cada *slice* e disco que contenha a partição em questão, e que ao se referir à *slice* você também identifique o nome do disco. Faça isso listado o nome do disco, *s*, o número da *slice*, e em seguida a letra da partição. Exemplos são apresentados em [Exemplo 3-1](#).

[Exemplo 3-2](#) apresenta o modelo conceitual das disposições de disco, o que pode ajudar a tornar as coisas mais claras.

Para instalar o FreeBSD, primeiro você deve configurar os *slices* de disco, e depois criar as partições em cada *slice* destinada ao FreeBSD, e criar um sistema de arquivo (ou *swap*) em cada partição, e decidir onde o sistema de arquivos será montado.

Tabela 3-1. Códigos de Dispositivos de Disco

Código	Significado
ad	disco ATAPI (IDE)
da	disco SCSI de acesso direto
acd	CDROM ATAPI (IDE)
cd	CDROM SCSI
fd	Disquete

Exemplo 3-1. Modelo de disco, partição e nomes

Nome	Significado
<i>ad0s1a</i>	Primeira partição (<i>a</i>) na primeira fatia de disco (<i>slice - s1</i>) no primeiro disco IDE (<i>ad0</i>).
<i>da1s2e</i>	Quinta partição (<i>e</i>) na segunda fatia de disco (<i>slice - s2</i>) no segundo disco SCSI (<i>da1</i>).

Exemplo 3-2. Modelo conceitual de um disco

Esse diagrama apresenta a abordagem do FreeBSD para o primeiro disco IDE ligado ao sistema. Assuma que o disco tem tamanho de 4 GB e contém duas partições de 2 GB (partições DOS). A primeira *slice* contém um disco DOS, o *c:*, e a segunda *slice* contém a instalação do FreeBSD. Esta instalação FreeBSD de exemplo tem três partições, e uma *swap*.

As três partições terão um sistema de arquivos cada. A partição *a* será utilizada pelo sistema de arquivos raiz, *e* será utilizada pela hierarquia de diretório */var*, e *f* pela hierarquia de diretório */usr*.

First slice,
called ad0s1
(C: in DOS)



Second slice,
called ad0s2

Parti
not n
calle

(not to scale)

3.6. Montando e desmontando sistemas de arquivos

O sistema de arquivos pode ser melhor assimilado se for visualizado como uma árvore, cuja raiz é `/`. `/dev`, `/usr`, e outros diretórios na raiz, são os ramos da árvore, que podem ainda ter seus próprios ramos, como `/usr/local`, e assim por diante.

Existem vários motivos para armazenar alguns destes diretórios em sistemas de arquivos distintos. `/var` contém os diretórios `log/`, `spool/`, e vários tipos de arquivos temporários, e como tal, pode facilmente encher. Encher o sistema de arquivos raiz não é uma boa idéia, de forma que separar o `/var` do `/` é sempre favorável.

Outra razão usual para manter certas árvores de diretórios em outros sistemas de arquivos é que eles podem ser alojados em discos físicos separados, ou em um disco virtual separado como pontos de montagem [NFS](#) ou dispositivos de CDROM.

3.6.1. O arquivo `fstab`

Durante o [processo de inicialização](#), os sistemas de arquivos listados no `/etc/fstab` são automaticamente montados (a não ser que eles tenham a opção `noauto` definida).

O arquivo `/etc/fstab` contém uma lista de linhas no seguinte formato:

```
device          /mount-point fstype  options      dumpfreq     passno
device
```

Um nome de dispositivo (que deve existir), conforme explicado em [Seção 12.2](#).

mount-point

Um diretório (que deve existir), onde o sistema de arquivos é montado.

fstype

O tipo de sistema de arquivo que será passado ao [mount\(8\)](#). O sistema de arquivos padrão do FreeBSD é *ufs*.

options

Pode ser *rw* para sistemas de arquivos com opções de leitura e escrita, ou *ro* para apenas leitura, seguidos de qualquer outra opção necessária. Uma opção bastante usual é *noauto* para os sistemas de arquivos que normalmente não devem ser montados automaticamente pelo processo de inicialização. Outras opções são listadas na página de manual do [mount\(8\)](#).

dumpfreq

Essa opção é utilizada pelo [dump\(8\)](#) para determinar que sistema de arquivos devem sofrer o processo de [dump\(8\)](#) (o [dump\(8\)](#) é uma ferramenta para cópia de segurança). Se o campo não estiver presente, o valor zero é assumido.

passno

Determina a ordem que os sistemas de arquivos devem ser verificados. Sistemas de arquivos cuja verificação pode ser omitida, devem ter seu *passno* definidos em zero. O sistema de arquivos raiz (que deve ser verificado antes de todos os outros) deve ter seu *passno* definido como um, e os outros sistemas de arquivos podem ter esta opção definida com um valor maior. Se mais de um sistema de arquivos disporem do mesmo número, o [fsck\(8\)](#) tentará fazer as verificações de forma paralela, se possível.

3.6.2. O comando `mount`

O comando [mount\(8\)](#) é a ferramenta usada para montar sistemas de arquivos.

Em sua forma mais básica, basta usar:

```
# mount dispositivo ponto de montagem
```

Existem inúmeras opções, e estão todas dispostas na página de manual do [mount\(8\)](#), mas as mais comuns são:

Opções de montagem

-a

Monta todos os sistemas de arquivos listados no `/etc/fstab`. Exceto aqueles assinalados com a opção `noauto`, excluídos pela opção `-t`, ou aqueles que já estiverem montados.

-d

Faz tudo, exceto a efetivação da chamada de sistema para a montagem. Essa opção é muito útil, em conjunto com a `-v` para descobrir o que o [mount\(8\)](#) está tentando fazer.

-f

Força a montagem de um sistema de arquivos que não tem garantia de integridade (isso é perigoso), ou força a revogação dos direitos de escrita ao alterar o estado de montagem de um sistema de arquivos que esteja em modo somente leitura, para o modo leitura e escrita.

-r

Monta o sistema de arquivos como somente leitura. Essa opção é idêntica ao argumento *rdonly* na opção *-o*.

-t tipo de sistema de arquivos

Monta o sistema de arquivos em questão, como do tipo em atribuído, ou monta apenas sistemas de arquivos do tipo atribuído se for usado em conjunto com a opção *-a*.

"*ufs*" é o tipo padrão de sistema de arquivos.

-u

Atualiza opções de montagem no sistema de arquivos.

-v

Torna-se verboso.

-w

Monta o sistema de arquivos como leitura e escrita.

A opção *-o* pode utilizar uma lista de opções separadas por vírgula, das quais, as principais são:

nodev

Não interpreta dispositivos especiais no sistema de arquivos em questão. Esta é uma opção útil para segurança.

noexec

Não permite execução de binários neste sistema de arquivos. Também é uma opção muito útil para segurança.

nosuid

Não interpreta opções de *setuid* ou *setgid* no sistema de arquivos. É uma opção útil para segurança.

3.6.3. O comando `umount`

O comando `umount(8)` recebe como parâmetro o ponto de montagem, o dispositivo, a opção *-a* ou a opção *-A*.

Todas estas maneiras podem ter também as opções *-f* para forçar desmontagem, e *-v* para ser verboso. Esteja atento ao fato que a opção *-f* geralmente não é uma boa idéia.

Desmontar os sistemas de arquivos forçozamente pode causar dano aos dados do sistema de arquivos.

`-a` and `-A` são usadas para desmontar todos os sistemas de arquivos; este comportamento é possivelmente modificado em conjunto com a opção `-t`. `-A`, contudo, não tenta

3.7. Processos

O FreeBSD é um sistema operacional multi-tarefa. Isso significa que ele cria um ambiente onde os recursos computacionais são divididos de forma à parecer que mais de um programa é executado ao mesmo tempo. Cada processo sendo executado à qualquer momento, é chamado de um *processo*. Todo comando que você executar, iniciará ao menos um novo processo, e existem vários processos do sistema que estão sempre em execução, mantendo a funcionalidade do ambiente como um todo.

Cada processo é identificado de forma única com um número, que é chamado de identificação do processo, ou *process ID*, ou então a forma abreviada e mais comum, *PID*, e, assim como os arquivos, cada processo também tem um dono e grupo. As informações sobre dono e grupo são utilizadas com o intuito de determinar que arquivos e dispositivos o processo pode abrir, utilizando as permissões discutidas anteriormente. A maioria dos processos tem ainda um processo pai. O processo pai é o processo que o iniciou. Por exemplo, quando você digita um comando em uma *shell*, então essa *shell* é um processo, e todo comando executado é outro processo. Todo comando executado desta forma terá sua *shell* como o processo pai. A única exceção é um processo especial chamado de [init\(8\)](#). O `init` é sempre o primeiro processo do sistema, portanto seu PID é sempre 1. O `init` é executado automaticamente pelo *kernel* quando o FreeBSD é iniciado.

Dois comandos são particularmente úteis para verificar os processos ativos no sistema, o [ps\(1\)](#) e o [top\(1\)](#). O comando `ps` é usado para apresentar uma lista estática dos processos sendo executados no momento, e pode apresentar seu PID, quanta memória o processo está utilizando, a linha de comando que o iniciou, e assim por diante. O comando `top` apresenta todos os processos em execução, e atualiza a forma como os apresenta na tela a cada poucos segundos, de forma que você possa acompanhar o que o computador está fazendo.

Por padrão o `ps` apenas apresenta os comandos que estão sendo executados por você, e cuja propriedade também seja sua. Por exemplo:

```
% ps
  PID TT  STAT      TIME COMMAND
  298 p0  Ss      0:01.10 tcsh
 7078 p0  S       2:40.88 xemacs mdoc.xsl (xemacs-21.1.14)
37393 p0  I       0:03.11 xemacs freebsd.dsl (xemacs-21.1.14)
48630 p0  S       2:50.89 /usr/local/lib/netscape-linux/navigator-linux-4.77.bi
48730 p0  IW      0:00.00 (dns helper) (navigator-linux-)
72210 p0  R+      0:00.00 ps
   390 p1  Is      0:01.14 tcsh
 7059 p2  Is+     1:36.18 /usr/local/bin/mutt -y
```

```

6688 p3 IWs      0:00.00 tcsh
10735 p4 IWs      0:00.00 tcsh
20256 p5 IWs      0:00.00 tcsh
  262 v0 IWs      0:00.00 -tcsh (tcsh)
  270 v0 IW+      0:00.00 /bin/sh /usr/X11R6/bin/startx -- -bpp 16
  280 v0 IW+      0:00.00 xinit /home/nik/.xinitrc -- -bpp 16
  284 v0 IW       0:00.00 /bin/sh /home/nik/.xinitrc
  285 v0 S        0:38.45 /usr/X11R6/bin/sawfish

```

Como você pode acompanhar neste exemplo, a saída do comando [ps\(1\)](#) é organizada em várias colunas. *PID* é a identificação do processo discutida anteriormente. PIDs são atribuídos à partir do 1, e vão até 99999, e reiniciam a contagem quando ultrapassa esse valor. A coluna *TT* posta em que terminal o programa esta sendo executado, e pode seguramente ser ignorado no momento. *STAT* apresenta o estado atual do programa, e, mais uma vez, vamos ignora-lo por hora. *TIME* indica a quantidade de tempo que o programa está em execução na CPU em questão -- normalmente não é o tempo total desde que o processo foi iniciado, e sim tempo de execução, uma vez que vários programas ficam um bom tempo esperando que algo aconteça para eles utilizarem tempo de processamento da CPU. Finalmente, *COMMAND* é a linha de comando utilizada para executar o programa em questão.

[ps\(1\)](#) suporta uma série de opções distintas que modificam as informações apresentadas. Um dos conjuntos mais úteis de opções é *auxww*. *a* apresenta informações sobre todos os processos em execução, não apenas os de sua propriedade. *u* apresenta o nome do usuário que é o dono do processo, bem como o uso de memória do mesmo. *x* apresenta informações sobre processos *daemon*, e *ww* faz o [ps\(1\)](#) apresentar a linha de comando completa, ao invés de um trecho compactado para caber na tela.

A saída do [top\(1\)](#) é similar. Uma sessão modelo teria a seguinte aparência:

```

% top
last pid: 72257;  load averages:  0.13,  0.09,  0.03    up 0+13:38:33
22:39:10
47 processes:  1 running, 46 sleeping
CPU states: 12.6% user,  0.0% nice,  7.8% system,  0.0% interrupt,
79.7% idle
Mem: 36M Active, 5256K Inact, 13M Wired, 6312K Cache, 15M Buf, 408K
Free
Swap: 256M Total, 38M Used, 217M Free, 15% Inuse

  PID USERNAME PRI NICE SIZE     RES STATE   TIME  WCPU   CPU
COMMAND
72257 nik          28   0 1960K 1044K RUN      0:00 14.86%  1.42% top
 7078 nik          2   0 15280K 10960K select   2:54  0.88%  0.88% xemacs-
21.1.14
  281 nik          2   0 18636K 7112K select   5:36  0.73%  0.73% XF86_SVGA
  296 nik          2   0 3240K 1644K select   0:12  0.05%  0.05% xterm
48630 nik          2   0 29816K 9148K select   3:18  0.00%  0.00% navigator-
linu
  175 root          2   0  924K   252K select   1:41  0.00%  0.00%
syslogd
 7059 nik          2   0 7260K 4644K poll     1:38  0.00%  0.00% mutt
...

```

A saída está dividida em duas seções. O cabeçalho (as primeiras cinco linhas) apresenta o PID do último processo executado, a média de carga do sistema (que mede o quanto o

sistema está ocupado), e o tempo de atividade ininterrupta do sistema (tempo desde a última inicialização - o *uptime*). As outras informações do cabeçalho indicam quantos processos (47 neste caso) estão ativos, quando de memória e *swap* foi utilizado, e quanto tempo o sistema está gastando em estados diferentes de uso da CPU.

Abaixo está uma série de colunas com informações similares à saída do [ps\(1\)](#). Como antes, você pode ver o PID, o usuário, o tempo de CPU utilizado, e o comando executado. [top\(1\)](#) também mostra por padrão a quantidade de memória utilizada pelo processo. Essa informação é dividida em dois grupos, uma para o tamanho total de memória, e um para o tamanho residente -- tamanho total indica quanta memória a aplicação já precisou, e o tamanho residente é a quantidade em uso no momento. Nesse exemplo você pode ver que o **Netscape®** precisou quase de 30 MB de RAM, mas no momento está usando apenas 9 MB.

[top\(1\)](#) atualiza automaticamente as informações a cada dois segundos, o que pode ser modificado com a opção *s*.

3.8. Daemons, sinais e controle de processos

Ao usar um editor, é fácil controlá-lo, indicar que ele deve carregar arquivos, e assim por diante. Você pode fazê-lo pois o editor oferece algumas opções para fazê-lo, e por que o editor é diretamente interligado a um *terminal*. Alguns programas não são projetados de forma a utilizar informações contínuas do usuário, e assim ele se desconecta do terminal em sua primeira oportunidade. Por exemplo, um servidor HTTP passa o tempo todo respondendo requisições da rede, e normalmente ele não necessita intervenção alguma do usuário. Programas que transportam mensagens de correio de um local para outro, são outro exemplo dessa classe de aplicações.

Chamamos estes programas de *daemons*. Daemons são personagens da mitologia Grega; não são bons, nem maus, eram apenas espíritos assistentes que, na maior parte do tempo, realizavam tarefas úteis para a humanidade. É por isso que o mascote do BSD, há um bom tempo, é aquele daemon simpático que usa tênis e carrega um tridente.

Existe uma convenção para batizar os programas que normalmente são executados como daemon, que é terminar o nome do programa com um ``d". **BIND** é o Daemon de Nomes Internet de Berkeley, (*Berkeley Internet Name Daemon* - o programa que o executa, é chamado *named*), O servidor WWW **Apache** é chamado de *httpd*, o daemon de impressão em linha é o *lpd* e assim sucessivamente. Esta é uma convenção, e não uma regra de rápida assimilação; por exemplo, o daemon principal para a aplicação **Sendmail** é chamado *sendmail*, e não *maild*, como você poderia ter imaginado inicialmente.

Algumas vezes será necessário se comunicar com um processo daemon. Essa comunicação é chamada de *sinais*, e você pode se comunicar com o daemon (ou

qualquer outro processo em execução) enviando-o um sinal. Existem uma série de sinais diferentes que podem ser enviados--alguns dos quais tem um significado específico, e outros são interpretados pela aplicação, e a documentação dessa aplicação indicará o que ela faz quando recebe este tipo de sinal. Você só pode enviar um sinal para os processos que te pertencem. Se você tentar enviar sinais para um processo que não é seu, com [kill\(1\)](#) ou [kill\(2\)](#) terá permissão negada. A única exceção é o usuário `root`, que pode enviar sinais para os processos de todos.

O FreeBSD em certos casos também enviará sinais às aplicações. Se uma aplicação for mal escrita e tentar acessar endereços de memória que ele não deve, o FreeBSD envia o sinal de violação de segmento ao processo (*Segmentation Violation*), conhecido como *SIGSEGV*. Se alguma aplicação usou a chamada de sistema [alarm\(3\)](#) para solicitar que seja alertada depois de certo período de tempo, o sistema envia o sinal de alerta (*SIGALRM*) ao processo, e assim por diante.

Dois sinais podem ser usados para interromper um processo, *SIGTERM* e *SIGKILL*. *SIGTERM* é uma maneira educada de matar um processo; o processo pode *acatar* o sinal, entender que você deseja que ele deixe de ser executado, fechar qualquer arquivos de logs que ele tenha aberto, e geralmente terminar a tarefa que ele está realizando no momento, antes de deixar de executar. Em alguns casos um processo pode ignorar o *SIGTERM* se ele estiver no meio de uma tarefa que não pode ser interrompida.

SIGKILL não pode ser ignorado por qualquer processo. Este é o sinal que diz ao processo, "Eu não me importo com o que você esteja fazendo, pare agora mesmo". Se você enviar um *SIGKILL* a qualquer processo, o FreeBSD irá interromper aquele processo[1].

Outros sinais que você pode querer conhecer são *SIGHUP*, *SIGUSR1*, e *SIGUSR2*. Estes são sinais de propósitos gerais, e aplicações distintas terão comportamento distintos ao receber estes sinais.

Suponha que você modificou o arquivo de configuração do seu servidor HTTP--você deseja dizer ao daemon para ele reler essa configuração. Você poderia parar e reiniciar o `httpd`, mas isso resultaria em uma breve interrupção do serviço, o que pode não ser desejável. A maioria dos daemons são escritos de forma que respondam a um *SIGHUP* com a releitura de seu arquivo de configuração. Então, ao invés de matar e reiniciar o processo `httpd` você enviaria o sinal *SIGHUP* ao processo. Por não existir uma forma padrão de resposta à estes sinais, cada daemon pode ter um comportamento diferente, de forma que você deve ler a documentação para o programa em questão.

Sinais são enviados utilizando o comando [kill\(1\)](#), como mostra o exemplo a seguir:

Enviando sinal aos processos

Esse exemplo mostra como enviar sinais ao [inetd\(8\)](#). O arquivo de configuração do `inetd` é o `/etc/inetd.conf`, e o `inetd` irá reler sua configuração, quando receber o sinal *SIGHUP*.

1. Encontre o ID do processo que você quer enviar o sinal. Faça isso com o [ps\(1\)](#) em conjunto com [grep\(1\)](#). O [grep\(1\)](#) é usado para procurar por uma cadeia de

caracteres na saída desejada. O comando é executado como usuário normal, e o [inetd\(8\)](#) é executado como `root`, então as opções `ax` devem ser usadas com o [ps\(1\)](#).

2. `% ps -ax | grep inetd`
3. `198 ?? IWS 0:00.00 inetd -wW`

Portando o PID do [inetd\(8\)](#) é 198. Em alguns casos o comando `grep inetd` poderá estar incluso também na saída desses comandos. Isso se deve à maneira como o [ps\(1\)](#) procura na lista de processos em execução.

4. Use [kill\(1\)](#) para enviar sinais. Pelo fato do [inetd\(8\)](#) estar sendo executado pelo `root` você deve usar o [su\(1\)](#) antes, para se tornar `root`.
5. `% su`
6. Password:
7. `# /bin/kill -s HUP 198`

Como de costume com a maioria dos comandos UNIX®, o [kill\(1\)](#) não apresentará qualquer mensagem se o resultado da ação for bem sucedido. Se você enviou um processo que não é seu, você verá uma mensagem parecida com ``kill: PID: Operation not permitted"`. Se você digitar o PID erroneamente, ou você vai enviar o sinal ao processo errado, o que pode ser bem ruim, ou, se você tiver sorte, terá enviado o sinal a um processo que não está ativo, e verá algo como ``kill: PID: No such process"`.

Por que usar o `/bin/kill`? Muitos interpretadores de comandos oferecem o comando `kill` embutido; ou seja, o interpretador enviará o sinal diretamente, ao invés de executar o `/bin/kill`. Isso pode ser muito útil, mas interpretadores de comandos distintos podem ter sintaxe distinta. Ao invés de aprender a usar cada uma, é bem mais fácil aprender usar o comando `/bin/kill ...` diretamente.

Enviar outros sinais é muito parecido, basta substituir o `TERM` ou `KILL` na linha de comando, conforme for necessário.

Importante: Matar processos aleatórios no sistema pode ser uma má idéia. Em especial o processo cujo ID é 1, o [init\(8\)](#), que é bem particular. Enviar um `/bin/kill -s KILL 1` é uma forma singela de desligar seu sistema. *Sempre* verifique duas vezes os argumentos do [kill\(1\)](#) antes de apertar **Enter**.

Notas

- [1] Não é completamente verdade--existem algumas poucas tarefas que não podem ser interrompidas. Por exemplo, se o processo está tentando ler um arquivo que está em uma outra estação da rede, e essa outra estação não está mais disponível por alguma razão (foi desligado, ou houve falha na rede), então o processo é definido como um processo que não pode ser interrompido. Eventualmente ele irá passar do tempo máximo de atividade permitida, tipicamente dois minutos (o *timeout*). Tão logo este período aconteça, o processo é morto.

3.9. Shells

No FreeBSD, uma grande parte do trabalho do dia a dia é feito por interface de linha de comando, chamada de interpretador de comandos, ou *shell*. A principal função de uma *shell* é receber comandos por um canal de entrada e executá-los. Uma série de *shells* ainda possuem funções embutidas, o que ajuda a facilitar a realização de tarefas do dia-a-dia, como gerenciamento de arquivos, edição de linha de comando, macros de comandos e variáveis de ambiente. O FreeBSD vem com uma série de interpretadores de comandos, como o `sh`, chamado de *Bourne Shell*, e `tcsh`, A ``C-shell" melhorada. Muitos outros interpretadores de comandos estão disponíveis na Coleção de Ports do FreeBSD, como o `zsh` e o `bash`.

Que *shell* utilizar? Trata-se meramente de uma questão de gosto. Se você é um programador C, se sentirá mais confortável com um interpretador de comandos similar à linguagem C, como o `tcsh`. Se você teve suas origens no Linux, ou é novo ao UNIX®, tenta experimentar o `bash`. O ponto é que cada interpretador de comandos possui propriedades únicas, que pode ou não funcionar corretamente em seu ambiente de trabalho, e também, outro ponto importante é sua liberdade de escolha para decidir qual e quando interpretador utilizar.

Uma das características mais comuns em interpretadores de comandos é completar nome de arquivos. Ao digitar as primeiras letras de um comando ou de um arquivo, você normalmente pode fazer com que a *shell* automaticamente complete o restante do comando ou arquivo, pressionando a tecla **Tab** no teclado. Aqui está um exemplo. Suponha que você tenha dois arquivos chamados `foobar` e `foo.bar`. Você quer apagar o `foo.bar`. Então o que você faria no teclado é: `rm fo[Tab].[Tab]`.

O interpretador de comandos apresentaria como resultado algo similar à `rm foo[BEEP].bar`.

O `[BEEP]` é o sino do console, que indica que o interpretador de comandos não foi capaz de completar totalmente o nome do arquivo porque existe mais do que uma opções que coincide. Ambos, `foobar` e `foo.bar` começam com `fo`, mas a *shell* pode completar o nome do arquivo até `foo`. Se neste momento você digitar `.`, e pressionar **Tab** mais uma vez, a *shell* deve ser capaz de completar o restante do nome do arquivo para você.

Outra funcionalidade de um interpretador de comandos é o uso de variáveis de ambiente. Variáveis de ambiente são duplas variáveis de valores armazenados no espaço de ambiente da *shell*. Esse espaço de ambiente pode ser lido por qualquer programa chamado pelo interpretador, de forma que contenha uma série de configurações interpretadas pelo mesmo. Segue aqui uma lista de variáveis de ambiente comuns, e seus respectivos significados:

Variável	Descrição
<code>USER</code>	Nome do usuário logado no momento.

Variável	Descrição
PATH	Lista de diretórios, divididos por vírgula, indicando onde deve-se procurar por binários.
DISPLAY	Nome de rede do terminal gráfico do X11 onde o mesmo deve se conectar, caso esteja disponível.
SHELL	O interpretador de comandos atual.
TERM	Nome do terminal do usuário. É usado para determinar as características do terminal.
TERMCAP	Entrada com a base de dados de códigos de terminal, que realizam várias funções do console.
OSTYPE	A espécie do sistema operacional em questão, por exemplo, FreeBSD.
MACHTYPE	A arquitetura de CPU onde o sistema está sendo executado.
EDITOR	Editor de texto padrão para o usuário.
PAGER	O paginador de texto padrão para o usuário.
MANPATH	Lista de diretórios separados por vírgulas que indicam onde as páginas de manuais devem ser buscadas.

Ajustar as variáveis de ambiente é tarefa que se difere de algumas formas entre as várias *shells* disponíveis. Por exemplo, nas *shells* baseadas na linguagem C, como o `tcsh` e o `csh`, você usaria o `setenv` para justar as variáveis de ambiente. Sobre um interpretador estilo *Bourne shell* como o próprio `sh` e o conhecido `bash`, você usaria o `export` para definir a variável de ambiente. Por exemplo, para definir ou modificar o valor da variável de ambiente `EDITOR`, sob `csh` ou `tcsh`, para definir a variável `EDITOR` com o valor `/usr/local/bin/emacs`, você faria algo como:

```
% setenv EDITOR /usr/local/bin/emacs
```

Enquanto sob uma *Bourne shell* seria:

```
% export EDITOR="/usr/local/bin/emacs"
```

Ainda é possível fazer a maioria dos interpretadores expandirem a variável, simplesmente adicionado o carácter `$` na frente da própria variável. Por exemplo, `echo $TERM` imprimiria todo o conteúdo da variável `$TERM`, por quê o interpretador interpreta a variável `$TERM` como seu valor, e o passa para o `echo`.

Os interpretadores de comandos cuidam ainda de uma série de caracteres especiais, chamados de meta-caracteres, como representações especiais de dados. O caracter mais

comum é o `*`, que representa qualquer conjunto de caracteres em um nome de arquivo. Os meta-caractéres especiais podem ser usados como substituição de arquivos. Por exemplo, o comando `echo *` tem resultado similar à saída de um `ls` pois o interpretador de comandos pega todos os arquivos que batem com `*` e os coloca na linha de comando de forma que o `echo` possa encher-ga-los.

Para evitar que a *shell* interprete estes caracteres especiais, pode-se utiliza-los como exclusão, inserindo uma barra invertida (`\`) na frente deles. `echo $TERM` imprime em que terminal você está, enquanto `echo \$TERM` imprime `$TERM` exatamente.

3.9.1. Modificando sua *Shell*

A forma mais fácil de modificar seu interpretador de comandos é usar o comando `chsh`. Executar o `chsh` te colocará no editor definido na variável de ambiente `EDITOR`; se esta variável não estiver definida, você será colocado no `vi`. Modifique a linha ```Shell:"` de acordo com suas necessidades.

Você pode ainda usar o `chsh` com a opção `-s`; assim você pode ajustar sua *shell* sem entrar no editor. Por exemplo, se a intenção era modificar seu interpretador para usar o `bash`, o truque pode ser feito da seguinte forma:

```
% chsh -s /usr/local/bin/bash
```

Utilizar o `chsh` sem parâmetros e editar a *shell* também funcionaria.

Nota: O interpretador de comandos que você deseja utilizar *deve* estar listado no arquivo `/etc/shells`. Se você instalou o interpretador à partir da [Coleção de Ports](#), isso já foi feito. Se você instalou na mão, você mesmo deve fazê-lo.

Por exemplo, se você instalou o `bash` manualmente e o colocou no `/usr/local/bin`, você vai querer fazer o seguinte:

```
# echo "/usr/local/bin/bash" >> /etc/shells
```

3.10. Editores de Texto

Uma série de configurações no FreeBSD são feitas à partir da edição de arquivos de texto. Por tal motivo, seria uma boa idéia se tornar familiar com algum editor de texto. FreeBSD vem com alguns editores, como parte de seu sistema base, e muitos outros editores estão disponíveis na coleção de *ports*.

O editor mais simples e mais fácil para aprender, é um editor chamado **ee**, que significa *easy editor* ("editor fácil"). Para iniciar o **ee**, basta digitar na linha de comando `ee nomedoarquivo` onde *nomedoarquivo* é o nome do arquivo que deve ser editado. Por exemplo, para editar o `/etc/rc.conf`, digite `ee /etc/rc.conf`. Uma vez dentro do **ee**, todos os comandos para manipular as funções do editor estão listadas no topo da tela. O caracter `^` representa a tecla **Ctrl** do teclado, de forma que `^e` implica na combinação de teclas **Ctrl+e**. Para sair do **ee**, aperte a tecla **Esc**, depois escolha *leave editor* (sair do editor). O editor irá perguntar se você deseja salvar as modificações no arquivo, caso ele tenha sido alterado.

FreeBSD também oferece editores de texto mais poderosos, como o `vi`, como parte da base do sistema, enquanto outros editores como `emacs` e `vim`, estão disponíveis na Coleção de *Ports* do FreeBSD. Estes editores oferecem muitas funcionalidades e são extramente poderosos, ao custo de serem um pouco mais difíceis de aprender. Contudo se você planeja editar muitos textos, aprender algum editor de texto mais poderoso como `vim` ou `emacs` economizará a você mais tempo, à longo prazo.

3.11. Dispositivos e Nós de Dispositivos

Um dispositivo é um termo usado em sua maioria para fazer menção à atividades relacionadas à equipamentos físicos no sistema, incluindo discos, impressoras, placas de vídeo e teclados. Quando o FreeBSD é iniciado, a maioria dos dados que o sistema apresenta são referentes à detecção de dispositivos. É possível reavaliar as mensagens de inicialização, analisando o arquivo `/var/run/dmesg.boot`.

Por exemplo, o dispositivo `acd0` é o primeiro CDROM IDE disponível, enquanto `kbd0` representa o teclado.

A maioria destes dispositivos em um sistema operacional UNIX® devem ser acessados por intermédio de arquivos especiais, chamados nós de dispositivos, ou *device nodes*, que estão dispostos sob o diretório `/dev`.

3.11.1. Criando Nós de Dispositivos

Ao adicionar um novo dispositivo em seu sistema, ou compilar suporte à dispositivos adicionais, pode ser necessário criar um ou mais nós de dispositivos para controlar os recém adicionados.

3.11.1.1. MAKEDEV Script

Em sistemas sem *DEVFS* (isso diz respeito à todas versões do FreeBSD antes da 5.0), nós de dispositivos são criados utilizando a rotina [MAKEDEV\(8\)](#) como apresentado à seguir:

```
# cd /dev
# sh MAKEDEV ad1
```

Este exemplo cria os nós de dispositivos apropriados para o segundo dispositivo IDE, quando o mesmo for instalado.

3.11.1.2. DEVFS (DEVICE File System)

O sistema de arquivo de dispositivos, ou simplesmente *DEVFS*, oferece acesso ao espaço em kernel para os dados referentes aos nomes de dispositivos, no sistema de arquivos global. Ao invés de ter que criar e modificar os nós de dispositivos, *DEVFS* mantém este sistema de arquivos particular.

Veja a página de manual do [devfs\(5\)](#) para obter mais informações.

DEVFS é utilizado por padrão no FreeBSD 5.0 e posteriores.

3.12. Formatos de binários

Para entender o motivo pelo qual o FreeBSD usa o formato ELF você deve saber primeiro um pouco mais sobre os três formatos atualmente "dominantes" de arquivos executáveis UNIX®:

- [a.out\(5\)](#)

O mais antigo e "clássico" formato de objetos UNIX. Utiliza um cabeçalho curto e compacto com um número mágico (*magic number*) no início do mesmo, que geralmente é usado para caracterizar seu formato (veja [a.out\(5\)](#) para mais detalhes). Contém três segmentos carregados: `.text`, `.data` e `.bss`, acrescidos de uma tabela de símbolos e uma tabela de conjunto caracteres.

- COFF

O formato de objetivos SVR3. Seu cabeçalho compreende uma tabela de seções, de forma que é possível ter mais seções do que apenas `.text`, `.data` e `.bss`.

- ELF

O sucessor direto do COFF, oferecendo múltiplas seções e valores possíveis de 32-bit ou 64-bit. Uma grande desvantagem: ELF foi projetado assumindo-se que

haveria possivelmente apenas uma ABI por arquitetura de sistema. Essa consideração contudo é levemente incorreta hoje em dia, e nem mesmo no mundo comercial do SYSV (onde existem ao menos três ABIs: SVR4, Solaris e SCO) ela é real.

FreeBSD tenta amenizar este problema por meio de um utilitário que reconhece um executável ELF e obtém informações sobre a ABI compilada neste. O utilitário em questão é o `brandelf`. Veja a página de manual do [brandelf\(1\)](#) para obter mais informações.

FreeBSD vem do campo "clássico" e durante muito tempo usou o formato [a.out\(5\)](#), uma tecnologia utilizada e aprovada por muitas gerações de versões de sistemas BSD, até o ramo 3.X do FreeBSD. Apesar de ser possível gerar binários (e mesmo o kernel) no formato ELF bem antes do FreeBSD 3.X, o sistema inicialmente resistiu à "pressão" de mudar para o formato ELF como formato padrão. Por quê? Bem, quando o pessoal do campo Linux fez sua dolorosa transição para ELF, os executáveis `a.out` não podiam ser utilizados pois sua tabela de salto era consideravelmente inflexível, e a construção de bibliotecas compartilhadas parecia ser complexa demais no novo formato, para os fabricantes e desenvolvedores em geral. Depois que algumas ferramentas ELF ofereceram soluções ao problema de bibliotecas compartilhadas, o formato passou a ser visto como "o caminho a ser seguido", os custos de migração para este formato já eram então aceitáveis, e a transição foi feita. O mecanismo de bibliotecas compartilhadas do FreeBSD é baseado com maior proximidade ao estilo de bibliotecas compartilhadas do SunOS™ da Sun, e como tal, é muito fácil de ser utilizado.

Então, por que existem tantos formatos distintos?

De volta ao sombrio, gélido e distante passado, existiam apenas equipamentos simples de computação. Estes equipamentos simples, suportavam sistemas simples e pequenos. O formato `a.out` era completamente adequado às tarefas apresentadas pelos binários nestes sistemas simplistas (como PDP-11). Conforme o UNIX passou a ser convertido com base nestes sistemas simples, mantiveram o formato `a.out` pois era suficiente para as primeiras conversões do UNIX, para arquiteturas como Motorola 68k, VAXen, etc.

Depois algum engenheiro de *hardware* iluminado decidiu que ele poderia forçar os programas a realizarem algumas tarefas mais simples, de forma que ele poderia retirar alguns itens de seu projeto de equipamento e permitir que o núcleo de sua CPU processasse de forma mais rápida. Mesmo tendo sido projetado para funcionar com este novo tipo de hardware (conhecido hoje em dia como RISC), o formato `a.out` não se encaixava funcionalmente para o novo equipamento, então vários formatos foram desenvolvidos para tentar garantir melhor performance deste novo equipamento, quando comparado ao que o limitado formato `a.out` poderia oferecer. Algumas coisas como COFF, ECOFF, e alguns outros formatos obscuros foram inventados e suas limitações exploradas antes que as coisas pudessem se estabilizar no formato ELF.

Em adição, o tamanho dos programas estavam se tornando grandes, e os discos (bem como memória física) continuavam relativamente pequenos, então o conceito de bibliotecas compartilhadas nasceu. O sistema de memória virtual também se tornou mais sofisticado. Enquanto cada um desses avanços foram feitos utilizando o formato `a.out`, sua usabilidade foi se tornando cada vez mais duvidosa, na proporção que novas

características eram adicionadas. Em adição, as pessoas queriam poder carregar as coisas de forma dinâmica, no momento de execução do sistema, ou então dispensar trechos de seus programas depois que o código inicial tivesse sido executado, para economizar memória e espaço de troca (*swap*). As linguagens se tornaram mais sofisticadas, e as pessoas queriam que códigos fossem carregados antes do programa principal de forma automática. Uma série de modificações foram feitas no formato `a.out` para permitir que esse tipo de coisa pudesse acontecer, e basicamente tudo funcionou bem por um tempo. Em determinado momento, o `a.out` não podia mais contornar estes problemas sem sobrecarregar seu código e aumentar sua complexidade. O formato ELF resolvia a maioria desses problemas, mas seria doloroso migrar de um sistema que, basicamente, funcionava. Então o formato ELF teve que esperar até que fosse mais penoso continuar com o `a.out`, do que migrar para o ELF.

Contudo, com o passar do tempo, as ferramentas de construção de onde o FreeBSD derivou suas próprias ferramentas de construção (o montador *assembly* e o carregador, especialmente) se envolveram em duas árvores paralelas. A árvore FreeBSD adicionou bibliotecas compartilhadas e corrigiu algumas falhas. O pessoal GNU que originalmente escreviam esses programas, rescreveram os mesmos de adicionaram suporte simples à compilação cruzada, podendo adicionar formatos diferentes à bel prazer, e assim por diante. Partindo do princípio que muitas pessoas desejavam criar compiladores cruzados para o FreeBSD, tiveram problemas pois os fontes antigos do FreeBSD faziam chamadas ao `as` e `ld` o que não possibilitava tal tarefa. A nova corrente de ferramentas GNU (**binutils**) suporta compilação cruzada, ELF, bibliotecas compartilhadas, extensões C++, etc. Além disso, muitos fabricantes passaram a lançar binários ELF, e era bom que o FreeBSD pudesse rodá-los.

ELF é mais expressivo que `a.out` e oferece mais extensibilidade no sistema base. As ferramentas ELF são melhor mantidas, e oferecem suporte à compilação cruzada, que é importante para muita gente. O formato ELF pode ser um pouco mais lento do que `a.out`, mas tentar medir esta diferença é bem difícil. Existem ainda vários detalhes distintos entre ambos, em relação à como eles mapeiam páginas, tratam código de inicialização, carregamento das instruções, etc. Nenhum destes fatores são muito relevantes, mas são diferentes. Futuramente, o suporte à `a.out` será retirado do kernel `GENERIC`, e eventualmente removido do sistema quando a necessidade de executar programas `a.out` for passado.

3.13. Para mais informações

3.13.1. Páginas de Manuais

A documentação mais abrangente do FreeBSD é em forma de páginas de manuais. Quase todo programa no sistema é acompanhado com um breve manual de referência que explica a operação básica e os vários argumentos do programa. Estes manuais podem ser visualizados com o comando `man`. A utilização do comando `man` é simples:

```
% man comando
```

comando é o nome do comando sobre o qual você deseja aprender. Por exemplo, para aprender sobre o comando `ls`, digite:

```
% man ls
```

O manual ``online" é dividido em seções numeradas:

1. Comandos de usuário.
2. Chamadas de sistema e número de erros.
3. Funções de bibliotecas C.
4. Controladores de dispositivos.
5. Formato de arquivos.
6. Jogos e outras diversões.
7. Informações gerais.
8. Comandos de manutenção e operação do sistema.
9. Desenvolvedores do kernel.

Em alguns casos, o mesmo tópico pode aparecer em mais de uma seção do manual online. Por exemplo, se houver um comando `chmod` para o usuário, e se houver uma chamada de sistema denominada `chmod()`, você pode informar ao comando `man` qual delas você quer, especificando a seção:

```
% man 1 chmod
```

Esse comando apresentará a página de manual para o comando do usuário, `chmod`. Referências para uma seção particular do manual online normalmente são apresentadas entre parênteses, nos documentos escritos, de forma que [chmod\(1\)](#) se refere ao comando de usuário para `chmod`, e [chmod\(2\)](#) se refere à chamada de sistema.

É bem fácil utilizar essa ferramenta, quando você sabe o nome do comando sobre o qual quer obter mais informações, mas e se você não puder lembrar o nome do comando? Pode usar o `man` para procurar por palavras chaves nas descrições do comando, usando opção `-k`:

```
% man -k mail
```

Com esse comando, uma lista de programas será apresentada, contendo em sua descrição a expressão ``mail". Na verdade esse comando tem a mesma funcionalidade de se utilizar o comando `apropos`.

Então, você está dando uma olhada em todos esses simpáticos programas que você encontrou sob `/usr/bin` mas não tem a menor idéia do que quer que cada um deles faça? Simplesmente digite:

```
% cd /usr/bin
% man -f *
```

ou

```
% cd /usr/bin
% whatis *
```

que fazem a mesma coisa.

3.13.2. Arquivos de Informações GNU

O FreeBSD inclui várias aplicações e utilitários mantidos pela Fundação do Software Livre (FSF). Em complemento às páginas de manual, estes programas vem ainda com um documento hipertexto extenso, chamado arquivo *info* que pode ser visualizado com o comando `info` ou, se você dispõe do **emacs** instalado, o modo ```info"` do **emacs** tem essa função.

Para usar o comando [info\(1\)](#), simplesmente digite:

```
% info
```

Para uma breve introdução, digite *h*. Para uma rápida referência aos comandos, digite *?*.

Capítulo 4. Instalando aplicativos: Pacotes e Ports

Índice

- 4.1. [Sinótese](#)
- 4.2. [Visão geral do processo de instalação de *software*](#)
- 4.3. [Encontrando seu aplicativo](#)
- 4.4. [Usando o sistema de pacotes](#)
- 4.5. [Usando a coleção de *ports*](#)
- 4.6. [Atividades pós-instalação](#)
- 4.7. [Lidando com ports quebrados](#)

4.1. Sinótese

O FreeBSD tem como parte de seu pacote básico uma vasta coleção de ferramentas de sistemas. é possível fazer muita coisa antes de ter que instalar aplicativos adicionais de terceiros para trabalhar de verdade. O FreeBSD disponibiliza duas tecnologias complementares para a instalação de aplicativos em seu sistema: A coleção de *ports* do FreeBSD, e pacotes binários de *software*. Qualquer um dos dois sistemas pode ser utilizado para instalar as versões mais recentes das suas aplicações preferidas a partir de uma mídia local ou então direto da rede.

Após ler este capítulo você irá conhecer:

- Como instalar aplicativos a partir de pacotes binários do *software*.
- Como compilar aplicativos a partir da coleção do *Ports*.
- Como remover aplicativos instalados previamente através de *packages* ou *ports*.
- Como sobrescrever os valores padrões utilizados pela coleção de *ports*.
- Como atualizar sua coleção de *ports*.

4.2. Visão geral do processo de instalação de *software*

Se você já utilizou um sistema UNIX® anteriormente, você deve saber que o procedimento típico para instalar um aplicativo é parecido com isto:

1. Faça o download do software, o qual pode ser distribuído como código fonte ou no formato binário.
2. Descompacte o arquivo de distribuição do software (tipicamente um *tarball* comprimido com [compress\(1\)](#), [gzip\(1\)](#), ou [bzip2\(1\)](#)).
3. Localize a documentação (normalmente é um arquivo chamado `INSTALL` ou `README`, ou então um conjunto de arquivos no subdiretório `doc/`) e descubra como instalar o *software*.
4. Se o software for distribuído como código fonte, compile-o. Isto pode envolver a edição de um arquivo `Makefile` , ou a execução de um script chamado `configure` , além de outras tarefas.
5. Teste e instale o software.

E é só isso se tudo correu como esperado. Agora se você está instalando um aplicativo que não foi deliberadamente portado para o FreeBSD, você pode ter que fazer ajustes no código fonte do mesmo para que ele compile corretamente.

Se você desejar, você pode continuar instalando aplicativos no FreeBSD pelo método "tradicional". Contudo, o FreeBSD disponibiliza duas tecnologias que podem lhe economizar muito esforço: *packages* e *ports*. No momento, mais de 11,800 aplicativos já foram disponibilizados desta forma.

Independente de qual seja o aplicativo, o pacote FreeBSD do mesmo consiste-se em um único arquivo para download. O pacote contém cópias pré-compiladas de todos os comandos do aplicativo, bem como cópias dos arquivos de configuração e/ou da documentação. O gerenciamento de um pacote FreeBSD é realizado por uma série de comandos específicos, tais como [pkg_add\(1\)](#), [pkg_delete\(1\)](#), [pkg_info\(1\)](#) entre outros. A instalação de um novo aplicativo pode ser realizada através da execução de um único comando.

Um *port* FreeBSD para um aplicativo consiste-se em uma coleção de arquivos destinados a automatizar o processo de compilação da aplicação a partir do seu código fonte.

Lembre-se que existem diversos passos que você teria que executar para compilar o programa manualmente (baixar , descompactar, aplicar correções, compilar e instalar). Os arquivos de um port contém todas as informações necessárias para possibilitar que o sistema automatize o processo para você. Você executa um conjunto de comandos simples e o código fonte do aplicativo é automaticamente baixado, descompactado, corrigido, compilado e instalado para você.

De fato, o sistema de *ports* pode ser utilizado para pacotes binários do aplicativo, o qual pode ser manipulado mais tarde através do comando `pkg_add`, ou por qualquer outro comando de gerenciamento de pacotes, os quais iremos introduzir adiante.

Ambos os sistemas de instalação cuidam da instalação das *dependências* do aplicativo. Suponha que você deseja instalar uma aplicação que depende de uma biblioteca específica para funcionar. Tanto o aplicativo quanto a biblioteca são disponibilizadas no FreeBSD como pacote e como *ports*. Se você utilizar o comando `pkg_add` ou o sistema de *ports* para adicionar o aplicativo, ambos irão informar que a biblioteca necessária para o funcionamento do aplicativo não está instalada corretamente e irão instalar automaticamente a biblioteca antes de instalar o aplicativo.

dado que ambas as tecnologias são muito similares, você deve querer saber porque o FreeBSD se preocupa com as duas. Os sistemas de pacotes e *ports* possuem suas próprias virtudes, a que você irá utilizar dependerá da sua preferência pessoal.

Benefícios do sistema de Pacotes

- Um pacote *tarball* comprimido dos binários do aplicativo é normalmente menor que um *tarball* compactado contendo o código fonte do mesmo aplicativo.
- Um pacote não requer nenhuma compilação adicional. Para os grandes aplicativos como por exemplo **Mozilla**, **KDE**, ou **GNOME** isto pode ser importante, particularmente se você está utilizando uma máquina lenta.
- A instalação de um pacote não requer nenhum conhecimento do processo envolvido na compilação do software para o FreeBSD.

Benefícios do Sistema de *ports*

- Um pacote é compilado normalmente com opções conservadoras, pois ele tem que executar no maior numero de sistemas possível. Ao instalar a partir do *ports*, você pode ajustar as opções de compilação para (por exemplo) gerar um binário específico para processadores Pentium IV ou para processadores Athlon.
- Alguns aplicativos possuem opções que são especificadas no ato da compilação que estão relacionadas ao que eles podem ou não fazer. Por exemplo, o **Apache** pode ser configurado no momento de sua compilação com uma grande variedade de opções. Ao compilar a partir do *port* voce não é obrigado a aceitar as opções *default*, e pode defini-las de acordo com suas necessidades.

Em alguns casos, irão existir múltiplos pacotes de um mesmo aplicativo para especificar certas opções. Por exemplo, o aplicativo **Ghostscript** esta disponível em dois pacotes, `ghostscript` e `ghostscript-nox11`, você irá optar por um ou por outro dependendo se possui ou não um servidor X11 instalado. Esse tipo de ajuste fino é possível de ser realizado com pacotes, mas se torna inviável se o aplicativo possuir mais de uma ou duas opções, em tempo de compilação, diferentes.

- As condições de licenciamento de alguns softwares proíbem a distribuição de pacotes binários. Eles tem que ser distribuídos como código fonte.

- Algumas pessoas não confiam em distribuições binárias. Tendo acesso ao código fonte, você mesmo pode (em teoria) inspecioná-lo e procurar por problemas potenciais.
- Se você possui correções customizadas, você vai precisar do código fonte para poder aplicá-las.
- Muitos usuarios gostam de ter o código a sua disposição, para estudá-los, modificá-los, copiá-los (respeitando a licença, é claro) e assim por diante.

Para manter-se atualizado sobre a coleção de *ports*, inscreva-se nas listas [lista de discussão sobre FreeBSD ports](#) e [o sistema de ports FreeBSD](#) e [lista de discussão dos bugs do ports em FreeBSD](#).

O restante deste capítulo irá explicar como utilizar o sistema de *packages* e o de *ports* para instalar e gerenciar aplicativos de terceiros no FreeBSD.

4.3. Encontrando seu aplicativo

Antes que você possa instalar todas as aplicações de que necessita, você precisa saber do que precisa, e também como a aplicação se chama.

A lista de aplicativos disponíveis para FreeBSD cresce o tempo todo. Felizmente, existem diversas formas de procurar o que você necessita:

- O web site do projeto FreeBSD mantém uma lista atualizada de todos os aplicativos disponíveis e na qual você pode executar buscas, em <http://www.FreeBSD.org/ports/>. O *ports* está dividido em categorias, e você pode efetuar buscas pelo nome do aplicativo (se você souber qual é), ou então olhar todos os aplicativos disponíveis em uma categoria.
- Dan Langille mantém o *sítio web FreshPorts*, em <http://www.FreshPorts.org/>. O *FreshPorts* rastreia as alterações nas aplicações na árvore do *ports* a medida que elas acontecem, permitindo que você monitore um ou mais *ports*, enviando um email para você quando eles forem atualizados.
- Se você não sabe o nome da aplicação que deseja, consulte um *sítio* no estilo do *FreshMeat* (<http://www.freshmeat.net/>) para procurar um aplicativo que lhe atenda, quando encontrar volte ao *sítio* do FreeBSD e verifique se o aplicativo já foi incorporado ao *ports*.

4.4. Usando o sistema de pacotes

Contributed by Chern Lee.

4.4.1. Instalando um pacote

Você pode utilizar o utilitário [pkg_add\(1\)](#) para instalar um pacote de *software* no FreeBSD a partir de um arquivo local ou a partir de um servidor na rede.

Exemplo 4-1. Baixando manualmente um pacote e instalando-o localmente

```
# ftp -a ftp2.FreeBSD.org
Connected to ftp2.FreeBSD.org.
220 ftp2.FreeBSD.org FTP server (Version 6.00LS) ready.
331 Guest login ok, send your email address as password.
230-
230-      This machine is in Vienna, VA, USA, hosted by Verio.
230-      Questions? E-mail freebsd@vienna.verio.net.
230-
230-
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /pub/FreeBSD/ports/packages/sysutils/
250 CWD command successful.
ftp> get lsof-4.56.4.tgz
local: lsof-4.56.4.tgz remote: lsof-4.56.4.tgz
200 PORT command successful.
150 Opening BINARY mode data connection for 'lsof-4.56.4.tgz' (92375
bytes).
100% |*****| 92375
00:00 ETA
226 Transfer complete.
92375 bytes received in 5.60 seconds (16.11 KB/s)
ftp> exit
# pkg_add lsof-4.56.4.tgz
```

Se você não possui uma fonte local de pacotes (como por exemplo um conjunto de CD-ROMs do FreeBSD) será mais fácil utilizar a opção *-r* com o [pkg_add\(1\)](#). Esta opção fará com que o utilitário determine automaticamente o formato correto do objeto, a versão apropriada, bem como dirá ao utilitário para fazer o download e a instalação do aplicativo a partir de um servidor FTP.

```
# pkg_add -r lsof
```

O exemplo acima irá baixar o pacote correto e instalá-lo sem nenhuma outra intervenção do usuário. O [pkg_add\(1\)](#) utiliza o [fetch\(3\)](#) para baixar os arquivos, o qual considera variáveis de ambiente incluindo `FTP_PASSIVE_MODE`, `FTP_PROXY`, e `FTP_PASSWORD`. Você pode ter que ajustar uma ou mais delas caso esteja atrás de um *firewall*, ou então terá que utilizar um *proxy* FTP/HTTP. Consulte [fetch\(3\)](#) para visualizar a lista completa. Observe que no exemplo acima *lsof* é utilizado no lugar de *lsof-4.56.4*. Quando o recurso de busca remota é utilizado, o número de versão do pacote deve ser removido. O [pkg_add\(1\)](#) irá buscar automaticamente a última versão do aplicativo.

Os pacotes são distribuídos no formato `.tgz` e `.tbz`, você pode encontrá-los em <ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/packages/>, ou nos CDs de instalação do FreeBSD. Todos os CDs do conjunto de CDs de instalação do FreeBSD (e do *PowerPack*, etc) contém pacotes no diretório `/packages`. O layout do diretório `packages` é similar ao encontrado na árvore `/usr/ports`. Cada categoria possui seu próprio diretório, e cada pacote pode ser encontrado no diretório `All`.

A estrutura do diretório do sistema de pacotes é idêntica ao layout do *ports*; eles trabalham em conjunto para formar o sistema de pacotes/*ports*.

4.4.2. Gerenciando Pacotes

O [`pkg_info\(1\)`](#) é um utilitário que lista e descreve os vários pacotes instalados.

```
# pkg_info
cvsup-16.1          A general network file distribution system
                    optimized for CV
docbook-1.2        Meta-port for the different versions of the
DocBook DTD
...
```

O [`pkg_version\(1\)`](#) é um utilitário que sumariza a versão de todos os pacotes instalados. Ele compara a versão do pacote instalado com a versão atual encontrada na árvore do *ports*.

```
# pkg_version
cvsup              =
docbook            =
...
```

Os símbolos na segunda coluna indicam a idade relativa da versão instalada e da versão disponível na árvore local do *ports*.

Symbol	Meaning
=	A versão instalada do pacote é idêntica a versão disponível na árvore local do <i>ports</i> .
<	A versão instalada do pacote é mais antiga que a versão disponível na árvore local do <i>ports</i> .
>	A versão instalada do pacote é mais recente que a versão disponível na árvore local do <i>ports</i> . (Provavelmente sua árvore do <i>ports</i> está desatualizada)
?	O pacote instalado não foi localizado no índice do <i>ports</i> . (Isto pode ocorrer, por exemplo, se um aplicativo instalado for removido da árvore do <i>ports</i> ou então foi renomeado).
*	Existem múltiplas versões do pacote.

4.4.3. Deletando um pacote

Para remover um pacote previamente instalado, utilize o utilitário [pkg_delete\(1\)](#).

```
# pkg_delete xchat-1.7.1
```

4.4.4. Miscelânea

Todas as informações dos pacotes são armazenadas no diretório `/var/db/pkg`. A lista de arquivos instalados e a descrição de cada pacote pode ser encontrada nos arquivos destes diretórios.

4.5. Usando a coleção de *ports*

A sessão abaixo fornecerá as instruções básicas sobre o uso da coleção de *ports* para instalar e remover programas do seu sistema.

4.5.1. Obtendo a coleção do *ports*

Antes que você possa instalar um port, você primeiro precisa obter a coleção de *ports*, a qual é basicamente um conjunto de arquivos `Makefiles`, `patches`, e arquivos de descrição colocados sob `/usr/ports`

Quando você está instalando o FreeBSD, o **Sysinstall** pergunta se você deseja instalar a árvore do *ports*, se você respondeu não naquela ocasião, você pode seguir as seguintes instruções para instalar a coleção de *ports*:

Via *Sysinstall*

Este método envolve novamente a utilização do **sysinstall** para instalar manualmente a coleção de *ports*.

1. Estando ``logado" como o usuário `root`, execute o comando `/stand/sysinstall` como mostrado abaixo:
2. `# /stand/sysinstall`
3. Utilize as setas direcionais de seu teclado e selecione a opção `Configure`, e pressione **Enter** .
4. Selecione a opção `Distributions`, e pressione **Enter**.
5. Selecione a opção `ports`, e pressione a **barra de espaço** para marcá-la.
6. Selecione a opção `Exit`, e pressione **Enter**.
7. Agora você deve escolher a mídia a partir da qual a coleção de *ports* será instalada, como por exemplo `CDROM`, `FTP`, etc. Selecione a opção desejada e pressione **Enter** para iniciar a instalação.
8. Selecione a opção `Exit` e pressione **Enter** para voltar a tela inicial.
9. Pressione **X** para sair do **sysinstall**.

Um método alternativo para obter e manter atualizada a coleção de *ports* é através da utilização do aplicativo **CVSup**. Dê uma olhada no arquivo de configuração do **CVSup** para o *ports* em `/usr/share/examples/cvsup/ports-supfile`. Consulte a sessão [Utilizando o CVSup \(Seção A.5\)](#) para maiores informações sobre a utilização do **CVSup** e deste arquivo.

Via CVSup

Este é um guia rápido sobre como obter a coleção de *ports* através do uso do **CVSup**. Se voce deseja manter a sua árvore do *ports* atualizada, ou então se desejar aprender mais sobre o **CVSup**, leia a sessão mencionada anteriormente.

1. Instale o [portnet/cvsup](#). Consulte a sessão [Instalando o CVSup \(Seção A.5.2\)](#) para maiores detalhes.
2. Estando ``logado" como o usuário `root`, copie o arquivo `/usr/share/examples/cvsup/ports-supfile` para um novo local, como por exemplo para o diretório `/root` ou então para o seu diretório `home`.
3. Abra o arquivo `ports-supfile` com o seu editor de textos preferido.
4. Altere a entrada `CHANGE_THIS.FreeBSD.org` para o endereço válido de um servidor **CVSup** perto de você. Consulte a sessão [Espelhos CVSup \(Seção A.5.7\)](#) para visualizar a lista completa com todos os servidores disponíveis.
5. Execute o comando `cvsup`:
6.

```
# cvsup -g -L 2 /root/ports-supfile
```
7. Ao executar este comando novamente no futuro, ele irá atualizar a sua árvore do *ports*, baixando todas as alterações que ocorreram na coleção desde a ultima vez em que ele foi executado.

4.5.2. Instalando um aplicativo via Ports

A primeira coisa que precisa ser explicada a alguém que está iniciando no uso do *ports* é que na verdade um *port* é composto por um ``conjunto de arquivos". Em poucas palavras, um *port* é o conjunto mínimo de arquivos que diz ao sistema FreeBSD como compilar corretamente e como instalar um aplicativo. Cada *port* é composto por:

- Um arquivo `Makefile`. O `Makefile` contém várias instruções que definem como o aplicativo deve ser compilado e onde ele deve ser instalado em seu sistema.
- Um arquivo `distinfo`. Este arquivo contém informações sobre quais arquivos precisam ser obtidos para compilar o *port* e os seus respectivos *checksums*, os quais são utilizados para verificar se o o arquivo não foi corrompido durante o processo de transmissão, a verificação é realizada através do utilitário [md5\(1\)](#).
- Um diretório chamado `files`. Este diretório contém as correções necessárias para que o aplicativo possa ser compilado e instalado no FreeBSD. As correções são basicamente arquivos pequenos os quais especificam alterações a serem aplicadas a determinados arquivos. São arquivos em formato texto plano, e basicamente dizem ``remova a linha 10" ou ``Altere a linha 26 para ...". As correções são normalmente conhecidas como ``diffs", pois são gerados pelo utilitário [diff\(1\)](#)

Este diretório também pode conter outros arquivos necessários para a compilação do aplicativo.

- Um arquivo `pkg-descr`. Este arquivo contém a descrição detalhada do aplicativo.
- Um arquivo `pkg-plist`. Este arquivo contém a lista de todos os arquivos que serão instalados pelo *port*. Ele também diz ao sistema de *ports* quais arquivos devem ser removidos durante a desinstalação do aplicativo.

Alguns *ports* possuem outros arquivos, como por exemplo `pkg-message`. O sistema de *ports* utiliza estes arquivos para lidar com situações especiais. Se você deseja maiores detalhes sobre estes arquivos, e sobre o sistema de *ports* em geral, consulte o [FreeBSD Porter's Handbook](#).

Agora que você tem bastante informação básica para saber para que a coleção de *ports* é utilizada, você está pronto para instalar seu primeiro aplicativo pelo sistema de *ports*. Existem duas formas como isto pode ser feito, e cada uma delas está detalhada abaixo.

No entanto antes de iniciarmos você precisará escolher um aplicativo para instalar. Existem algumas formas de fazer isso, mas a maneira mais fácil é através da [listagem de ports no web site do FreeBSD](#). No site você pode navegar pela árvore de *ports* ou então utilizar-se do mecanismo de busca. Cada *port* possui uma descrição da finalidade do aplicativo a que ele se refere, a leitura destas descrições vai ajudá-lo na escolha de qual *port* instalar.

Uma outra opção é utilizar-se do utilitário [whereis\(1\)](#). Simplesmente execute `whereis arquivo`, onde *arquivo* é o programa que você deseja instalar. Se ele for encontrado em seu sistema, você será informado da sua localização, como mostrado abaixo:

```
# whereis lsof
lsof: /usr/ports/sysutils/lsof
```

A saída acima nos diz que o `lsof` (utilitário de sistema) pode ser encontrado no diretório `/usr/ports/sysutils/lsof`.

Outra forma de procurar um *port* em particular é através da utilização do mecanismo interno de busca disponibilizado pelo sistema de *ports*. Para usar o recurso de busca, você tem que estar dentro do diretório `/usr/ports`. Uma vez dentro do diretório, execute o comando `make search name=nome_do_programa` onde *nome_do_programa* é o nome do aplicativo que vc está procurando. Por exemplo, se você estiver procurando pelo `lsof`:

```
# cd /usr/ports
# make search name=lsof
Port:    lsof-4.56.4
Path:    /usr/ports/sysutils/lsof
Info:    Lists information about open files (similar to fstat(1))
Maint:   obrien@FreeBSD.org
Index:   sysutils
B-deps:
R-deps:
```

A parte da saída na qual você deve prestar um atenção especial é a linha iniciada com `Path:`, uma vez que é ela que diz onde encontrar o *port*. As outras informações

exibidas não são necessárias para a instalação do aplicativo pelo *ports*, e portanto não serão comentadas aqui.

Para uma busca mais profunda voce pode utilizar `make search key=string`, onde *string* é a palavra chave que deseja procurar. Esta busca irá varrer os nomes dos *ports*, os comentários, as descrições, as dependências e pode ser utilizada para buscar um aplicativo que esteja relacionado a um assunto em particular caso você não saiba o nome do aplicativo pelo qual você está procurando.

Em ambos os casos descritos acima, o argumento da busca é *case-insensitive*. A busca por ```LSOF``` irá apresentar o mesmo resultado que uma busca por ```lsOf```.

Nota: Voce precisa estar ```logado``` como `root` para instalar um aplicativo pelo *ports*.

Agora que você já encontrou um aplicativo que gostaria de instalar, você já está pronto para executar a instalação. O *port* inclui instruções sobre como compilar o código fonte, porém não inclui o código fonte propriamente dito. Você pode obter o código fonte a partir de um CDROM ou a partir da internet. O código fonte é distribuído da forma como o autor do software deseja, normalmente o formato utilizado é um *tarball* compactado com o utilitário `gzip`, porém ele pode estar compactado com outro utilitário ou mesmo estar descompactado. O código fonte de um software, independente de como ele for distribuído, é chamado de ```distfile```. Voce pode obter o ```distfile``` a partir de um CDROM ou da internet.

4.5.2.1. Instalando um aplicativo através do *ports* via CDROM

Os ISOs oficiais distribuídos pelo projeto FreeBSD não incluem nenhum ```distfile```. Eles ocupariam muito espaço, o qual é melhor utilizado pelos pacotes pré-compilados. Os CDROMs da linha *PowerPack* incluem ```distfiles```, e voce pode comprar este conjunto de CDs em vendedores como a [FreeBSD Mall](#). Esta seção assume que você possui este conjunto de CDs.

Insira o CDROM do FreeBSD no ```drive```. Monte-o como `/cdrom`. Se você utilizar um ponto de montagem diferente o processo de instalação não irá funcionar. Para iniciar, vá para o diretório do *ports* que você deseja instalar:

```
# cd /usr/ports/sysutils/lsof
```

Uma vez dentro do diretório `lsof`, você irá ver o esqueleto do *port*. O próximo passo é compilar, ou construir, o *port*. Isto é realizado pela simples execução do comando `make`. Ao executar o comando, você irá visualizar uma saída semelhante a esta:

```
# make
>> lsof_4.57D.freebsd.tar.gz doesn't seem to exist in
/usr/ports/distfiles/.
>> Attempting to fetch from file:/cdrom/ports/distfiles/.
===> Extracting for lsof-4.57
...
[extraction output snipped]
...
>> Checksum OK for lsof_4.57D.freebsd.tar.gz.
===> Patching for lsof-4.57
```



```
====> Applying FreeBSD patches for lsof-4.57
====> Configuring for lsof-4.57
...
[configure output snipped]
...
====> Building for lsof-4.57
...
[compilation output snipped]
...
#
```

Observe que após completar o processo de compilação você irá voltar para o prompt. O próximo passo é instalar o *port*. Para instalá-lo, basta adicionar a palavra `install` ao comando `make`, como mostrado abaixo:

```
# make install
====> Installing for lsof-4.57
...
[installation output snipped]
...
====> Generating temporary packing list
====> Compressing manual pages for lsof-4.57
====> Registering installation for lsof-4.57
====> SECURITY NOTE:
      This port has installed the following binaries which execute
with
      increased privileges.
#
```

Uma vez que você tenha retornado ao prompt, você já deve ser capaz de executar a aplicação que acabou de instalar. Como o aplicativo `lsof` executa com privilégios altos, um aviso de segurança é exibido. Durante o processo de compilação e de instalação, outros avisos podem ser exibidos.

Nota: Você pode economizar um passo executando diretamente o comando `make install`, ao invés de executar o comando `make` seguido do comando `make install` como duas etapas separadas.

Nota: Alguns *shells* mantêm um *cache* dos comandos disponíveis nos diretórios listados na variável de ambiente `PATH`, com o objetivo de agilizar o processo de busca para a execução desses comandos. Se você estiver utilizando um destes *shells*, você terá que executar o comando `rehash` logo após a instalação do *ports*, antes que você possa utilizar o aplicativo recém instalado. Esta operação é necessária com todos os shells disponibilizados na base do sistema (como por exemplo o `tcsh`) e com alguns dos disponibilizados no *ports* (por exemplo, [shells/zsh](#)).

Nota: Por favor, esteja ciente de que a licença de alguns aplicativos não permite a inclusão dos mesmos em um CDROM. Isto pode ocorrer devido a necessidade de se preencher algum formulário antes de se fazer o *download* ou então porque a redistribuição não é permitida, ou por qualquer outra razão. Se você deseja instalar um *ports* cujo `distfile` não esteja presente no CDROM, você terá que estar online para instalá-lo (consulte a [próxima seção](#)).

4.5.2.2. Instalando um aplicativo através do *ports* via Internet

Como na última seção, esta seção assume que você possui uma conexão internet funcional. Se você não possuir, você terá que executar a [instalação a partir do CDROM](#), ou então colocar manualmente uma cópia do ``distfile" do *port* desejado no diretório `/usr/ports/distfiles`.

A instalação de um *port* a partir da internet é realizada de forma idêntica a instalação a partir do CDROM, a única diferença entre as duas é que agora o ``distfile" será obtido da internet ao invés de ser lido do CDROM.

Os passos envolvidos são idênticos:

```
# make install
>> lsof_4.57D.freebsd.tar.gz doesn't seem to exist in
/usr/ports/distfiles/.
>> Attempting to fetch from
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/.
Receiving lsof_4.57D.freebsd.tar.gz (439860 bytes): 100%
439860 bytes transferred in 18.0 seconds (23.90 kBps)
==> Extracting for lsof-4.57
...
[extraction output snipped]
...
>> Checksum OK for lsof_4.57D.freebsd.tar.gz.
==> Patching for lsof-4.57
==> Applying FreeBSD patches for lsof-4.57
==> Configuring for lsof-4.57
...
[configure output snipped]
...
==> Building for lsof-4.57
...
[compilation output snipped]
...
==> Installing for lsof-4.57
...
[installation output snipped]
...
==> Generating temporary packing list
==> Compressing manual pages for lsof-4.57
==> Registering installation for lsof-4.57
==> SECURITY NOTE:
    This port has installed the following binaries which execute
with
increased privileges.
#
```

Como voce pode ver a única diferença é a linha que informa que o sistema esta baixando o ``distfile".

O sistema de *ports* utiliza o [fetch\(1\)](#) para baixar os arquivos, o qual considera várias variáveis de ambiente, incluindo `FTP_PASSIVE_MODE`, `FTP_PROXY`, e `FTP_PASSWORD`. Você pode precisar definir uma ou mais delas caso esteja atrás de um firewall, ou mesmo fazer uso de um proxy FTP/HTTP, consulte [fetch\(3\)](#) para visualizar a lista de todas as variáveis.

Para os usuários que não podem ficar conectados o tempo todo, é oferecido o comando `make fetch` como opção. Ao executar este comando na raiz da árvore do `ports` (`/usr/ports`), ele irá baixar todos os "distfiles" necessários para o uso do `ports` para você. Este comando também funciona para as demais categorias do segundo nível (por exemplo, `/usr/ports/net`). Observe que se um `port` depender de uma biblioteca ou de outros `ports` para funcionar, o comando acima não irá baixar os "distfiles" destes `ports`. Se você desejar que sejam baixadas todas as dependências, substitua a opção `fetch` por `fetch-recursive`.

Nota: Você pode compilar todos os `ports` de uma determinada categoria, ou mesmo todos os `ports`, executando o comando `make` na raiz da árvore do `ports`, exatamente como mencionado acima para o `make fetch`. Isto é perigoso, pois alguns `ports` não podem co-existir no mesmo sistema. Em outros casos, alguns `ports` podem instalar 2 arquivos diferentes com o mesmo nome, um sobrescrevendo o arquivo do outro.

Em alguns casos raros, os usuários podem precisar obter os `tarballs`, a partir de um sítio diferente do especificado no `MASTER_SITES` (local de onde os "distfiles" são obtidos). Você pode sobrescrever o valor da opção `MASTER_SITES` com o seguinte comando:

```
# cd /usr/ports/directory
# make MASTER_SITE_OVERRIDE= \
    ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/ fetch
```

No exemplo acima, nós alteramos a opção `MASTER_SITES` para `ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/`.

Nota: Alguns `ports` permitem (ou mesmo precisam) que você forneça algumas opções no momento da compilação, com o objetivo de habilitar ou desabilitar partes da aplicação que não são necessárias, opções de segurança e outras customizações. Alguns `ports` onde ocorrem isso, e que vem a mente no momento são [www/mozilla](http://www.mozilla.com), security/gpgme, e mail/sylpheed-claws. Uma mensagem é exibida quando este tipo de opções estiverem disponíveis.

4.5.2.3. Sobrescrevendo o diretório *default* do *Ports*

Algumas vezes é útil (ou obrigatório) utilizar diferentes "distfiles" e diretórios de `ports`. As variáveis `PORTSDIR` e `PREFIX` podem sobrescrever os diretórios *default*. Por exemplo:

```
# make PORTSDIR=/usr/home/example/ports install
```

Irá compilar o `port` disponível no diretório `/usr/home/example/ports` e instalar tudo sob o diretório `/usr/local`.

```
# make PREFIX=/usr/home/example/local install
```

Irá compilar o `port` em `/usr/ports` e instalar tudo em `/usr/home/example/local`.

E é claro,

```
# make PORTSDIR=./ports PREFIX=./local install
```

Irá combinar os dois (é muito longo para escrever nesta página, mas já deu para você ter uma idéia geral de como isso funciona).

Alternativamente, estas variáveis podem fazer parte do seu ambiente. Leia o manual do seu shell para ver instruções de como fazer desta forma.

4.5.2.4. Lidando com o `imake`

Alguns *ports* que fazem uso do `imake` (uma parte do Sistema X Windows) não funcionam bem com a variável `PREFIX`, e irão insistir em instalar o aplicativo sob `/usr/X11R6`. De forma semelhante, alguns *ports* Perl ignoram a variável `PREFIX` e se instalam sob a árvore do Perl. Fazer estes *ports* respeitarem a variável `PREFIX` é um trabalho difícil, senão impossível.

4.5.3. Desinstalando um Ports

Agora que você sabe como instalar um *ports*, você provavelmente está se perguntando como remove-lo do sistema, apenas para o caso de você instalar um e mais tarde descobrir que instalou o *port* errado. Nós iremos remover o nosso exemplo anterior (era o `lsof` caso não tenha prestado atenção). Assim como na instalação, você primeiro deve entrar no diretório do *port*, no nosso caso `/usr/ports/sysutils/lsof`. Depois que você tiver entrado no diretório, você estará pronto para desinstalar o `lsof`. O que é feito através do comando `make deinstall`:

```
# cd /usr/ports/sysutils/lsof
# make deinstall
====> Deinstalling for lsof-4.57
```

Simple assim, você já removeu o `lsof` do seu sistema. Se você desejar reinstalá-lo, basta executar o comando `make reinstall` dentro do diretório `/usr/ports/sysutils/lsof`.

Os comandos `make deinstall` e `make reinstall`, não irão funcionar depois que voce executar o comando `make clean`. Se voce desejar desinstalar um *ports* depois de ter limpado o diretório de trabalho, utilize o comando [pkg_delete\(1\)](#) como mostrado na [Sessão sobre pacotes do Handbook](#).

4.5.4. Ports e espaço em disco

Utilizar a coleção de *ports* pode definitivamente devorar espaço do seu HD. Por esta razão você precisa se lembrar de limpar o diretório de trabalho utilizando o comando `make clean`. Ao executá-lo ele irá remover o diretório `work` depois do *port* ter sido compilado e instalado. Você também pode remover os arquivos *tarballs* do diretório `distfiles`, além de remover os aplicativos instalados quando eles não tiverem mais utilidade.

Alguns usuários preferem limitar as categorias do *port*, colocando uma entrada no arquivo *refuse*. Desta forma quando eles executam o **CVSup**, ele não irá baixar nenhum arquivo desta categoria.

4.5.5. Atualizando um aplicativo instalado via *Ports*

Manter um aplicativo atualizado pode ser uma tarefa tediosa. Pois para atualizar um aplicativo você precisa ir ao diretório do *ports*, compilar o *port*, desinstalar a versão antiga do *port*, instalar a nova versão e então limpar o diretório de trabalho após a compilação. Imagine fazer isso para 5 aplicativos, entediante certo ? Este é um dos grandes problemas com que um administrador de sistema tem que lidar, mas agora temos um utilitário para fazer este trabalho para nós. Por exemplo o utilitário `sysutils/portupgrade` irá fazer tudo para você. Instale-o da mesma forma como faz com qualquer outro port, usando o comando `make install clean`.

Agora crie o banco de dados com o comando `pkgdb -F`. Este comando irá ler a lista de *ports* instalados em seu sistema e irá criar um banco de dados com informações dos mesmos no diretório `/var/db/pkg`. Agora quando você executar `portupgrade -a`, ele irá ler este banco de dados e o arquivo `INDEX` na raiz da árvore do *ports*. Em seguida o `portupgrade` irá iniciar o processo de download, compilação, backup, instalação e limpeza dos aplicativos que precisarem de atualização. Existem outros utilitários que fazem um trabalho semelhante, verifique as demais opções no diretório `ports/sysutils`.

4.6. Atividades pós-instalação

Depois de instalar um novo aplicativo você normalmente irá querer ler a documentação instalada com ele, editar algum arquivo de configuração necessário para o correto funcionamento do aplicativo, certificar-se de que o aplicativo irá iniciar automaticamente no próximo ``boot" (se ele for um servidor) e assim por diante.

Os passos exatos necessários para configurar cada aplicativo são obviamente diferentes. No entanto, se você acabou de instalar um novo aplicativo e está se perguntando ``O que fazer agora ? ", as dicas abaixo podem ajudá-lo:

- Utilize o comando [pkg_info\(1\)](#) para descobrir quais aplicativos estão instalados e onde eles estão instalados. Por exemplo, você tiver instalado o aplicativo *FooPackage* versão 1.0.0, então o comando
- `# pkg_info -L foopackage-1.0.0 | less`

Irá exibir todos os arquivos instalados com o pacote. Preste atenção especial aos arquivos instalados no diretório `man/`, pois eles são os manuais dos aplicativos que compoem o pacote, Também preste atenção aos arquivos instalados no diretório `etc/`, pois são os arquivos de configuração, e por fim preste atenção aos arquivos instalados no diretório `doc/`, pois nele ficam os documentos complementares sobre o aplicativo.

Se você não tem certeza de qual versão do aplicativo esta instalada, utilize um comando semelhante ao abaixo

```
# pkg_info | grep -i foopackage
```

Irá procurar todos os aplicativos instalados que possuem a palavra *foopackage* no nome. Substitua a palavra *foopackage* no seu comando de acordo com sua necessidade.

- Uma vez que você tiver identificado as paginas de manual instaladas com o seu aplicativo, utilize o comando [man\(1\)](#) para visualizá-las. De forma análoga, leia os arquivos de configuração instalados como exemplo, bem como qualquer outra documentação complementar que tenha sido disponibilizada.
- Se o aplicativo possuir um sítio *web*, faça uma visita e verifique se ele disponibiliza alguma documentação adicional, uma lista com as perguntas e repostas mais frequentes, e assim por diante. Se você não tiver certeza do endereço do sítio *web* do aplicativo, ele pode ser listado na saída do comando
- ```
pkg_info foopackage-1.0.0
```

Verifique se existe uma linha iniciada por *WWW:* , se ela estiver presente, você terá descoberto o endereço do sítio *web* do aplicativo.

- Os aplicativos que inicializam no ``boot" (aplicativos servidores) irão instalar um script shell no diretório `/usr/local/etc/rc.d`. Voce deve verificar este script para ter certeza de que está correto, editando-o ou renomeando-o se for necessário. Consulte a sessão [Iniciando Serviços](#) para maiores informações.

---

## 4.7. Lidando com ports quebrados

Se você encontrar um *port* que não está funcionando, existem algumas coisas que você pode fazer, incluindo:

1. Conserte-o! O [Porter's Handbook](#) inclui informações detalhadas sobre a infraestrutura do ``Ports", de forma que você pode ocasionalmente corrigir um *port* quebrado ou então submeter o seu próprio!
2. Reclamar--*por email apenas!* Primeiro envie um email para o responsável pelo *port*. Execute `make maintainer` ou então leia o arquivo `Makefile` para encontrar o endereço de email do responsável pelo *port*. Lembre-se de incluir o nome e a versão do *port* (envie a linha `$FreeBSD:`, do arquivo `Makefile`) e a mensagem de erro no email que irá enviar. Se você não obtiver uma resposta do responsável, voce pode utilizar o [send-pr\(1\)](#) para enviar um relatório de falha.
3. Pegue o pacote em um servidor de FTP próximo de você. A coleção de pacotes ``master" fica no servidor `ftp.FreeBSD.org`, no diretório [packages](#), mas tenha certeza de verificar o seu *mirror* local *antes!* Esta abordagem é muito garantida de que vai funcionar do que tentar compilar o aplicativo apartir do código fonte, além de ser muito mais rápida também. Utilize o utilitário [pkg\\_add\(1\)](#) para instalar aplicativos no seu sistema.

# Capítulo 5. O X Window System

## Índice

- 5.1. [Sinopse](#)
- 5.2. [Compreendendo o X](#)
- 5.3. [Instalando o XFree86™](#)
- 5.4. [Configuração do XFree86™](#)
- 5.5. [Usando fontes no XFree86™](#)
- 5.6. [O gerenciador de exibição do X](#)
- 5.7. [Ambientes para a área de trabalho](#)

## 5.1. Sinopse

O FreeBSD usa **XFree86™** para prover aos usuários uma poderosa interface gráfica. **XFree86** é uma implementação de código aberto do *X Window System*. Este capítulo cobrirá a instalação e configuração do **XFree86** num sistema FreeBSD. Para mais informações sobre o **XFree86** e hardware de vídeo suportado, verifique o sítio [XFree86](#).

Após ler este capítulo, você saberá:

- Os vários componentes do *X Window System* e como eles interagem.
- Como instalar e configurar o **XFree86**.
- Como instalar e utilizar diferentes gerenciadores de janelas.
- Como utilizar fontes TrueType® no **XFree86**.
- Como preparar seu sistema para o uso de *login* gráfico (**XDM**).

Antes de ler este capítulo, é recomendável:

- Saber como instalar softwares adicionais de terceiros ([Capítulo 4](#)).

---

## 5.2. Compreendendo o X

Usar o X pela primeira vez pode causar certo impacto em quem já está familiarizado com outros ambientes gráficos, como o Microsoft® Windows® ou o Mac OS®.

Não é necessário entender todos os detalhes dos vários componentes do X e como eles interagem; entretanto, algum conhecimento básico torna possível tirar vantagens dos pontos positivos do X.

### 5.2.1. Por que X?

O X não é o primeiro sistema de interface gráfica escrito para o UNIX®, mas é o mais popular. O grupo de desenvolvimento original do X já havia trabalhado em outro sistema de interface gráfica antes de escrever o X. O nome desse sistema era ``W" (de ``Window"). X é apenas a próxima letra no alfabeto Romano.

O X pode ser chamado de ``X'', ``X Window System'' (Sistema de Janela X), ``X11'', e outros termos. Chamar X11 de ``X Windows'' (n.t. note o ``s'') pode ofender algumas pessoas; veja [X\(7\)](#) para mais dicas sobre isso.

### 5.2.2. O modelo Cliente/Servidor do X

O X foi concebido desde o início com foco na rede, e adota um modelo ``cliente-servidor''. No modelo do X, o ``servidor X'' executa em um computador que possui teclado, monitor e mouse conectados. O servidor é responsável por gerenciar o vídeo, tratar as entradas pelo teclado e mouse, e assim por diante. Cada aplicativo do X (como o **XTerm**, ou **Netscape®**) é um ``cliente''. Um cliente envia mensagens para o servidor como ``Por favor, desenhe uma janela nestas coordenadas'', e o servidor responde a mensagens como ``O usuário acabou de clicar o botão OK ''.

Se há apenas um computador envolvido, como em casa ou em um ambiente de escritório pequeno, o servidor X e os clientes X estarão executando no mesmo computador. Entretanto, é perfeitamente possível executar o servidor X em um computador de mesa menos poderoso, e executar os aplicativos do X (os clientes), por assim dizer, naquela máquina poderosa e caríssima que serve a todo o escritório. Neste cenário a comunicação entre o cliente X e o servidor se dá na rede.

Isto causa confusão a algumas pessoas, pois a terminologia do X é exatamente contrária à que eles esperam. O que se espera é que o ``servidor X '' seja a máquina grande e poderosa do porão e o ``cliente X'' a máquina sobre suas mesas.

Lembre-se que o servidor X é a máquina com monitor e teclado, e os clientes X são os programas que exibem as janelas.

Não há nada no protocolo que force máquinas clientes e servidoras a estarem executando o mesmo sistema operacional ou ainda no mesmo tipo de computador. É certamente possível executar um servidor X no Microsoft Windows ou no Mac OS da Apple, e há várias aplicações gratuitas e comerciais disponíveis que fazem exatamente isso.

O servidor X que é distribuído com o FreeBSD é chamado **XFree86™**, e está disponível gratuitamente, sob uma licença muito similar à licença do FreeBSD. Há também servidores X comerciais disponíveis para o FreeBSD.

### 5.2.3. O Gerenciador de Janela

A filosofia de concepção do X é muito semelhante à filosofia de concepção do UNIX, ``ferramentas, não políticas''. Isso significa que o X não tenta dizer como a tarefa deve ser realizada. Ao invés disso, as ferramentas são fornecidas ao usuário, e é sua responsabilidade decidir como vai usar essas ferramentas.

Essa filosofia se estende ao X quando o mesmo não dita com o quê as janelas devem parecer na exibição, como movê-las com o mouse, que atalho deve ser usado para mover-se entre as janelas (i.e., **Alt+Tab**, no caso do Microsoft Windows), a aparência da barra de títulos em cada janela, se elas devem ou não apresentar botões para fechá-las e assim por diante.



Ao invés disso, o X delega essa responsabilidade para um aplicativo chamado "Gerenciador de Janela" (do inglês "Window Manager"). Há dúzias de gerenciadores de janela disponíveis para o X: **AfterStep**, **Blackbox**, **ctwm**, **Enlightenment**, **fvwm**, **Sawfish**, **twm**, **Window Maker**, e outras. Cada um desses gerenciadores de janela provê diferentes visuais e sensações; alguns deles suportam "áreas de trabalho virtuais"; alguns permitem personalizar atalhos de teclado para gerenciar a área de trabalho; outros tem um botão "Iniciar" ou dispositivo semelhante; outros são "tematizáveis", permitindo uma modificação completa do visual e sensação pela aplicação de um novo tema. Estes gerenciadores de janela e muitos outros, estão disponíveis na categoria `x11-wm` da Coleção de *Ports*.

Para completar, ambos ambientes de trabalho **KDE** e o **GNOME** têm seus próprios gerenciadores de janela integrados na área de trabalho.

Cada gerenciador de janela também possui um mecanismo de configuração diferente; alguns esperam que a configuração seja escrita manualmente, outros possuem GUI como ferramentas para a maior parte das tarefas de configuração; pelo menos um (**sawfish**) tem seu arquivo de configurações escrito em um dialeto da linguagem Lisp.

**Política do Foco:** Outro recurso de responsabilidade do gerenciador de janela é a "política de foco" do mouse. Todo sistema de enjanelamento precisa de parâmetros para escolher uma janela que responda ao acionamento do teclado, e deve indicar visivelmente qual janela está ativa também.

Uma política de foco bem familiar é a chamada "clique para focar". Este é o modelo utilizado pelo Microsoft Windows, no qual uma janela se torna ativa após receber um clique do mouse.

O X não suporta nenhuma política de foco em particular. Ao invés disso, o gerenciador de janela controla qual tem o foco a qualquer tempo. Os diferentes gerenciadores de janela vão suportar métodos de foco distintos. Todos suportam clique para focar e a maioria deles suporta vários outros.

As políticas de foco mais populares são:

o foco segue o mouse

A janela abaixo do ponteiro do mouse é a janela que possui o foco. Esta não é necessariamente a janela que está sobre todas as demais. O foco é modificado ao apontar outra janela, não há necessidade de clicar nela.

foco desleixado

Esta política é uma pequena extensão de "o foco segue o mouse". Com o foco seguindo o mouse, se o mouse é movido para a janela raiz (ou a área de trabalho) então nenhuma janela tem o foco, e o acionamento do teclado é simplesmente perdido. Com o foco desleixado, o foco é modificado apenas quando o cursor entra em uma nova janela, e não quando se sai da janela atual.

clique para focar

A janela ativa é selecionada pelo clique do mouse. A janela pode então ser "elevada" e aparecer na frente de todas as demais janelas. Todo acionamento do teclado será agora direcionado para essa janela, mesmo se o cursor for movido para outra janela.

Muitos gerenciadores de janela suportam outras políticas, bem como variações nestas. Certifique-se de consultar a documentação própria do gerenciador de janela.

#### 5.2.4. Widgets

A abordagem do X de prover ferramentas e não políticas se estende para os *widgets* vistos nas telas de cada aplicativo.

"Widget" é um termo para todos os itens da interface com o usuário que podem ser clicados ou manipulados de alguma forma; botões, *check boxes*, *radio buttons*, ícones, listas, e outros. O Microsoft Windows chama estes de "controles".

O Microsoft Windows e o Mac OS da Apple's têm uma política muito rígida para os *widgets*. Espera-se que os desenvolvedores de aplicações compartilhem o mesmo visual e sensação em seus aplicativos. Com o X, não foi considerado sensato determinar um estilo gráfico particular, ou conjunto de *widgets* a aderir.

Como resultado, não espere que as aplicações do X tenham o mesmo visual e sensação. Há vários conjuntos de *widgets* populares e variações, incluindo o conjunto original de *widgets* Athena do MIT, **Motif**® (no qual o conjunto de *widgets* do Microsoft Windows foi modelado, todas as bordas chanfradas em três tons de cinza), **OpenLook**, e outros.

A maior parte das novas aplicações para o X vão utilizar um conjunto de *widgets* de visual moderno, ou Qt, utilizado pelo **KDE**, ou **GTK**, utilizado pelo projeto **GNOME**. No que diz respeito a isso, há alguma convergência no visual e sensação da área de trabalho do UNIX, o que certamente torna tudo mais fácil para os novos usuários.

---

## 5.3. Instalando o XFree86™

Antes de instalar o **XFree86**™, decida que versão executar. O **XFree86 3.X** é um ramo de manutenção do desenvolvimento do **XFree86**. É muito estável e suporta um número bem grande de placas vídeo. Entretanto, nenhum recurso novo está sendo desenvolvido para o software. O **XFree86 4.X** é uma reconcepção do sistema com diversos novos recursos como melhor suporte para fontes e *anti-aliasing*. Infelizmente esta nova arquitetura requer que os drivers de vídeo sejam reescritos, e algumas placas mais velhas que eram suportadas no 3.X não são ainda suportadas no 4.X. Como todo desenvolvimento novo e suporte para as placas de vídeo são feitos nesse ramo, o **XFree86 4.X** é agora a versão padrão do sistema de janela X no FreeBSD.

O programa de instalação do FreeBSD oferece aos usuários a oportunidade de instalar e configurar o **XFree86 4.X** durante a instalação (detalhes na [Seção 2.9.12](#)). Para instalar e executar o **XFree86 3.X**, aguarde até que o FreeBSD mínimo seja instalado, e então

instale o **XFree86**. Por exemplo, para construir e instalar o **XFree86 3.X** da coleção de *ports*:

```
cd /usr/ports/x11/XFree86
make all install clean
```

Como alternativa, quaisquer versões do **XFree86** podem ser instaladas diretamente dos binários do FreeBSD fornecidos no [sítio XFree86](#). Um pacote binário a ser usado com a ferramenta [pkg\\_add\(1\)](#) também está disponível para o **XFree86 4.X**. Quando o recurso de baixar da fonte remota do [pkg\\_add\(1\)](#) é utilizado, o número da versão do pacote deve ser removido. O [pkg\\_add\(1\)](#) vai baixar automaticamente a versão mais recente do aplicativo. Logo para baixar e instalar o pacote do **XFree86 4.X**, é só digitar:

```
pkg_add -r XFree86
```

Você também pode utilizar a coleção de *ports* para instalar o **XFree86 4.X**, para isso você precisa apenas digitar os comandos a seguir:

```
cd /usr/ports/x11/XFree86-4
make install clean
```

**Nota:** Os exemplos acima vão instalar a distribuição completa do **XFree86** incluindo os servidores, clientes, fontes etc. Pacotes individuais e *ports* para diferentes partes do **XFree86 4.X** também estão disponíveis.

O restante deste capítulo explicará como configurar o **XFree86** e como instalar um ambiente produtivo na área de trabalho.

---

## 5.4. Configuração do XFree86™

*Contribuição de Christopher Shumway.*

### 5.4.1. Antes de começar

Antes de configurar o **XFree86™ 4.X**, é necessário conhecer as seguintes informações sobre o sistema alvo:

- As especificações do monitor
- O *chipset* da placa de vídeo
- A quantidade de memória da placa de vídeo

As especificações do monitor são usadas pelo **XFree86** para determinar a resolução e taxa de atualização em que deve operar. Estas especificações podem geralmente ser obtidas da documentação que vem com o monitor ou a partir do sítio do fabricante. São necessários dois intervalos numéricos, a taxa de varredura horizontal e a taxa de sincronização vertical.

O *chipset* da placa de vídeo define qual módulo de *driver* o **XFree86** usa para conversar com o hardware. Com a maioria dos *chipsets* isso pode ser determinado automaticamente, mas ainda é útil saber identificá-lo caso a detecção automática não funcione corretamente.

A memória da placa de vídeo determina que resolução e a profundidade das cores nas quais o sistema vai executar. É importante que o usuário conheça isso para saber das limitações do sistema.

#### 5.4.2. Configurando o XFree86 4.X

A configuração do **XFree86 4.X** é um processo de vários passos. O primeiro passo é construir um arquivo de configuração inicial com a opção `-configure` do **XFree86**. Como o super usuário, simplesmente execute:

```
XFree86 -configure
```

Isso vai gerar o esqueleto do arquivo de configuração do **XFree86** no diretório `/root` chamado `XF86Config.new` (na verdade a pasta utilizada é a registrada na variável de ambiente `$HOME` e ela vai depender de como você conseguiu os direitos de super usuário). O programa do **XFree86** vai tentar investigar o hardware da placa de vídeo no sistema e vai escrever um arquivo de configuração que carregue os drivers adequados ao hardware detectado no sistema alvo.

O próximo passo é testar a configuração existente para verificar se o **XFree86** pode trabalhar com o hardware do sistema alvo. Para fazer esta tarefa, o usuário precisa executar:

```
XFree86 -xf86config XF86Config.new
```

Se a grade preto e cinza e um ponteiro de mouse do X aparecem, a configuração foi bem sucedida. Para sair do teste, basta pressionar simultaneamente **Ctrl+Alt+Backspace**.

**Nota:** Se o mouse não funcionar, certifique-se que o dispositivo foi configurado. Consulte a [Seção 2.9.10](#) no capítulo de instalação do FreeBSD.

Em seguida, ajuste o arquivo de configurações `XF86Config.new` a seu gosto. Abra o arquivo em um editor de texto como o [emacs\(1\)](#) ou [ee\(1\)](#). Primeiro, adicione as frequências do monitor do seu sistema. Elas geralmente são expressas como taxas de sincronização horizontal e vertical. Estes valores são adicionados ao arquivo `XF86Config.new` sob a seção "Monitor":

```
Section "Monitor"
 Identifier "Monitor0"
 VendorName "Monitor Vendor"
 ModelName "Monitor Model"
 HorizSync 30-107
 VertRefresh 48-120
EndSection
```

As expressões *HorizSync* e *VertRefresh* podem não existir no arquivo de configurações. Se for o caso, elas precisam ser adicionadas, com a taxa de sincronização horizontal posicionada depois da expressão *Horizsync* e a taxa de sincronização vertical após a expressão *VertRefresh*. Confira no exemplo acima as taxas de um monitor já preenchidas.

O X permite que recursos DPMS (*Energy Star*) sejam utilizados em monitores com suporte. O programa [xset\(1\)](#) controla os intervalos entre atividades e pode forçar os modos de espera, suspenso e desligado. Se você deseja habilitar os recursos de DPMS para o seu monitor, você deve acrescentar a linha a seguir na seção Monitor:

```
Option "DPMS"
```

Enquanto o arquivo de configurações `XF86Config.new` ainda estiver aberto no editor, selecione a resolução padrão do monitor e a profundidade das cores desejada. Isso está definido na seção "Screen":

```
Section "Screen"
 Identifier "Screen0"
 Device "Card0"
 Monitor "Monitor0"
 DefaultDepth 24
 SubSection "Display"
 Depth 24
 Modes "1024x768"
 EndSubSection
EndSection
```

A expressão *DefaultDepth* descreve a profundidade de cores empregada por padrão. Isso pode ser sobreposto com a opção da linha de comando `-bpp` existente para o [XFree86\(1\)](#). A expressão *Modes* descreve a resolução utilizada para a profundidade de cores dada. Observe que apenas os modos do padrão VESA são suportados pela definição do hardware gráfico do sistema alvo. No exemplo acima, a profundidade de cores padrão é de vinte e quatro bits por ponto. Nesta profundidade de cores, a resolução aceita é de mil e vinte e quatro pontos por setecentos e sessenta e oito pontos.

Por fim, salve o arquivo de configurações e teste-o utilizando o modo de teste dado acima. Se tudo estiver bem, o arquivo de configuração precisa ser instalado em um local conhecido pelo [XFree86\(1\)](#), a fim de encontrá-lo. Este local é tipicamente a pasta `/etc/X11/XF86Config` ou `/usr/X11R6/etc/X11/XF86Config`.

```
cp XF86Config.new /etc/X11/XF86Config
```

Uma vez que o arquivo de configuração tenha sido colocado em um lugar conhecido, a configuração está completa. Para iniciar o **XFree86 4.X** com [startx\(1\)](#), instale o [port x11/wrapper](#). O **XFree86 4.X** também pode ser executado via [xdm\(1\)](#).

**Nota:** Também existe uma ferramenta gráfica capaz de fazer a configuração, o [xf86cfg\(1\)](#), que vem na distribuição do **XFree86 4.X**. Ela permite definir interativamente sua configuração pela escolha dos *drivers* e parâmetros apropriados. Este programa pode ser utilizado no console também, basta utilizar o comando `xf86cfg -textmode`. Para mais detalhes, verifique a página do manual [xf86cfg\(1\)](#).

### 5.4.3. Tópicos de configuração avançados

#### 5.4.3.1. Configuração dos *Chipsets* Gráficos Intel® i810

A configuração dos *chipsets* integrados da Intel® i810 requerem que a interface de programação para o AGP e `agpgart` do **XFree86** seja instalada para controlar a placa. O *driver* [agp\(4\)](#) está no *kernel* `GENERIC` desde as *releases* 4.8-RELEASE e 5.0-RELEASE. Nas *releases* anteriores, você deve adicionar as linhas a seguir:

```
device agp
```

no arquivo de configuração do seu *kernel* e reconstruir um novo *kernel*. Ao invés disso, você pode querer carregar o módulo `agp.ko` do *kernel* automaticamente com o [loader\(8\)](#) durante o processo de inicialização. Para isso, basta adicionar esta linha ao `/boot/loader.conf`:

```
agp_load="YES"
```

A seguir, se estiver utilizando o FreeBSD 4.X ou posterior, um novo nó para o dispositivo (*device node*) precisa ser criado para a interface de programação. Para criar o nó para o dispositivo AGP, execute o [MAKEDEV\(8\)](#) no diretório `/dev`:

```
cd /dev
sh MAKEDEV agpgart
```

**Nota:** O FreeBSD 5.X ou posterior vai utilizar o [devfs\(5\)](#) para alocar os nós dos dispositivos de modo transparente, logo, o passo do [MAKEDEV\(8\)](#) não é necessário.

Isso vai permitir a configuração do dispositivo como se faz com qualquer outra placa de vídeo. Observe que em sistemas sem o *driver* [agp\(4\)](#) compilado no *kernel*, tentar carregar o módulo com o [kldload\(8\)](#) não vai funcionar. Este *driver* precisa estar no *kernel* durante o processo de inicialização, seja sendo compilado ou pelo uso do `/boot/loader.conf`.

Se estiver utilizando o **XFree86 4.1.0** (ou posterior) e mensagens sobre símbolos não resolvidos como `fbPictureInit` aparecerem, tente adicionar a linha a seguir após *Driver "i810"* no arquivo de configuração do **XFree86**:

```
Option "NoDDC"
```

---

## 5.5. Usando fontes no XFree86™

Com contribuições de Murray Stokely.

### 5.5.1. Fontes *Type1*

As fontes padrão que são distribuídas com o **XFree86™** estão aquém do ideal para aplicações comuns de editoração eletrônica. Fontes grandes em apresentações aparecem

recortadas e com aparência nada profissional, e fontes pequenas no **Netscape®** são quase ininteligíveis. Entretanto, há várias fontes *Type1* (PostScript®) gratuitas e de alta qualidade disponíveis, as quais podem ser usadas imediatamente com o **XFree86**, seja a versão 3.X ou a versão 4.X. Por exemplo, a coleção de fontes URW ([x11-fonts/urwfonts](#)) inclui versões de alta qualidade de fontes *type1* padrão (Times Roman®, Helvetica®, Palatino® e outras). A coleção *Freefonts* ([x11-fonts/freefonts](#)) inclui muitas outras fontes, mas a maioria delas são destinadas ao uso em softwares de manipulação de gráficos como o **Gimp**, não sendo suficientemente completas para o uso como fontes de tela. Para completar, o **XFree86** pode ser configurado para usar fontes TrueType® com um mínimo de esforço: veja a [seção sobre fontes TrueType](#) adiante.

Para instalar as coleções de fontes *Type1* acima da coleção de *ports*, execute os comandos a seguir:

```
cd /usr/ports/x11-fonts/urwfonts
make install clean
```

E da mesma forma para o *fontfont* e outras coleções. Para dizer ao servidor X que estas fontes existem, adicione a linha apropriada ao arquivo `XF86Config` (em `/etc/` para **XFree86** versão 3, ou em `/etc/X11/` para a versão 4), onde se lê:

```
FontPath "/usr/X11R6/lib/X11/fonts/URW/"
```

Alternativamente, na linha de comandos da sessão X execute:

```
% xset fp+ /usr/X11R6/lib/X11/fonts/URW
% xset fp rehash
```

Isso vai funcionar mas será perdido quando a sessão do X for fechada, a menos que seja adicionada ao arquivo de inicialização (`~/.xinitrc` para uma sessão normal via `startx`, ou `~/.xsession` quando o registro (*login*) for feito via um gerenciador de registro gráfico como o **XDM**). Uma terceira forma é utilizar o novo arquivo `XftConfig`: consulte a seção sobre [anti-aliasing](#).

### 5.5.2. Fontes TrueType®

O **XFree86 4.X** tem suporte incorporado para a exibição produzida de fontes TrueType. Há dois módulos diferentes que podem permitir essa funcionalidade. O módulo *freetype* é utilizado neste exemplo por ser mais coerente com outros suportes para produção de fontes. Para habilitar o módulo *freetype* basta acrescentar a linha a seguir à seção `Module` do arquivo `/etc/X11/XF86Config`.

```
Load "freetype"
```

Para o **XFree86 3.3.X**, um servidor de fontes TrueType separado se faz necessário. O **Xfstt** é de uso comum para esta finalidade. Para instalar o **Xfstt**, basta instalar o *port* [x11-servers/Xfstt](#).

Crie uma pasta para as fontes TrueType (por exemplo, `/usr/X11R6/lib/X11/fonts/TrueType`) e copie todas as fontes TrueType nesta pasta.

Tenha em mente que as fontes TrueType não podem ser importadas diretamente de um Macintosh®; elas precisam estar no formato UNIX®/DOS/Windows® para que possam ser usadas pelo **XFree86**. Uma vez que os arquivos tenham sido copiados para esta pasta, use o **ttmkfdir** para criar um arquivo `fonts.dir`, de modo que o produtor de fontes do X saiba que estes novos arquivos foram instalados. O `ttmkfdir` está disponível na Coleção de *Ports* do FreeBSD como [x11-fonts/ttmkfdir](http://x11-fonts/ttmkfdir).

```
cd /usr/X11R6/lib/X11/fonts/TrueType
ttmkfdir > fonts.dir
```

Agora adicione a pasta TrueType no caminho para as fontes (o *font path*). Esta é a mesma descrição utilizada anteriormente para as fontes [Type1](#), isto é, utilize

```
% xset fp+ /usr/X11R6/lib/X11/fonts/TrueType
% xset fp rehash
```

ou adicione uma linha *FontPath* ao arquivo `XF86Config`.

Isso é tudo. Agora o **Netscape**, o **Gimp**, o **StarOffice™**, e todos os outros aplicativos do X devem estar reconhecendo as fontes TrueType instaladas. Fontes muito pequenas (como um texto em um vídeo de alta resolução de uma página da internet) e fontes muito grandes (dentro do **StarOffice**) vão ter uma aparência muito melhor agora.

### 5.5.3. Fontes com Anti-Alias

*Atualizado para o XFree86 4.3 por Joe Marcus Clarke.*

O *Anti-aliasing* está disponível no **XFree86** desde a versão 4.0.2. Mesmo assim, configurar tais fontes era incômodo antes da introdução do **XFree86** 4.3.0. Da versão 4.3.0 em diante, todas as fontes em `/usr/X11R6/lib/X11/fonts/` e em `~/ .fonts/` são automaticamente disponibilizadas com *anti-aliasing* para os aplicativos prontos para o Xft. Nem todas as aplicações são preparadas para usar o Xft ainda, mas muitas já contam com o suporte Xft. Exemplos de aplicações que reconhecem o Xft incluem o Qt 2.3 e posteriores (conjunto de ferramentas feito para a área de trabalho do **KDE**), Gtk+ 2.0 e posteriores (conjunto de ferramentas para a área de trabalho do **GNOME**), e o **Mozilla** 1.2 e posteriores.

Para distinguir que fontes são *anti-aliased*, ou configurar as propriedades do *anti-aliasing*, crie (ou edite, se já existir) o arquivo `/usr/X11R6/etc/fonts/local.conf`. Vários recursos avançados do sistema de fontes Xft podem ser ajustadas através deste arquivo; esta seção descreve apenas algumas possibilidades simples. Para mais detalhes, por favor verifique em [fonts-conf\(5\)](#).

Este arquivo deve ser em formato XML. Preste bastante atenção à caixa dos caracteres e assegure-se de ter fechado todos os rótulos (*tags*). O arquivo começa com o cabeçalho usual do XML seguido de uma definição DOCTYPE, e então pelo rótulo `<fontconfig>`:

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```



Como citado anteriormente, toda fonte em `/usr/X11R6/lib/X11/fonts/` bem como em `~/.fonts/` são automaticamente disponibilizadas para aplicações prontas para o Xft. Se você deseja acrescentar outra pasta que esteja fora dessas duas hierarquias, acrescente uma linha similar a apresentada abaixo ao arquivo

```
/usr/X11R6/etc/fonts/local.conf:
```

```
<dir>/path/to/my/fonts</dir>
```

Após acrescentar novas fontes, e especialmente novas pastas com fontes, você deve executar o comando a seguir para reconstruir os *caches* de fonte:

```
fc-cache -f
```

O *anti-aliasing* torna as bordas pouco nítidas, o que proporciona a textos pequenos maior legibilidade e remove "escadinhas" de textos grandes, mas pode causar fadiga visual se aplicado ao texto normal. Para excluir fontes com tamanho em pontos menor 14 do *anti-aliasing*, inclua estas linhas:

```
<match target="font">
 <test name="size" compare="less">
 <double>14</double>
 </test>
 <edit name="antialias" mode="assign">
 <bool>>false</bool>
 </edit>
</match>
```

O espaçamento para fontes monoespaçadas também pode ser inapropriado com o *anti-aliasing*. Isso parece ser uma questão presente no **KDE**, em particular. Uma correção possível para isso é forçar o espaçamento destas fontes para 100. Acrescente as seguintes linhas:

```
<match target="pattern" name="family">
 <test qual="any" name="family">
 <string>fixed</string>
 </test>
 <edit name="family" mode="assign">
 <string>mono</string>
 </edit>
</match>
<match target="pattern" name="family">
 <test qual="any" name="family">
 <string>console</string>
 </test>
 <edit name="family" mode="assign">
 <string>mono</string>
 </edit>
</match>
```

(isso apelidará todo nome comum para fontes fixas como "mono"), e então acrescente:

```
<match target="pattern" name="family">
 <test qual="any" name="family">
 <string>mono</string>
 </test>
 <edit name="spacing" mode="assign">
```

```
<int>100</int>
</edit>
</match>
```

Certas fontes, como a Helvética, podem apresentar problemas quando usadas com *anti-alias*. Geralmente isso se apresenta como uma fonte que parece cortada ao meio verticalmente. No pior caso, isso pode causar a parada na execução de aplicativos como o **Mozilla**. Para evitar isso, considere acrescentar o seguinte ao arquivo `local.conf`:

```
<match target="pattern" name="family">
 <test qual="any" name="family">
 <string>Helvetica</string>
 </test>
 <edit name="family" mode="assign">
 <string>sans-serif</string>
 </edit>
</match>
```

Uma vez terminada a edição do `local.conf` assegure-se de colocar no final do arquivo o rótulo `</fontconfig>`. Não fazê-lo fará que suas modificações sejam desprezadas.

O conjunto padrão de fontes que vem com o **XFree86** não é muito conveniente quando se trata de *anti-alias*. Um conjunto bem melhor de fontes padrão pode ser encontrado no [port `x11-fonts/bitstream-vera`](#). Esse *port* irá instalar um arquivo `/usr/X11R6/etc/fonts/local.conf` se ele não existir. Se o arquivo existir, o *port* vai criar o arquivo `/usr/X11R6/etc/fonts/local.conf-vera`. Faça a fusão do conteúdo deste arquivo com o `/usr/X11R6/etc/fonts/local.conf`, e as fontes Bitstream automaticamente substituirão as fontes Serif, Sans Serif e Monospaced padrão do **XFree86**.

Por fim, os usuários podem acrescentar suas próprias configurações em seus arquivos `.fonts.conf` pessoais. Para fazer isso, cada usuário deve apenas criar o arquivo `~/.fonts.conf`. Este arquivo precisa também estar no formato XML.

Um último ponto: com telas de LCD, amostragem de subpontos pode ser desejável. Isso basicamente trata os componentes (horizontalmente separados) vermelho, verde e azul separadamente a fim de melhorar a resolução horizontal; os resultados podem ser espetaculares. Para permitir isso, acrescente a linha em algum lugar do arquivo `local.conf`:

```
<match target="font">
 <test qual="all" name="rgba">
 <const>unknown</const>
 </test>
 <edit name="rgba" mode="assign">
 <const>rgb</const>
 </edit>
</match>
```

**Nota:** Conforme o tipo de tela, o `rgb` pode precisar ser substituído por `bgr`, `vrgb` ou `vbgr`: experimente e veja qual funciona melhor.

O *anti-aliasing* deve estar habilitado na próxima vez que o servidor X for iniciado. Entretanto, os programas precisam saber como tirar vantagem disso. No presente, o conjunto de ferramentas Qt o faz, logo todo o ambiente do **KDE** pode utilizar fontes com *anti-alias* (consulte a [Seção 5.7.3.2](#) sobre o **KDE** para detalhes). O Gtk+ e o **GNOME** também podem ser adaptados ao *anti-alias* através do *caplet* ``Font" (consulte a [Seção 5.7.1.3](#) para mais detalhes). Por padrão, o **Mozilla** 1.2 e posteriores vão usar o *anti-alias* automaticamente. Para desativar isso, reconstrua o **Mozilla** com o parâmetro `-DWITHOUT_XFT`.

---

## 5.6. O gerenciador de exibição do X

*Contribuições de Seth Kingsley.*

### 5.6.1. Visão geral

O Gerenciador de Exibição do X (**XDM**; n.t. de *X Display Manager*) é uma parte opcional do Sistema de Janela X que é usado para gerenciar o *login* nas sessões. Isso é útil em vários tipos de situações, incluindo áreas de trabalho como ``Terminais X" mínimos e grandes servidores de exibição para rede. Como o Sistema de Janela X é independente de rede e protocolos, há uma grande variedade de configurações possíveis para a execução de clientes e servidores X em máquinas diferentes e conectadas a rede. O **XDM** fornece uma interface gráfica que possibilita escolher o servidor de exibição no qual se quer conectar e a entrada de informações de autorização como o a combinação de *login* e senha.

Pense no **XDM** como o provedor da mesma funcionalidade para o usuário proporcionada pelo utilitário [getty\(8\)](#) (veja [Seção 17.3.2](#) para detalhes). Isto é, ele faz o *login* no sistema da tela a que se está conectado e então executa um gerente de sessão a favor do usuário (geralmente um gerente de janela X). O **XDM** então aguarda o término do programa, sinalizando que o usuário terminou e que deve ser feito seu *log out* da tela. Neste ponto, o **XDM** pode exibir a tela de *login* e exibir telas com opções para que o próximo usuário faça seu *login*.

### 5.6.2. Usando o XDM

O programa *daemon* do **XDM** está localizado na pasta `/usr/X11R6/bin/xdm`. Este programa pode ser executado em qualquer momento como `root` e vai iniciar o gerenciamento da tela do X na máquina local. Se for executar o **XDM** a cada vez que o computador inicializar (*boot*), um modo conveniente de fazê-lo é adicionar uma entrada no arquivo `/etc/ttys`. Para mais informações sobre o uso e o formato deste arquivo, consulte [Seção 17.3.2.1](#). Há uma linha no arquivo `/etc/ttys` padrão para executar o *daemon* do **XDM** em um terminal virtual:

```
tttyv8 "/usr/X11R6/bin/xdm -nodaemon" xterm off secure
```

Por padrão esta entrada está desativada; para ativá-la modifique o campo 5 de *off* para *on* e reinicie o [init\(8\)](#) usando as instruções em [Seção 17.3.2.2](#). O primeiro campo, o

nome do terminal a ser gerenciado por este programa, é `ttv8`. Isso significa que o **XDM** vai iniciar sua execução no 9º terminal virtual.

### 5.6.3. Configurando o XDM

A pasta com a configuração do **XDM** está localizada em `/usr/X11R6/lib/X11/xdm`. Nesta pasta há vários arquivos usados para modificar o comportamento e a aparência do **XDM**. Tipicamente esses arquivos serão encontrados:

Arquivo	Descrição
<code>Xaccess</code>	Regulamento para autorização de clientes.
<code>Xresources</code>	Valores padrão para recursos do X.
<code>Xservers</code>	Lista de telas locais e remotas para gerenciar.
<code>Xsession</code>	<i>Script</i> de sessão padrão para <i>logins</i> .
<code>Xsetup_*</code>	<i>Script</i> que executa aplicações antes da interface de <i>login</i> .
<code>xdm-config</code>	Configuração global para todas as telas em execução nesta máquina.
<code>xdm-errors</code>	Erros gerados pelo programa servidor.
<code>xdm-pid</code>	O ID do processo do XDM que está em execução.

Nesta pasta também existem alguns *scripts* e programas utilizados para organizar a área de trabalho quando o **XDM** está em execução. O propósito de cada um desses arquivos será brevemente descrito. A sintaxe exata e o uso de todos estes arquivos está descrito em [xdm\(1\)](#).

A configuração padrão é uma janela retangular simples para *login* com o nome do host (*hostname*) exibido no topo com uma fonte grande e os campos ``Login:" e ``Password:" em prontidão logo abaixo. Esse é um bom ponto de partida para trocar o visual e sensação das telas do **XDM**.

#### 5.6.3.1. Xaccess

O protocolo para a conexão com as telas do **XDM** é chamado Protocolo de Conexão com o Gerente de Telas X (X Display Manager Connection Protocol, ou apenas XDMCP). Este arquivo contém o regulamento que controla as conexões via XDMCP vindas de máquinas remotas. Por padrão, ela permite que qualquer cliente se conecte, mas isso é irrelevante a menos que o `xdm-config` seja modificado para esperar por conexões remotas.

### 5.6.3.2. Xresources

Este é um arquivo de valores padrão para aplicações, para o seletor de telas e para a tela de *login*. Este é o lugar onde a aparência do programa de *login* pode ser modificada. O formato é idêntico ao do arquivo de padrões para aplicações descrito na documentação do **XFree86**<sup>TM</sup>.

### 5.6.3.3. Xservers

Esta é a lista de telas remotas selecionáveis das quais o seletor de telas deve prover como escolhas possíveis.

### 5.6.3.4. Xsession

Este é o *script* padrão das sessões que o **XDM** executa após o *login* de um usuário. Normalmente cada usuário terá um *script* para a sessão personalizado no `~/.xsession` que predomina sobre esse *script*.

### 5.6.3.5. Xsetup\_\*

Estes executarão automaticamente antes que o seletor ou a interface de *login* sejam exibidas. Há um *script* para cada tela em uso, cujo nome é `xsetup_` seguido do número da tela local (por exemplo `xsetup_0`). Tipicamente estes *scripts* executarão um ou mais programas no plano de fundo como o `xconsole`.

### 5.6.3.6. xdm-config

Este contém atributos na forma de padrões de aplicações que são aplicados para toda tela gerenciada por esta instalação.

### 5.6.3.7. xdm-errors

Este contém as saídas geradas pelos servidores X os quais o **XDM** está tentando executar. Se uma tela a qual o **XDM** está tentando iniciar congela por alguma razão, este é um bom lugar para procurar por mensagens de erro. Estas mensagens são também escritas no arquivo do usuário chamado `~/.xsession-errors` sempre a cada sessão.

## 5.6.4. Executando um Servidor de Telas em Rede

Para que outros clientes possam conectar-se ao servidor de telas, modifique as regras de controle de acesso e ative a espera por conexões (um *listener*). Por padrão estes são atribuídos a valores conservadores. Para que o **XDM** aguarde conexões deve-se primeiro comentar uma linha no arquivo `xdm-config`:

```
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with xdm
DisplayManager.requestPort: 0
```

que quer dizer: ``AVISO: não se deve esperar por conexões XDMCP ou requisições de um selecionador. Comente esta linha se você quiser que os terminais X sejam gerenciados pelo xdm". Após comentar a linha que segue o aviso, reinicie o **XDM**. Lembre-se que comentários em arquivos de padrões para aplicações começam com um caracter ``!", e não o ``#" ao qual estamos acostumados. Controles mais rígidos para o acesso podem ser convenientes. Observe as entradas de exemplo no arquivo `xaccess`, e consulte a página do manual [xdm\(1\)](#).

### 5.6.5. Substitutos para o XDM

Existem vários programas que substituem o programa **XDM** padrão. Um deles, o **KDM** (fornecido junto com o **KDE**) é descrito adiante neste capítulo. O **KDM** oferece vários avanços visuais e floreios cosméticos, bem com as funções que permitem aos usuários escolher seu gerenciador de janela preferidos no momento do *login*.

---

## 5.7. Ambientes para a área de trabalho

*Contribuição de Valentino Vaschetto.*

Esta seção descreve os diferentes ambientes para a área de trabalho disponíveis para o X no FreeBSD. Um ``ambiente para a área de trabalho" pode significar qualquer coisa desde um simples gerenciador de janelas a uma coleção completa de aplicativos, como o **KDE** ou o **GNOME**.

### 5.7.1. GNOME

#### 5.7.1.1. Sobre o GNOME

O **GNOME** é um ambiente para a área de trabalho amigável que permite aos usuários usar e configurar facilmente seus computadores. O **GNOME** inclui um painel (para iniciar os aplicativos e para a exibição do estado), uma área de trabalho (onde os dados e os aplicativos podem ser colocados), um conjunto de ferramentas e aplicativos padrão, e um conjunto de convenções que facilitam a cooperação e a consistência entre as aplicações. Usuários de outros sistemas operacionais ou ambientes se sentirão em casa usando um poderoso ambiente dirigido por gráficos provido pelo **GNOME**. Mais informações sobre o **GNOME** no FreeBSD podem ser encontrados no sítio da [www FreeBSD GNOME Project](#).

#### 5.7.1.2. Instalando o GNOME

A forma mais fácil de instalar o **GNOME** é através do menu ``Desktop Configuration", que significa ``configuração da área de trabalho", durante o processo de instalação do FreeBSD como descrito em [Seção 2.9.13](#) do Capítulo 2. Ele também pode ser facilmente instalado de um pacote ou da coleção de *ports*:

Para instalar o pacote do **GNOME** a partir da rede, basta entrar:

```
pkg_add -r gnome2
```

Para construir o **GNOME** a partir dos fontes, utilize a árvore de *ports*:

```
cd /usr/ports/x11/gnome2
make install clean
```

Uma vez que o **GNOME** esteja instalado, o servidor X deve ser instruído para iniciar o **GNOME** ao invés do gerenciador de janela padrão. Se um `.xinitrc` personalizado estiver em uso, basta substituir a linha que inicia o gerenciador de janela atual para que, ao invés dele, inicie o `/usr/X11R6/bin/gnome-session`. Se nada de especial foi feito no arquivo de configurações, então é suficiente apenas entrar:

```
% echo "/usr/X11R6/bin/gnome-session" > ~/.xinitrc
```

A seguir, entre com `startx`, e o ambiente da área de trabalho **GNOME** vai ser iniciado.

**Nota:** Se um gerenciador de exibição, como o **XDM**, estiver sendo usado, isso não vai funcionar. Ao invés disso, crie um arquivo executável `.xsession` contendo o mesmo comando. Para fazer isso, edite o arquivo e substitua o comando do gerenciador de janela existente por `/usr/X11R6/bin/gnome-session`:

```
% echo "#!/bin/sh" > ~/.xsession
% echo "/usr/X11R6/bin/gnome-session" >> ~/.xsession
% chmod +x ~/.xsession
```

Outra opção é configurar o gerenciador de exibição para que permita escolher o gerenciador de janela no momento do *login*; a seção [Detalhes do KDE](#) explica como fazer isso para o **kdm**, o gerenciador de exibição do **KDE**.

### 5.7.1.3. Fontes com *anti-alias* no GNOME

A partir da versão 4.0.2, o **XFree86**<sup>TM</sup> suporta *anti-alias* através de sua extensão ```RENDER`". O **Gtk+** 2.0 e posteriores (a caixa de ferramentas usada pelo **GNOME**) pode utilizar essa funcionalidade. A configuração do *anti-alias* está descrita na [Seção 5.5.3](#). Então, com software atual, *anti-alias* é possível dentro da área de trabalho **GNOME**. Apenas vá para Applications (aplicações)->Desktop Preferences (preferências da área de trabalho)->Font (fonte), e escolha entre Best shapes (melhores formas), Best contrast (melhor contraste), ou Subpixel smoothing (LCDs) (suavização de subpixel). Para uma aplicação **Gtk+** que não é parte da área de trabalho **GNOME**, ajuste a variável de ambiente `GDK_USE_XFT` para `1` antes de executar o programa.

## 5.7.2. KDE

### 5.7.2.1. Sobre o KDE

O **KDE** é um ambiente da área de trabalho contemporânea fácil de usar. Algumas das coisas que o **KDE** oferece ao usuário são:

- Uma área de trabalho contemporânea bonita
- Uma área de trabalho que exhibe transparência de rede completa

- Um sistema de ajuda integrado que permite acesso conveniente e consistente para esclarecer o uso da área de trabalho **KDE** e seus aplicativos
- Visual e sensações consistentes de todos os aplicativos do **KDE**
- Menus e barras de ferramentas padronizados, teclas de atalho, esquemas de cores, etc.
- Internacionalização: o **KDE** está disponível em mais de 40 idiomas
- Configuração da área de trabalho baseada em diálogos centralizados consistentemente
- Um grande número de aplicativos **KDE** úteis

O **KDE** tem uma suíte de aplicação para escritórios baseada na tecnologia ``KParts" do **KDE** que consiste em uma planilha, uma aplicação para apresentações, um organizador, um cliente de *news* e mais. O **KDE** também vem com um navegador da *www* chamado **Konqueror**, que representa um sólido competidor para outros navegadores da *www* nos sistemas UNIX®. Mais informações sobre o **KDE** podem ser encontrados no sítio da *www* [KDE](#) . Para informações e recursos específicos do **KDE** para o FreeBSD, consulte o sítio da *www* [FreeBSD-KDE team](#).

### 5.7.2.2. Instalando o KDE

Tal como o **GNOME** ou qualquer outro ambiente para a área de trabalho, a forma mais fácil de instalar o **KDE** é através do menu ``Desktop Configuration" (configuração da área de trabalho) durante o processo de a instalação do FreeBSD como descrito na [Seção 2.9.13](#) do Capítulo 2. Mais uma vez, o software pode ser facilmente instalado de um pacote ou da coleção de *ports*:

Para instalar o pacote do **KDE** a partir da rede, basta entrar:

```
pkg_add -r kde
```

O [pkg\\_add\(1\)](#) vai automaticamente buscar a última versão da aplicação.

Para construir o **KDE** a partir do código fonte, use a árvore de *ports*:

```
cd /usr/ports/x11/kde3
make install clean
```

Após ter instalado o **KDE**, o servidor X precisa ser instruído em como lançar este aplicativo ao invés do gerenciador de janela padrão. Isso pode ser feito editando o arquivo *.xinitrc*:

```
% echo "exec startkde" > ~/.xinitrc
```

Agora, sempre que o sistema de janela X for invocado com *startx*, o **KDE** será a área de trabalho.

Se um gerenciador de exibição como o **xdm** estiver sendo usado, a configuração é um pouco diferente. Ao invés disso, edite o arquivo *.xsession*. Instruções para o **kdm** são descritas adiante neste capítulo.



### 5.7.3. Mais detalhes sobre o KDE

Agora que o **KDE** está instalado no sistema, a maioria das coisas pode ser descoberta através das páginas de ajuda, ou apenas apontando e clicando em vários menus. Os usuários do Windows® ou Mac® vão se sentir quase em casa.

A melhor referência para o **KDE** é a sua documentação on-line. O **KDE** vem com seu próprio navegador da [www](http://www.kde.org), o **Konqueror**, dúzias de aplicativos úteis, e ampla documentação. O resto desta seção discute os itens técnicos que são difíceis de aprender pela exploração aleatória.

#### 5.7.3.1. O gerenciador de exibição do KDE

Um administrador de um sistema multiusuário pode desejar ter uma tela para *login* gráfica para saudar seus usuários. O [xdm](#) pode ser usado, como descrito anteriormente. Entretanto, o **KDE** inclui uma alternativa, o **kdm**, que é concebido para aparentar melhor e incluir mais opções no momento do *login*. Em particular, os usuários podem escolher facilmente (em um menu) qual ambiente para a área de trabalho (**KDE**, **GNOME**, ou outra coisa) será executado após efetuar o *login*.

Para começar, execute o painel de controle do **KDE**, `kcontrol`, como `root`. Geralmente considera-se inseguro executar todo o ambiente do X como `root`. Ao invés disso, execute o gerenciador de janela como um usuário normal, abra uma janela de terminal (como o `xterm` ou o `konsole` do **KDE**), mude para `root` com `su` (o usuário deve estar no grupo `wheel` em `/etc/group` para isso), e então digite `kcontrol`.

Clique no ícone da esquerda marcado como System (sistema), então em Login manager (gerenciador de *login*). À direita há várias opções configuráveis as quais o manual do **KDE** vai explicar em maiores detalhes. Clique em sessions (sessões) à direita. Clique em New type (novo tipo) para acrescentar variados gerenciadores de janela e ambientes de área de trabalho. Estes são apenas rótulos, de forma que eles podem representar **KDE** e **GNOME** ao invés de `startkde` ou `gnome-session`. Inclua um rótulo *failsafe* (modo de segurança).

Experimente com outros menus também, eles são principalmente auto-explicativos. Quando terminar clique o botão Apply (aplicar) na parte de baixo, e saia da central de controle (*control center*).

Para ter certeza que o **kdm** entende o que os rótulos (**KDE**, **GNOME** etc) significam, edite os arquivos usados pelo [xdm](#).

**Nota:** No **KDE 2.2** isso foi modificado: o **kdm** agora usa seus próprios arquivos de configuração. Por favor, verifique a documentação do **KDE 2.2** para detalhes.

Em um terminal, como `root`, edite o arquivo `/usr/X11R6/lib/X11/xdm/Xsession`. Há uma seção no meio como isso:

```

case $# in
1)
 case $1 in
 failsafe)
 exec xterm -geometry 80x24-0-0
 ;;
 esac
esac

```

Poucas linhas precisam ser acrescentadas a esta seção. Assumindo que os rótulos utilizados foram ``KDE" e ``GNOME", use o seguinte: (n.t. Aparentemente há um erro de caixa na inscrição ``kde")

```

case $# in
1)
 case $1 in
 kde)
 exec /usr/local/bin/startkde
 ;;
 GNOME)
 exec /usr/X11R6/bin/gnome-session
 ;;
 failsafe)
 exec xterm -geometry 80x24-0-0
 ;;
 esac
esac

```

Para que o fundo da área de trabalho no momento do *login* no **KDE** seja contemplado, as linhas a seguir precisam ser acrescentadas em `/usr/X11R6/lib/X11/xdm/Xsetup_0:`

```
/usr/local/bin/kdmdesktop
```

Agora, assegure-se de ter o **kdm** listado em `/etc/ttys` para ser iniciado na próxima inicialização. Para fazer isso, basta seguir as instruções da seção anterior em [xdm](#) e substituir as referências anteriores ao programa `/usr/X11R6/bin/xdm` por `/usr/local/bin/kdm`.

### 5.7.3.2. Fontes com *anti-alias*

A partir da versão 4.0.2, o **XFree86** suporta *anti-alias* através de sua extensão ``RENDER", e desde a versão 2.3, o Qt (a caixa de ferramentas usada pelo **KDE**) pode utilizar essa funcionalidade. Essa configuração está descrita na [Seção 5.5.3](#) sobre o *anti-alias* em fontes do X11. Então, com software atualizado, é possível usar *anti-alias* em uma área de trabalho **KDE**. Basta ir ao menu do KDE: vá para Preferences (preferências)->Look and Feel (visual e sensação)->Fonts (fontes), e clique na caixa de checagem Use Anti-Aliasing for Fonts and Icons (usar anti-alias para fontes e ícones). Para uma aplicação Qt a qual não é parte do **KDE**, a variável de ambiente `QT_XFT` precisa ser ajustada para *true* antes de iniciar o programa.

## 5.7.4. XFce

### 5.7.4.1. Sobre o XFce

O **XFce** é um ambiente da área de trabalho baseado na caixa de ferramentas GTK usado pelo **GNOME**, mas é muito mais leve e indicado para quem quer uma área de trabalho simples e eficiente, e ainda fácil de usar e configurar. Visualmente, ela se parece muito com o **CDE**, encontrado em sistemas UNIX comerciais. Alguns recursos do **XFce** são:

- Uma área de trabalho simples e fácil de lidar
- Totalmente configurável via mouse, com arrastar e soltar (*drag and drop*), etc
- O painel principal é muito similar ao **CDE**, com menus, *applets* e lançadores de aplicativos
- Gerenciador de janela integrado, gerenciador de arquivo, gerenciador de som, um módulo de compatibilidade com o **GNOME**, e outras coisas
- Tematizável (já que usa o GTK)
- Rápido, leve e eficiente: ideal para máquinas mais antigas/lentas ou máquinas com limitações de memória

Mais informações sobre o **XFce** podem ser encontradas no [sítio da www XFce](#) .

### 5.7.4.2. Instalando o XFce

Existe um pacote binário do **XFce** (quando este texto foi escrito). Para instalar, basta entrar:

```
pkg_add -r xfce
```

Alternativamente, para construir do código fonte, use a coleção de *ports*:

```
cd /usr/ports/x11-wm/xfce
make install clean
```

Agora, diga ao servidor X para lançar o **XFce** na próxima vez que o X for iniciado. Basta entrar com isso:

```
% echo "/usr/X11R6/bin/startxfce" > ~/.xinitrc
```

A próxima vez que o X for iniciado, o **XFce** vai ser a área de trabalho. Como antes, se um gerenciador de exibição como o *xdm* estiver sendo usado, crie um *.xsession*, como descrito na seção sobre o [GNOME](#), mas com o comando `/usr/X11R6/bin/startxfce`; ou configure o gerenciador de exibição para permitir a escolha da área de trabalho no momento do *login*, como foi explicado na seção sobre o [kdm](#).

## II. Administração do Sistema

Os capítulos restantes do handbook, abordam todos os aspectos da administração de um sistema FreeBSD. Cada capítulo inicia descrevendo o que você irá aprender como resultado da sua leitura, e também detalha o que é esperado que você conheça antes de iniciar a leitura.

Estes capítulos foram projetados para serem lidos quando você precisar de informação. Você não tem que lê-los em nenhuma ordem particular, nem você precisa ler todos antes de começar a utilizar o FreeBSD.

### Índice

6. [Configuração e Ajuste](#)
  7. [O processo de inicialização do FreeBSD](#)
  8. [Usuários e Administração Básica de contas](#)
  9. [Configurando o \*kernel\* do FreeBSD](#)
  10. [Segurança](#)
  11. [Impressão](#)
  12. [Armazenamento](#)
  13. [O Gerenciador de Volumes Vinum](#)
  14. [Localização - I18N/L10N Uso e Configuração](#)
  15. [Aplicações de \*Desktop\*](#)
  16. [Multimedia](#)
  17. [Comunicações Seriais](#)
  18. [PPP e SLIP](#)
  19. [Uso Avançado de Redes](#)
  20. [Correio eletrônico](#)
  21. [A Versão Mais Recente](#)
  22. [Compatibilidade Binária Linux](#)
- 

## Capítulo 6. Configuração e Ajuste

### Índice

- 6.1. [Sinótese](#)
- 6.2. [Configuração Inicial](#)
- 6.3. [Configuração Principal](#)
- 6.4. [Configuração de Aplicações](#)
- 6.5. [Inicializando Serviços](#)
- 6.6. [Configurando o Utilitário \*cron\*](#)
- 6.7. [Usando \*rc\* no FreeBSD 5.X](#)
- 6.8. [Configurando Interfaces de Rede](#)
- 6.9. [Servidores Virtuais](#)
- 6.10. [Arquivos de Configuração](#)
- 6.11. [Ajustando com o \*sysctl\*](#)
- 6.12. [Ajustando Discos](#)
- 6.13. [Ajustando Limites do Kernel](#)
- 6.14. [Adicionando Espaço de Swap](#)

## 6.15. [Gerenciamento de Energia e Recursos](#)

*Escrito por Chern Lee. Baseado em um tutorial escrito por Mike Smith. Também baseado em [tuning\(7\)](#) escrito por Matt Dillon.*

# 6.1. Sinótese

Um dos aspectos mais importantes do FreeBSD é a configuração do sistema. A configuração correta ajudará a prevenir dores de cabeça durante atualizações futuras. Este capítulo explicará muito do processo de configuração do FreeBSD, incluindo alguns parâmetros que podem ser configurados para ajustar um sistema FreeBSD.

Após ler este capítulo, você saberá:

- Como trabalhar eficientemente com sistemas de arquivos e partições de swap.
- Noções de configuração dos sistemas de inicialização `rc.conf` e `/usr/local/etc/rc.d`.
- Como configurar e testar uma placa de rede.
- Como configurar servidores virtuais em seus dispositivos de rede.
- Como usar os diversos arquivos de configuração em `/etc`.
- Como ajustar o FreeBSD usando as variáveis do `sysctl`.
- Como ajustar a performance de disco e modificar limitações do kernel.

Antes de ler este capítulo, você deverá:

- Compreender o básico de UNIX® e do FreeBSD ([Capítulo 3](#)).
- Estar familiarizado em manter os fontes do FreeBSD atualizados ([Capítulo 21](#)), e possuir noções de configuração/compilação do kernel. ([Capítulo 9](#)).

---

# 6.2. Configuração Inicial

## 6.2.1. Estrutura de Particionamento

### 6.2.1.1. Partições Base

Ao configurar sistemas de arquivo com [disklabel\(8\)](#) ou [sysinstall\(8\)](#), lembre-se que discos rígidos transferem dados mais rapidamente das trilhas mais externas para as internas. Desta forma, sistemas de arquivos menores e mais acessados devem estar mais próximos da parte de fora do disco, enquanto partições maiores como `/usr` devem estar mais próximas à parte de dentro. É uma boa idéia criar partições nesta ordem: `raíz`, `swap`, `/var`, `/usr`.

O tamanho de `/var` reflete na intenção de uso da máquina. `/var` é usado para abrigar caixas de correio, arquivos de log e filas de impressão. Caixas postais e arquivos de log podem crescer em tamanhos não esperados dependendo da quantidade de usuários e por

quanto tempo os arquivos de log são mantidos. Muitos usuários nunca precisarão de um gigabyte, mas lembre-se que `/var/tmp` deve ser grande o bastante para abrigar pacotes.

A partição `/usr` abriga a maioria dos arquivos necessários ao sistema, a coleção [ports\(7\)](#) (recomendada) e o código fonte (opcional). Ambos são opcionais no momento da instalação. Pelo menos 2 gigabytes são recomendados para esta partição.

Ao selecionar tamanhos de partições, tenha em mente os requisitos de tamanho. Ficar sem espaço em uma partição enquanto outra é pouco usada pode ser um pouco chato.

**Nota:** Alguns usuários perceberam que o dimensionador padrão *Auto-defaults* do [sysinstall\(8\)](#) algumas vezes selecionam partições `/var` e `/` em tamanhos menores que o necessário. Particione de maneira inteligente e generosa.

### 6.2.1.2. Partição de Swap

Como regra geral, a partição de swap deve ser mais ou menos o dobro do tamanho da memória do sistema (RAM). Por exemplo, se uma máquina possui 128 megabytes de memória, o arquivo de swap deveria ser de 256 megabytes. Sistemas com pouca memória podem obter melhor performance com mais swap. Menos que 256 megabytes de memória não é recomendado e uma expansão deve ser considerada. Os algoritmos de paginação da memória virtual (VM) são sintonizados para obter melhor desempenho quando a partição de swap é pelo menos duas vezes o tamanho da memória. Configurar um swap muito pequeno pode levar à ineficiência no código de procura de página da memória virtual e pode criar problemas se mais memória for adicionada.

Em sistemas maiores com múltiplos discos SCSI (ou múltiplos discos IDE operando em controladoras diferentes), é recomendado que o swap seja configurado em cada disco (para até quatro discos). As partições de swap devem ser aproximadamente do mesmo tamanho. O kernel pode manipular tamanhos arbitrários mas estruturas de dados internas escalam à até 4 vezes para a maior partição. Manter as partições de swap com aproximadamente o mesmo tamanho permitirá ao *kernel* otimizar a distribuição do espaço de swap pelas partições. Não há problemas em ter grandes partições de swap, mesmo que o swap não seja muito usado. Isto pode facilitar a recuperação de dados de um programa antes de ser forçado a reinicializar a máquina.

### 6.2.1.3. Porque Particionar?

Muitos usuários pensam que uma partição grande e única é suficiente mas, por várias razões, isto seria uma má idéia. Primeiro, cada partição possui características operacionais diferentes e separá-las permite ao sistema de arquivos uma sintonia de acordo. Por exemplo, as partições `root` e `/usr` recebem mais acessos de leitura, e pouca escrita. Enquanto que muita leitura e escrita pode ocorrer em `/var` e `/var/tmp`.

Através de um particionamento adequado, a fragmentação introduzida em partições pequenas com muitos acessos de escrita não prejudicará as partições com mais acessos de leitura. Manter as partições com mais acesso de escrito mais próximas à borda do disco proporcionará melhora na performance de I/O nas partições onde isto ocorre com mais frequência. Enquanto a performance de I/O pode ser necessária em partições maiores, colocá-las mais na parte externa do disco não promoverá uma melhora de

performance em relação à mover o `/var` para a borda do disco. Finalmente, existem preocupações com a segurança. Uma partição raiz menor, mais pura, que tem mais acessos de leitura, tem uma chance maior de sobreviver à um *crash*.

---

## 6.3. Configuração Principal

O local principal para informações de configuração de sistema está dentro de `/etc/rc.conf`. Este arquivo contém uma grande quantidade de informações de configuração, essencialmente usadas na inicialização do sistema. Seu nome implica diretamente nisto; é a informação de configuração para os arquivos `rc*`.

Um administrador deve inserir entradas no arquivo `rc.conf` para sobrepor as configurações padrão em `/etc/defaults/rc.conf`. Os arquivos padrão não devem ser copiados para `/etc` - pois contém valores padrões, não exemplos. Todas mudanças de sistema específicas devem ser feitas diretamente no arquivo `rc.conf`

Uma variedade de estratégias podem ser aplicadas em aplicações em *cluster* para separar configurações globais de sistema das configurações específicas com o intuito de diminuir a carga de administração. O procedimento adotado é o de colocar configurações globais em outro arquivo, como `/etc/rc.conf.site`, e então incluir este arquivo em `/etc/rc.conf`, que possuirá apenas informações específicas de sistema.

Como o `rc.conf` é lido pelo [sh\(1\)](#) é simples fazer isto. Por exemplo:

- `rc.conf`:
- `. rc.conf.site`
- `hostname="node15.example.com"`
- `network_interfaces="fxp0 lo0"`
- `ifconfig_fxp0="inet 10.1.1.1"`
- `rc.conf.site`:
- `defaultrouter="10.1.1.254"`
- `saver="daemon"`
- `blanktime="100"`

O arquivo `rc.conf.site` pode ser distribuído para todos os sistemas usando-se `rsync` ou um programa similar, enquanto que o arquivo `rc.conf` permanece único.

Atualizar o sistema usando o [sysinstall\(8\)](#) ou `make world` não sobrescreverá o arquivo `rc.conf`, assim a configuração de sistema não será perdida.

## 6.4. Configuração de Aplicações

Tipicamente, aplicações instaladas possuem seus próprios arquivos de configuração, com sua própria sintaxe, etc. É importante que estes arquivos sejam mantidos separados do sistema base, para que eles possam ser facilmente localizados e gerenciados pelas ferramentas de gerenciamento de pacotes.

Tipicamente, estes arquivos são instalados em `/usr/local/etc`. No caso em que uma aplicação possui um grande número de arquivos de configuração, um subdiretório será criado para abrigá-los.

Normalmente, quando um *port* ou pacote é instalado, arquivos de configuração de exemplo também são instalados. Estes arquivos normalmente são identificados como tendo o sufixo `.default`. Se não existirem arquivos de configuração para a aplicação, eles serão criados através da cópia dos arquivos `.default`.

Por exemplo, considere o conteúdo do diretório `/usr/local/etc/apache`:

```
-rw-r--r-- 1 root wheel 2184 May 20 1998 access.conf
-rw-r--r-- 1 root wheel 2184 May 20 1998 access.conf.default
-rw-r--r-- 1 root wheel 9555 May 20 1998 httpd.conf
-rw-r--r-- 1 root wheel 9555 May 20 1998 httpd.conf.default
-rw-r--r-- 1 root wheel 12205 May 20 1998 magic
-rw-r--r-- 1 root wheel 12205 May 20 1998 magic.default
-rw-r--r-- 1 root wheel 2700 May 20 1998 mime.types
-rw-r--r-- 1 root wheel 2700 May 20 1998 mime.types.default
-rw-r--r-- 1 root wheel 7980 May 20 1998 srm.conf
-rw-r--r-- 1 root wheel 7933 May 20 1998 srm.conf.default
```

O tamanho dos arquivos mostram que apenas o arquivo `srm.conf` foi modificado. Uma atualização posterior do *port* do **Apache** não sobrescreverá este arquivo modificado.

---

## 6.5. Inicializando Serviços

Hospedar vários serviços é comum para um sistema. Eles podem ser inicializados de formas diferentes, cada um com suas vantagens.

Aplicativos instalados através do *port* ou da coleção de pacotes frequentemente colocará um *script* em `/usr/local/etc/rc.d`, que será invocado na inicialização do sistema com o argumento *start*, e no desligamento do sistema com o argumento *stop*. Esta é a maneira recomendada de inicializar serviços globais de sistema que são executados sob o usuário `root`, ou que esperam ser inicializados como `root`. Estes *scripts* são registrados como parte da instalação do pacote, e serão removidos quando o pacote for removido.

Um *script* genérico de inicialização em `/usr/local/etc/rc.d` seria parecido com:

```
#!/bin/sh
echo -n ' FooBar '
```



```

case "$1" in
start)
 /usr/local/bin/foobar
 ;;
stop)
 kill -9 `cat /var/run/foobar.pid`
 ;;
*)
 echo "Usage: `basename $0` {start|stop}" >&2
 exit 64
 ;;
esac

exit 0

```

O *script* de inicialização do FreeBSD procurará em `/usr/local/etc/rc.d` por *scripts* que possuem a extensão `.sh` e são executáveis pelo `root`. Estes *scripts* encontrados são chamados com o argumento *start* na inicialização, e *stop* no desligamento, para permitir que executem suas funções. Sendo assim, se você desejar que o *script* de exemplo acima seja selecionado e executado no momento de inicialização do sistema, você deve gravá-lo em um arquivo, como por exemplo `FooBar.sh` em `/usr/local/etc/rc.d` e ter certeza de que ele é executável. Você pode fazer com que um *script* executor de comandos seja executável com o [chmod\(1\)](#) como mostrado abaixo:

```
chmod 755 FooBar.sh
```

Alguns serviços esperam ser invocados pelo [inetd\(8\)](#) quando uma conexão é recebida em uma porta disponível. É comum para servidores de correio (POP e IMAP, etc). Estes serviços são habilitados editando-se o arquivo `/etc/inetd.conf`. Veja [inetd\(8\)](#) para detalhes de como editar este arquivo.

Alguns serviços de sistema adicionais podem não possuir entradas no arquivo `/etc/rc.conf`. Estes serviços são tradicionalmente habilitados incluindo-se comandos no arquivo `/etc/rc.local`. Como no FreeBSD 3.1 não existe o arquivo `/etc/rc.local`; mas se for criado pelo administrador ele será executado pelo sistema. Note que o arquivo `rc.local` é geralmente considerado como um local de último recurso; se existir um lugar melhor para inicializar o serviço, coloque o *script* neste lugar melhor.

**Nota:** Não coloque qualquer comando no arquivo `/etc/rc.conf`. Para inicializar *daemons*, ou executar qualquer comando durante a inicialização, coloque um *script* em `/usr/local/etc/rc.d`.

Também é possível usar o *daemon* [cron\(8\)](#) para inicializar serviços de sistema. Este método tem algumas vantagens, não é para menos, pois o [cron\(8\)](#) executa estes processos sob o proprietário do `crontab`, e os serviços podem ser inicializados e mantidos por usuários não-`root`.

Isto tira vantagem de uma característica do [cron\(8\)](#): a especificação de tempo pode ser trocada por `@reboot`, que fará com que o trabalho seja executado quando o [cron\(8\)](#) for inicializado logo após a inicialização do sistema.

## 6.6. Configurando o Utilitário `cron`

Contribuição de Tom Rhodes.

Uma das ferramentas mais úteis no FreeBSD é o [cron\(8\)](#). O utilitário `cron` roda em segundo plano e constantemente verifica o arquivo `/etc/crontab`. Ele também verifica o diretório `/var/cron/tabs` na procura novos arquivos do tipo `crontab`. Estes arquivos `crontab` armazenam informações específicas sobre funções que o `cron` é designado a desempenhar em algumas ocasiões.

Vamos dar uma olhada no arquivo `/etc/crontab`:

```
/etc/crontab - root's crontab for FreeBSD
#
$FreeBSD: src/etc/crontab,v 1.32 2002/11/22 16:13:39 tom Exp $
❶
#
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin ❷
HOME=/var/log
#
#
#minute hour mday month wday who command ❸
#
*/5 * * * * root /usr/libexec/atrun ❹
```

❶

Como muitos arquivos de configuração do FreeBSD, o caracter `#` representa um comentário. Um comentário pode ser colocado em um arquivo como um lembrete de como e porque uma ação desejada é executada. Os comentários não podem estar na mesma linha que o comando senão serão interpretados como parte do comando; eles devem estar em uma nova linha. Linhas em branco são ignoradas.

❷

Primeiramente, o ambiente deve ser definido. Os caracteres (`=`) são usados para definir qualquer configuração de ambiente, como neste exemplo que são usados para as opções `SHELL`, `PATH`, e `HOME`. Se a linha do shell for omitida, o `cron` usará o padrão, que é o `sh`. Se a variável `PATH` for omitida, nenhum padrão será usado e o caminho para os arquivos deverá ser absoluto. Se o `HOME` for omitido, o `cron` usará o diretório base do usuário que o invocou.

❸

Esta linha define um total de sete campos. Aqui são listados os valores *minute*, *hour*, *mday*, *month*, *wday*, *who*, e *command*. Estes são praticamente autoexplicativos. Minute é o tempo em minutos em que o comando será executado. Hours é similar à opção Minute, em horas. Month é similar à Minute e Hour, designando o mês. As opções *mday* vem de dia da semana. Todos estes campos devem ter valores

numéricos, e obedecer ao relógio do tipo 24h. O campo ```who``` é especial, e existe apenas no arquivo `/etc/crontab`. Este arquivo especifica sob que usuário o comando deve ser executado. Quando um usuário instala seu arquivo `crontab`, ele ou ela não terá esta opção. Finalmente, opções do comando são listadas. Este é o último campo, sendo assim, naturalmente deve designar o comando a ser executado.

#### 4

Esta última linha definirá os valores discutidos acima. Note aqui que tempos a listagem `*/5`, seguida por vários caracteres `*`. Estes caracteres `*` significam ```primeiro-último```, e podem ser interpretados como *qualquer* tempo. Então, julgando por esta linha, fica claro que o comando `atrun` será invocado pelo `root` a cada cinco minutos não importando o dia ou mês. Para mais informações sobre o `atrun`, veja a página de manual [atrun\(8\)](#).

Comandos podem ter qualquer número de parâmetros passados à eles mesmos; entretando, comandos que se estendem à várias linhas devem ser quebrados com o caracter contrabarra ```\```.

Esta é a configuração básica para todo arquivo `crontab`, embora exista algo diferente a respeito deste. O campo número seis, onde é especificado o nome de usuário, que existe somente no arquivo `/etc/crontab`. Este arquivo deve ser omitido para arquivos `crontab` individuais de usuários.

### 6.6.1. Instalando a Crontab

Para instalar sua recém-escrita `crontab`, basta usar o utilitário `crontab`. O uso mais comum é:

```
crontab arquivo-crontab
```

Existe também uma opção para listar os arquivos `crontab` instalados, basta passar a opção `-l` para o `crontab` e observar a saída do comando.

Usuários que desejem começar seu próprio arquivo `crontab` do zero, sem usar um exemplo, podem usar `crontab -e`. Este comando invocará o editor com um arquivo vazio. Quando o arquivo for gravado será automaticamente instalado pelo comando `crontab`.

## 6.7. Usando rc no FreeBSD 5.X

*Contribuição de Tom Rhodes.*

O FreeBSD recentemente integrou o sistema de inicialização `rc.d` do NetBSD. Os usuários podem notar os arquivos no diretório `/etc/rc.d`. Muitos destes arquivos são para serviços básicos que podem ser controlados com os parâmetros `start`, `stop`, e `restart`. Por exemplo, o [sshd\(8\)](#) pode ser reinicializado com o seguinte comando:

```
/etc/rc.d/sshd restart
```

Este procedimento é similar para outros serviços. Claro, serviços que normalmente são inicializados como especificado em [rc.conf\(5\)](#). Por exemplo, habilitar o *daemon Network Address Translation* na inicialização basta adicionar a seguinte linha em `/etc/rc.conf`:

```
natd_enable="YES"
```

Se a linha `natd_enable="NO"` já existe, então simplesmente mude a opção `NO` para `YES`. Os *scripts rc* automaticamente carregarão outros serviços dependentes durante a próxima inicialização, como descrito abaixo.

Uma vez que o sistema `rc.d` foi feito primariamente para iniciar/parar serviços na ocasião de inicialização/desligamento do sistema; os parâmetros `start`, `stop` e `restart` somente cumprirão suas funções se as variáveis apropriadas estiverem configuradas no `/etc/rc.conf`. Por exemplo, o comando `sshd restart` acima só funcionará se `sshd_enable` estiver configurado para `YES` no `/etc/rc.conf`. Para iniciar (`start`), parar (`stop`) ou reiniciar (`restart`) um serviço, independente das configurações em `/etc/rc.conf`, os comandos devem ter o prefixo ```force"`. Por exemplo, para reiniciar o `sshd` independente da configuração atual no `/etc/rc.conf`, execute o seguinte comando: :

```
/etc/rc.d/sshd forcerestart
```

É fácil verificar se um serviço está habilitado no `/etc/rc.conf`, basta executar o *script rc.d* com o parâmetro `rcvar`. Desta forma, o administrador pode verificar se o `sshd` está de fato habilitado no `/etc/rc.conf`:

```
/etc/rc.d/sshd rcvar
sshd
$sshd_enable=YES
```

**Nota:** A segunda linha (`# sshd`) é a saída do comando `sshd`; não um console `root`.

Para determinar se o serviço está no ar, use o parâmetro `status`. Por exemplo, para verificar se o `sshd` está no ar:

```
/etc/rc.d/sshd status
sshd is running as pid 433.
```

Também é possível recarregar (*reload*) um serviço. Este parâmetro tentará enviar um sinal para um serviço em particular, forçando uma recarga dos arquivos de configuração. Em muitos casos isto significa enviar um sinal *SIGHUP* para o serviço.

A estrutura do **rcNG** não é apenas usada para serviços de rede, também contribui em grande parte para a inicialização do sistema. Por exemplo, considere o arquivo `bgfsck`. Quando este script for executado imprimirá a seguinte mensagem:

```
Starting background file system checks in 60 seconds.
```

Portanto, este arquivo é usado para checagem do sistema de arquivos em segundo plano, que é feita durante a inicialização.

Muitos serviços de sistema dependem de outros serviços para funcionar corretamente. Por exemplo, NIS e outros serviços baseados em RPC podem falhar durante a inicialização até que o serviço `rpcbind` (portmapper) tenha sido iniciado. Para resolver esta questão, informações sobre dependência e outros meta-dados estão incluídos no topo de cada *script* de inicialização. O programa [rcorder\(8\)](#) é usado para analisar estes comentários durante a inicialização do sistema para determinar a ordem na qual os serviços de sistema devem ser invocados para satisfazer as dependências. As seguintes palavras podem ser incluídas no topo de cada arquivo de *script* de inicialização:

- **PROVIDE:** Especifica os serviços que este arquivo oferece.
- **REQUIRE:** Lista os serviços que são necessários para este serviço. Este arquivo será executado *depois* dos serviços especificados.
- **BEFORE:** Lista os serviços que dependem deste serviço. Este serviço será executado *antes* dos serviços especificados.
- **KEYWORD:** FreeBSD ou NetBSD. Isto é usado para características dependentes de \*BSD.

Usando este método, um administrador pode controlar facilmente os serviços de sistema sem o stress dos ``runlevels'' como em outros sistemas operacionais UNIX®.

Informações adicionais sobre sistema `rc` no FreeBSD 5.X podem ser encontradas em [rc\(8\)](#) e [rc.subr\(8\)](#).

## 6.8. Configurando Interfaces de Rede

*Contribuição de Marc Fonvieille.*

Nos dias de hoje é difícil pensar em um computador sem conexão de rede. Adicionar e configurar uma placa de rede é uma tarefa comum para qualquer administrador FreeBSD.

### 6.8.1. Encontrando o Driver Correto

Antes de começar, você deve saber o modelo da placa que você possui, o chip que ela usa, se é PCI ou ISA. FreeBSD tem suporte à uma grande variedade de placas PCI e

ISA. Verifique a Lista de Compatibilidade de Hardware da sua versão e veja se a sua placa é suportada.

Uma vez que a placa possui suporte, você precisa determinar o *driver* para ela. O arquivo `/usr/src/sys/i386/conf/LINT` te dará uma lista de drivers de placa de rede com alguma informação sobre os *chipsets* suportados. Se você estiver com dúvidas sobre qual *driver* é o correto, leia a página de manual do *driver*. O manual te dará informações sobre os dispositivos suportados e até sobre problemas que podem ocorrer.

Se você possui uma placa comum, provavelmente você não precisará procurar tanto por um *driver*. *Drivers* para as placas de rede mais comuns estão presentes no arquivo de *kernel* `GENERIC`, então sua placa deve aparecer durante a inicialização, tipo:

```
dc0: <82c169 PNIC 10/100BaseTX> port 0xa000-0xa0ff mem 0xd3800000-
0xd38
000ff irq 15 at device 11.0 on pci0
dc0: Ethernet address: 00:a0:cc:da:da:da
miibus0: <MII bus> on dc0
ukphy0: <Generic IEEE 802.3u media interface> on miibus0
ukphy0: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
dc1: <82c169 PNIC 10/100BaseTX> port 0x9800-0x98ff mem 0xd3000000-
0xd30
000ff irq 11 at device 12.0 on pci0
dc1: Ethernet address: 00:a0:cc:da:da:db
miibus1: <MII bus> on dc1
ukphy1: <Generic IEEE 802.3u media interface> on miibus1
ukphy1: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
```

Neste exemplo, note que duas placas usam o driver [dc\(4\)](#), que já está presente no sistema.

Para usar a sua placa de rede você precisa carregar o driver apropriado. Isto pode ser feito de duas maneiras. A mais fácil é simplesmente carregar um módulo de *kernel* para a sua placa com o [kldload\(8\)](#). Um módulo não é disponível para todos os drivers de placa de rede (placas ISA que usam [ed\(4\)](#), por exemplo). Alternativamente, você pode compilar estaticamente o suporte para a sua placa no *kernel*. Verifique o arquivo `/usr/src/sys/i386/conf/LINT` e a página de manual do *driver* para saber o que adicionar ao seu arquivo de configuração do *kernel*. Para maiores informações de como recompilar seu *kernel* por favor veja o [Capítulo 9](#). Se a sua placa foi detectada pelo seu *kernel* (`GENERIC`) durante a inicialização você não precisa construir um novo *kernel*.

## 6.8.2. Configurando a Placa de Rede

Uma vez que o *driver* correto da placa de rede está carregado, a placa precisa ser configurada. Como muitas outras coisas, a placa de rede pode ter sido configurada durante a instalação pelo `sysinstall`.

Para mostrar a configuração das placas de rede do seu sistema, entre com o seguinte comando:

```
% ifconfig
dc0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
 inet 192.168.1.3 netmask 0xfffff00 broadcast 192.168.1.255
```

```

ether 00:a0:cc:da:da:da
media: Ethernet autoselect (100baseTX <full-duplex>)
status: active
dc1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
inet 10.0.0.1 netmask 0xffffffff broadcast 10.0.0.255
ether 00:a0:cc:da:da:db
media: Ethernet 10baseT/UTP
status: no carrier
lp0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
inet 127.0.0.1 netmask 0xff000000
tun0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500

```

**Nota:** Versões antigas do FreeBSD podem requerer o parâmetro *-a* seguido de [ifconfig\(8\)](#), para mais detalhes sobre a sintaxe correta do [ifconfig\(8\)](#) por favor leia a página de manual. Note que também existem entradas referentes à IPv6 (*inet6* etc.) que foram omitidas neste exemplo.

Neste exemplo, os seguintes dispositivos foram mostrados:

- dc0: A primeira interface Ethernet
- dc1: A segunda interface Ethernet
- lp0: A interface de porta paralela
- lo0: O dispositivo *loopback*
- tun0: O dispositivo *tunnel* usado para **ppp**

O FreeBSD usa o nome do driver seguido pela ordem na qual cada placa é detectada na inicialização do *kernel* para dar nome à placa. Por exemplo, *sis2* pode ser a terceira placa de rede no sistema usando o driver [sis\(4\)](#).

Neste exemplo, o dispositivo dc0 está no ar. Os indicadores chave são:

1. *UP* significa que a placa está configurada e pronta para uso.
2. A placa tem um endereço Internet (*inet*) (neste caso 192.168.1.3).
3. Tem uma máscara de rede válida (*netmask*; 0xffffffff00 que é o mesmo que 255.255.255.0).
4. Possui um endereço de *broadcast* válido (neste caso, 192.168.1.255).
5. Possui um endereço MAC para a placa (*ether*) is 00:a0:cc:da:da:da
6. A seleção física de mídia está no modo automático (*media: Ethernet autoselect (100baseTX <full-duplex>)*). Podemos ver que o dispositivo dc1 foi configurado para *10baseT/UTP*. Para mais informações a respeito de tipos de mídia para um determinado *driver*, por favor verifique a respectiva página de manual.
7. O estado da conexão (*status*) é *active*, ou seja, a conexão foi detectada. Para dc1, vemos *status: no carrier*. Isto é normal quando o cabo não está conectado à placa.

Se a saída do [ifconfig\(8\)](#) mostrou algo parecido com

```

dc0: flags=8843<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
ether 00:a0:cc:da:da:da

```

pode indicar que a placa não foi configurada.

Para configurar sua placa, você precisa de privilégios de `root`. A configuração da placa de rede pode ser feita a partir da linha de comando com o [ifconfig\(8\)](#) mas você vai precisar refazer a configuração após cada inicialização. O arquivo `/etc/rc.conf` é o lugar para adicionar a configuração da placa de rede.

Abra o arquivo `/etc/rc.conf` no seu editor preferido. Você precisa adicionar uma linha para cada placa de rede presente no sistema, por exemplo, em nosso caso, adicionamos estas duas linhas:

```
ifconfig_dc0="inet 192.168.1.3 netmask 255.255.255.0"
ifconfig_dc1="inet 10.0.0.1 netmask 255.255.255.0 media 10baseT/UTP"
```

Você precisa substituir `dc0`, `dc1`, e assim sucessivamente, com nomes de dispositivos corretos para suas placas, e os endereços apropriados. Você deve ler as páginas de manual do *driver* e [ifconfig\(8\)](#) para mais detalhes a respeito das opções permitidas e também [rc.conf\(5\)](#) para mais informações sobre a sintaxe em `/etc/rc.conf`.

Se você configurou a rede durante a instalação, algumas linhas de placa(s) de rede podem já estar presentes. Faça uma verificação dupla no arquivo `/etc/rc.conf` antes de adicionar qualquer linha.

Você também terá que editar o arquivo `/etc/hosts` para adicionar nomes e endereços IP das máquinas da Rede Local, se estes nomes ainda não estiverem lá. Para mais informações veja [hosts\(5\)](#) e `/usr/share/examples/etc/hosts`.

### 6.8.3. Testes e Resolução de Problemas

Uma vez feitas as configurações necessárias no `/etc/rc.conf`, você deve reinicializar seu sistema. Isto fará com que as mudanças em sua(s) placa(s) de rede sejam aplicadas, e verifique se seu sistema reinicializa sem erros de configuração.

Após reinicializar o sistema, você deve testar as interfaces de rede.

#### 6.8.3.1. Testando a Placa Ethernet

Para verificar se uma placa Ethernet foi configurada corretamente, você precisa tentar duas coisas. Primeiro, execute o comando `ping` para a sua própria interface, e então faça o mesmo para outra máquina na sua Rede Local.

Primeiro teste para a interface local:

```
% ping -c5 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=0.082 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.074 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.108 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.076 ms

--- 192.168.1.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.074/0.083/0.108/0.013 ms
```



Agora temos que usar o `ping` para outra máquina na Rede Local:

```
% ping -c5 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=0.726 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.766 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.700 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.747 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.704 ms

--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.700/0.729/0.766/0.025 ms
```

Você também pode usar o nome da máquina ao invés de `192.168.1.2` se você configurou o arquivo `/etc/hosts`.

### 6.8.3.2. Resolução de Problemas

Resolução de problemas de configuração de *hardware* e *software* sempre é doloroso, e uma dor que pode ser aliviada através da checagem de coisas simples antes de qualquer coisa. Seu cabo de rede está conectado? Você configurou os serviços de rede corretamente? Você configurou o *firewall* corretamente? A placa de rede que você está usando é suportada pelo FreeBSD? Sempre verifique as notas de hardware antes de enviar um relatório de falha. Atualize sua versão do FreeBSD para a última versão *STABLE*. Verifique os arquivos das listas de discussão, ou procure na Internet.

Se a placa funcionar, e a performance estiver ruim, vale dar uma lida na página de manual [tuning\(7\)](#). Você também pode verificar configurações de rede incorretas, que podem provocar lentidão nas conexões.

Alguns usuários passam por problemas de um ou dois "device timeouts", que é normal em algumas placas. Se eles continuarem, ou começar a incomodar, você pode querer ter certeza de que o dispositivo não está conflitando com outro dispositivo. Faça uma verificação dupla das conexões de cabeamento. Talvez você apenas precise de outra placa.

Algumas vezes, os usuários percebem pequenos erros "watchdog timeout". A primeira coisa a ser feita é checar o cabo da conexão. Muitas placas requerem um *slot* PCI que suporte gerenciamento de barramento. Em algumas placas-mãe antigas, apenas um *slot* permite isso (geralmente o *slot* 0). Verifique a documentação da placa de rede e da placa-mãe para saber se isto pode ser um problema.

A mensagem "No route to host" acontece quando o sistema não é capaz de rotear um pacote para a máquina de destino. Isto pode acontecer se nenhuma rota padrão for especificada, ou se um cabo estiver desconectado. Verifique a saída do comando `netstat -rn` e tenha certeza de que exista uma rota válida para a máquina que você está tentando alcançar. Se não existir, leia o [Capítulo 19](#).

As mensagens de erro "ping: sendto: Permission denied" geralmente são provocadas por um *firewall* desconfigurado. Se o `ipfw` estiver habilitado no *kernel* mas

não existirem regras definidas, então a política padrão é de negar todo o tráfego, até mesmo as requisições de ping! Leia a [Seção 10.8](#) para maiores informações.

Algumas vezes a performance da placa é ruim, ou abaixo do normal. Nestes casos é melhor configurar o modo de seleção de mídia de *autoselect* para a velocidade correta. Enquanto isto pode funcionar para grande parte dos equipamentos, talvez não resolva o problema de todos. Novamente, verifique toda a configuração de rede, e leia a página de manual [tuning\(7\)](#).

---

## 6.9. Servidores Virtuais

Um uso muito comum do FreeBSD é a hospedagem virtual de servidores, onde um servidor parece ser muitos para a rede. Isto é feito designando-se múltiplos endereços de rede para uma única interface.

Uma interface de rede possui um endereço "real", e pode possuir qualquer número de endereços de "alias". Estes *aliases* normalmente são adicionados através de entradas no arquivo `/etc/rc.conf`.

Uma entrada de *alias* para a interface `fxp0` é assim:

```
ifconfig_fxp0_alias0="inet xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx"
```

Note que as entradas de *alias* devem começar com `alias0` e continuar em ordem crescente, por (por exemplo, `_alias1`, `_alias2`, e assim sucessivamente). O processo de configuração se encerrará assim que faltar um número.

O cálculo das máscaras de *alias* é importante, mas ainda bem que é bem simples. Para uma interface qualquer, deve existir um endereço que represente corretamente a máscara de rede. Quaisquer outros endereços dentro desta rede devem ter a máscara de `1s`.

Por exemplo, considere o caso onde a interface `fxp0` é conectada à duas redes, a rede `10.1.1.0` com a máscara `255.255.255.0` e à rede `202.0.75.16` com a máscara `255.255.255.240`. Queremos que o sistema responda de `10.1.1.1` à `10.1.1.5` e de `202.0.75.17` à `202.0.75.20`.

As seguintes entradas configuram a placa para este arranjo:

```
ifconfig_fxp0="inet 10.1.1.1 netmask 255.255.255.0"
ifconfig_fxp0_alias0="inet 10.1.1.2 netmask 255.255.255.255"
ifconfig_fxp0_alias1="inet 10.1.1.3 netmask 255.255.255.255"
ifconfig_fxp0_alias2="inet 10.1.1.4 netmask 255.255.255.255"
ifconfig_fxp0_alias3="inet 10.1.1.5 netmask 255.255.255.255"
ifconfig_fxp0_alias4="inet 202.0.75.17 netmask 255.255.255.240"
ifconfig_fxp0_alias5="inet 202.0.75.18 netmask 255.255.255.255"
ifconfig_fxp0_alias6="inet 202.0.75.19 netmask 255.255.255.255"
ifconfig_fxp0_alias7="inet 202.0.75.20 netmask 255.255.255.255"
```

---

## 6.10. Arquivos de Configuração

### 6.10.1. Estrutura do `/etc`

Existem vários diretórios onde as informações de configuração são mantidas. Incluindo:

<code>/etc</code>	Informação genérica de configuração; aqui os dados são específicos do sistema.
<code>/etc/defaults</code>	Versões padrão dos arquivos de configuração do sistema.
<code>/etc/mail</code>	Configuração extra do <a href="#">sendmail(8)</a> , outros arquivos de configuração de Agentes de Transporte de Correio.
<code>/etc/ppp</code>	Configuração para programas ppp de usuário e de <i>kernel</i> .
<code>/etc/namedb</code>	Configuração padrão dos dados do <a href="#">named(8)</a> . Normalmente o arquivo <code>named.conf</code> e arquivos de zona são armazenados aqui.
<code>/usr/local/etc</code>	Arquivos de configuração para aplicações instaladas. Pode conter subdiretórios por aplicação.
<code>/usr/local/etc/rc.d</code>	<i>Scripts</i> de inicialização/parada para aplicações instaladas.
<code>/var/db</code>	Arquivos de banco de dados específicos de sistema gerados automaticamente, como base de dados de pacotes, a base de dados <i>locate</i> , e assim por diante.

### 6.10.2. Nomes de Servidores

#### 6.10.2.1. `/etc/resolv.conf`

O arquivo `/etc/resolv.conf` determina como o resolvidor do FreeBSD acessa o Sistema de Nome de Domínios da Internet (DNS).

As entradas mais comuns no arquivo `resolv.conf` são:

<code>nameserver</code>	O endereço IP do servidor de nomes que o resolvidor deve pesquisar. Os servidores são na ordem em que estão listados num máximo de três.
<code>search</code>	Lista de pesquisa para a procura do nome de servidor. Normalmente determinado pelo domínio do servidor local.
<code>domain</code>	O nome do domínio local.

Um resolv.conf típico:

```
search example.com
nameserver 147.11.1.11
nameserver 147.11.100.30
```

**Nota:** Apenas as opções *search* e *domain* devem ser usadas.

Se você estiver usando geralmente o [dhclient\(8\)](#), ele reescreve o resolv.conf com a informação recebida do servidor DHCP.

### 6.10.2.2. /etc/hosts

O arquivo /etc/hosts é uma base em texto remanescente da velha Internet. Funciona em conjunto com o servidores DNS e NIS oferecendo mapeamento de nome para endereços IP. Computadores locais conectados à Rede Local podem ser colocados aqui para propósitos simples de nomeamento ao invés de configurar um servidor [named\(8\)](#). Adicionalmente, o arquivo /etc/hosts pode ser usado para oferecer uma base local de nomes Internet, reduzindo a necessidade de pesquisa externa para nomes geralmente mais usados.

```
$FreeBSD$
#
Host Database
This file should contain the addresses and aliases
for local hosts that share this file.
In the presence of the domain name service or NIS, this file may
not be consulted at all; see /etc/nsswitch.conf for the resolution
order.
#
#
::1 localhost localhost.my.domain myname.my.domain
127.0.0.1 localhost localhost.my.domain myname.my.domain

#
Imaginary network.
#10.0.0.2 myname.my.domain myname
#10.0.0.3 myfriend.my.domain myfriend
#
According to RFC 1918, you can use the following IP networks for
private nets which will never be connected to the Internet:
#
10.0.0.0 - 10.255.255.255
172.16.0.0 - 172.31.255.255
192.168.0.0 - 192.168.255.255
#
In case you want to be able to connect to the Internet, you need
real official assigned numbers. PLEASE PLEASE PLEASE do not try
to invent your own network numbers but instead get one from your
network provider (if any) or from the Internet Registry (ftp to
rs.internic.net, directory `/templates').
#
```

O arquivo /etc/hosts possui o seguinte formato:

```
[Endereço Internet] [hostname oficial] [apelido1] [apelido2] ...
```

Por exemplo:

```
10.0.0.1 meuHostnameReal.exemplo.com meuHostnamereal foobar1 foobar2
```

Consulte [hosts\(5\)](#) para mais informações.

## 6.10.3. Configuração dos Arquivos de Log

### 6.10.3.1. syslog.conf

syslog.conf é um arquivo de configuração para o programa [syslogd\(8\)](#). Indica que tipo de mensagens do syslog são armazenadas nos arquivos de log.

```
$FreeBSD$
#
Spaces ARE valid field separators in this file. However,
other *nix-like systems still insist on using tabs as field
separators. If you are sharing this file between systems, you
may want to use only tabs as field separators here.
Consult the syslog.conf(5) manual page.
*.err;kern.debug;auth.notice;mail.crit /dev/console
*.notice;kern.debug;lpr.info;mail.crit;news.err /var/log/messages
security.* /var/log/security
mail.info /var/log/maillog
lpr.info /var/log/lpd-errs
cron.* /var/log/cron
*.err root
*.notice;news.err root
*.alert root
*.emerg *
uncomment this to log all writes to /dev/console to
/var/log/console.log
#console.info /var/log/console.log
uncomment this to enable logging of all log messages to
/var/log/all.log
#*. * /var/log/all.log
uncomment this to enable logging to a remote log host named loghost
#*. * @loghost
uncomment these if you're running inn
news.crit /var/log/news/news.crit
news.err /var/log/news/news.err
news.notice /var/log/news/news.notice
!startslip
*. * /var/log/slip.log
!ppp
*. * /var/log/ppp.log
```

Consulte a página de manual do [syslog.conf\(5\)](#) para mais informação.

### 6.10.3.2. newsyslog.conf

newsyslog.conf é o arquivo de configuração para o [newsyslog\(8\)](#), um programa que normalmente é agendado para ser executado pelo [cron\(8\)](#). O [newsyslog\(8\)](#) determina quando os arquivos de log necessitam ser arquivados ou reorganizados. arquivolog é renomeado para arquivolog.0, arquivolog.0 é renomeado para arquivolog.1, e assim sucessivamente. Alternativamente, os arquivos de log podem ser arquivados no

formato [gzip\(1\)](#), fazendo com que sejam nomeados: `arquivolog.0.gz`, `arquivolog.1.gz`, e assim sucessivamente.

`newsyslog.conf` indica os arquivos de log a serem gerenciados, quantos devem ser mantidos, e quando devem ser modificados. Arquivos de log podem ser rearranjados e/ou arquivados ao chegarem à um determinado tamanho, ou em um período de tempo/data.

```
configuration file for newsyslog
$FreeBSD$
#
filename [owner:group] mode count size when [ZB] [/pid_file]
[sig_num]
/var/log/cron 600 3 100 * Z
/var/log/amd.log 644 7 100 * Z
/var/log/kerberos.log 644 7 100 * Z
/var/log/lpd-errs 644 7 100 * Z
/var/log/maillog 644 7 * @T00 Z
/var/log/sendmail.st 644 10 * 168 B
/var/log/messages 644 5 100 * Z
/var/log/all.log 600 7 * @T00 Z
/var/log/slip.log 600 3 100 * Z
/var/log/ppp.log 600 3 100 * Z
/var/log/security 600 10 100 * Z
/var/log/wtmp 644 3 * @01T05 B
/var/log/daily.log 640 7 * @T00 Z
/var/log/weekly.log 640 5 1 $W6D0 Z
/var/log/monthly.log 640 12 * $M1D0 Z
/var/log/console.log 640 5 100 * Z
```

Consulte página de manual do [newsyslog\(8\)](#) para mais informação.

#### 6.10.4. `sysctl.conf`

`sysctl.conf` se parece muito com o `rc.conf`. Os valores são ajustados no formato *variável=valor*. Os valores especificados são configurados antes do sistema alcançar o modo multi usuário. Nem todas as variáveis são ajustáveis neste modo.

Um exemplo de `sysctl.conf` desligando a emissão de log de sinais de saídas fatais fazendo com que os programas Linux saibam que estão realmente sendo executados sob o FreeBSD.

```
kern.logsigexit=0 # Do not log fatal signal exits (e.g. sig 11)
compat.linux.osname=FreeBSD
compat.linux.osrelease=4.3-STABLE
```

---

## 6.11. Ajustando com o `sysctl`

O [sysctl\(8\)](#) é uma interface que permite que você faça mudanças em um sistema FreeBSD em tempo real. Isto inclui muitas opções avançadas da pilha TCP/IP e sistema de memória virtual que podem ser configurados através do [sysctl\(8\)](#).

Na sua essência, o [sysctl\(8\)](#) tem duas funções: ler e modificar configurações do sistema.

Para ver todas as variáveis que podem ser lidas:

```
% sysctl -a
```

Para ler uma variável em particular, por exemplo, *kern.maxproc*:

```
% sysctl kern.maxproc
kern.maxproc: 1044
```

Para configurar um valor em particular, use a sintaxe intuitiva *variável=valor*

```
sysctl kern.maxfiles=5000
kern.maxfiles: 2088 -> 5000
```

Configurações de variáveis do sysctl geralmente são *strings*, números ou booleanos (sendo um booleano *1* para sim ou *0* para não).

### 6.11.1. [sysctl\(8\)](#) Apenas para Leitura

*Contribuição de Tom Rhodes.*

Em alguns casos pode ser desejável modificar valores [sysctl\(8\)](#) que possam apenas ser lidos. Enquanto isto não seja recomendável, algumas vezes também pode ser inevitável.

Por exemplo, em alguns modelos de computadores portáteis o dispositivo [cardbus\(4\)](#) não provará algumas faixas de memória, e causará falhas com erros parecidos com este:

```
cbb0: Could not map register memory
device_probe_and_attach: cbb0 attach returned 12
```

Casos como este acima geralmente requerem a modificação de algumas configurações padrão do [sysctl\(8\)](#) que são apenas para leitura. Para mudar esta situação o usuário pode colocar ``OIDs" de [sysctl\(8\)](#) em seu arquivo local `/boot/loader.conf`. Configurações padrão estão localizadas no arquivo `/boot/defaults/loader.conf`.

Consertar o problema mencionado acima pode requerer que o usuário configure a opção `hw.pci.allow_unsupported_io_range=1` no arquivo mencionado anteriormente. Agora o [cardbus\(4\)](#) funcionará adequadamente.

## 6.12. Ajustando Discos

### 6.12.1. Variáveis Sysctl

#### 6.12.1.1. *vfs.vmiodirenable*

A variável sysctl *vfs.vmiodirenable* pode ser configurada tanto para 0 (desligada) ou 1 (ligada); onde o padrão é 1. Esta variável controla como os diretórios são cacheados pelo sistema. Muitos diretórios são pequenos, utilizando apenas um fragmento (tipicamente 1 K) no sistema de arquivos e menos no cache do *buffer* (tipicamente 512 bytes). Entretanto, ao operar em modo padrão o cache do *buffer* irá cachear apenas um número fixo de diretório ainda que você tenha uma grande quantidade de memória. Ligar este sysctl permite que o cache do *buffer* use o *VM Page Cache* para cachear os diretórios, fazendo com que toda memória disponível possa ser usada para cachear diretórios. Entretanto, o mínimo de memória usada para cachear um diretório é o tamanho de uma página física (tipicamente 4 K) ao invés de 512 bytes. Recomendamos que esta variável seja ligada se você estiver usando serviços que manipulem um grande número de arquivos. Serviços como web caches, grandes sistemas de correio eletrônico, e sistemas de notícias. Ligar esta variável geralmente não reduzirá a performance, mesmo com a memória desperdiçada, mas você deve tentar para descobrir.

#### 6.12.1.2. *vfs.write\_behind*

A variável sysctl *vfs.write\_behind* possui valor padrão 1 (ligada). Isto diz para que o sistema de arquivos faça uma escrita completa assim que uma seqüência de *clusters* seja coletada, isto tipicamente ocorre numa escrita de grandes arquivos sequenciais. A idéia é evitar a saturação do cache do *buffer* com *buffers* sujos quando isto não beneficiar a performance de E/S. Entretanto, isto pode paralisar processos, e sob algumas circunstâncias você pode querer desligar esta variável.

#### 6.12.1.3. *vfs.hirunningspace*

A variável sysctl *vfs.hirunningspace* determina a quantidade excedente de escrita de I/O que pode ser enfileirada nos controladores de disco globais do sistema em qualquer ocasião. O padrão normalmente é suficiente, mas em máquinas com muitos discos você pode querer aumentar para quatro ou cinco *megabytes*. Note que configurar um valor alto (excedendo o ponto inicial do buffer de escrita do cache) pode levar a uma performance de *clustering* degradante. Não configure esta variável com um valor arbitrariamente alto! Valores altos de escrita podem adicionar latência em leituras ocorridas simultaneamente.

Existem outras variáveis sysctl de cache de buffer e página VM. Não recomendamos a modificação de seus valores. Assim como no FreeBSD 4.3, o sistema VM realiza um excelente trabalho ajustando-se automaticamente.

#### 6.12.1.4. *vm.swap\_idle\_enabled*



A variável `sysctl vm.swap_idle_enabled` é útil em grandes sistemas multi-usuário onde você tem muitos usuários entrando e saindo do sistema e muitos processos ociosos. Estes sistemas tendem a gerar uma pressão considerável e contínua nas reservas de memória livre. Sintonizar esta característica e ajustar a histeria de *swapout* (em segundos fora de uso) através de `vm.swap_idle_threshold1` e `vm.swap_idle_threshold2` permite que você diminua mais rapidamente a prioridade das páginas de memória associadas à processos ociosos do algoritmo padrão de *pageout*. Isto dá uma ajuda ao *daemon de pageout*. Não ligue esta opção a menos que você precise, pois a troca que você está fazendo é essencialmente memória de pré-página antes ao invés de tardia; consumindo mais *swap* e banda de disco. Em um sistema pequeno esta opção terá um efeito determinável, mas em um sistema maior que já está fazendo uma paginação moderada esta opção permite ao sistema VM organizar os processos que entram e saem da memória mais facilmente.

#### 6.12.1.5. *hw.ata.wc*

O FreeBSD 4.3 flertou com o desligamento do cache de escrita IDE. Isto reduziu a largura de banda de escrita em discos IDE e foi considerado necessário devido a sérias questões de consistência de dados introduzidas pelos fabricantes de discos rígidos. O problema é que discos IDE mentem quando uma escrita é completada. Com o cache de escrita de IDE ligado, estes discos não somente escrevem dados no disco fora de ordem, mas algumas vezes atrasará a escrita de alguns blocos indefinidamente quando estiver sob grande carga. Um problema ou uma falha de energia pode causar corrupções sérias no sistema de arquivos. O padrão do FreeBSD foi modificado para ser seguro. Infelizmente, o resultado foi uma grande perda de performance, e voltamos a ligar o cache de escrita por padrão. Você pode checar o padrão do seu sistema observando a variável `sysctl hw.ata.wc`. Se o cache de escrita IDE estiver desligado, você pode ligá-lo de volta configurando a variável de *kernel* de volta para 1. Isto deve ser feito pelo carregador de inicialização durante a inicialização. Tentar fazer isso depois da inicialização do *kernel* não surtirá efeito.

Para mais informações veja [ata\(4\)](#).

#### 6.12.1.6. *SCSI\_DELAY (kern.cam.scsi\_delay)*

A configuração de kernel *SCSI\_DELAY* pode ser usada para reduzir o tempo de inicialização. O padrão é um pouco alto e pode ser responsável 15+ segundos de espera no processo de inicialização. Reduzir para 5 segundos funciona bem (especialmente para drives modernos). Versões mais novas do FreeBSD (5.0+) podem ajustar a variável `kern.cam.scsi_delay`. As opções de configuração de kernel e ajuste de variável aceitam valores em *milisegundos* e não em *seconds*.

### 6.12.2. Soft Updates

O programa [tunefs\(8\)](#) pode ser usado para fazer um ajuste fino no sistema de arquivos. Este programa possui muitas opções, mas por enquanto vamos nos preocupar apenas em ligar e desligar o *Soft Updates* da seguinte maneira:

```
tunefs -n enable /sistemadearquivos
tunefs -n disable /sistemadearquivos
```

Um sistema de arquivos não pode ser modificado com o [tunefs\(8\)](#) enquanto estiver montado. Uma boa hora para habilitar o *Soft Updates* é antes que qualquer partição seja montada, em modo mono-usuário.

**Nota:** Assim como no FreeBSD 4.5, é possível habilitar o *Soft Updates* no momento da criação do sistema de arquivos, através da opção `-U` do [newfs\(8\)](#).

O *Soft Updates* melhora drasticamente a performance do *meta-dados*, principalmente a criação e eliminação de arquivos, através do uso de um cache de memória. Recomendamos o uso do *Soft Updates* em todos os seus sistemas de arquivos. Existem dois pequenos problemas no seu uso, que você deve estar ciente: Primeiro, o *Soft Updates* garante a consistência no sistema de arquivos no caso de problemas, mas pode ficar alguns segundos (até mesmo minuto!) atualizando o disco físico em segundo plano. Se seu sistema der problema você pode perder mais dados do que da outra maneira. Segundo, o *Soft Updates* atrasa a liberação de blocos do sistema de arquivo. Se você possui um sistema de arquivo (como a raiz) que está praticamente lotado, fazer uma atualização geral tipo `make installworld`, pode fazer com que o sistema de arquivos fique sem espaço e falhe a atualização.

### 6.12.2.1. Mais detalhes sobre o Soft Updates

Existe duas maneiras tradicionais de escrever meta-dados de sistemas de arquivos de volta para o disco. (Atualização de meta-dados são atualizações para dados que não sejam de conteúdo, como inodos ou diretórios.)

Historicamente, o comportamento padrão era o de escrever atualizações de meta-dados sincronamente. Se um diretório foi modificado, o sistema esperou até que a mudança fosse realmente feita no disco. Os buffers de dados de arquivo (conteúdo do arquivo) foram passados através do buffer de cache e guardados mais tarde em disco assincronamente. A vantagem desta implementação é a segurança na operação. Se existir uma falha durante uma atualização, os meta-dados sempre estarão em um estado consistente. Um arquivo é criado de forma completa ou não de qualquer forma. Se os blocos de dados de um arquivo não encontrar seu caminho do cache de buffer no disco na hora do problema, o [fsck\(8\)](#) será capaz de reconhecer e reparar o sistema de arquivos configurando o tamanho do arquivo para 0. Além disso, a implementação é clara e simples. A desvantagem é que as mudanças de meta-dados são lentas. O comando `rm -r`, por exemplo, toca em todos os arquivos em um diretório sequencialmente, mas cada mudança de diretório (exclusão de um arquivo) será escrita sincronamente no disco. Isto inclui atualizações no próprio diretório, na tabela de inodo, e possivelmente nos blocos indiretamente alocados pelo arquivo. Considerações similares são aplicáveis no desenrolar de hierarquias grandes (`tar -x`).

O segundo caso é o de atualizações assíncronas de meta-dados. Isto é o padrão para Linux/ext2fs e `mount -o async` para ufs nos \*BSD. Todas as atualizações de meta-dados são simplesmente passadas pelo cache de buffer também, eles serão misturados com as atualizações de conteúdo de dados de arquivo. A vantagem desta implementação é que não existe a necessidade de esperar até que cada atualização de meta-dados tenha sido escrita em disco, uma vez que todas as operações que causam grandes quantidades de atualização de meta-dados trabalhem mais rápido que no caso síncrono. Além disso, a implementação é simples e limpa, possuindo um baixo risco de falhas na codificação.

A desvantagem é que não existe garantia para um estado consistente do sistema de arquivos. Se ocorrer uma falha durante uma operação que atualizou grandes quantidades de meta-dados (como uma falta de energia, ou alguém pressionar o botão de reinicialização), o sistema de arquivos ficará em um estado imprevisível. Não existe uma oportunidade para examinar o estado do sistema de arquivos quando o sistema voltar para o ar novamente; os blocos de dados de um arquivo já podem ter sido escritos no disco enquanto as atualizações na tabela de inodo, ou diretório associado, ainda não foram. Na verdade é impossível implementar um `fsck` que seja capaz de limpar todo o caos resultante (pelo fato de que a informação necessária não está disponível no disco). Se o sistema de arquivos foi danificado além das possibilidades de reparação, a única escolha é utilizar o [newfs\(8\)](#) e restaurar um backup.

A solução normal para este problema foi a implementação de uma *região suja de logging*, também conhecida como *journaling*, embora este termo não seja usado consistentemente e é ocasionalmente aplicado à outras formas de transação de *logging*. Atualizações de meta-dados ainda são escritas sincronamente, mas apenas em uma pequena região do disco. Posteriormente serão movidos para a localização apropriada. Devido ao pequeno tamanho da área de *logging*, a região contígua do disco, não existem grandes distâncias a serem percorridas pela cabeça do disco, mesmo durante operações pesadas, fazendo com que estas operações sejam mais rápidas do que as atualizações síncronas. Além disso, a complexidade de implementação é um tanto quanto limitada, fazendo com que o risco de existência de falhas na codificação sejam menores. A desvantagem é que todos os meta-dados são escritos duas vezes (uma vez na região de *logging* e outra no local apropriado) causando perda de performance em condições normais. Por outro lado, no caso de pane, todas as operações de meta-dados pendentes podem ser rapidamente desfeitas ou completadas da área de *logging* depois que o sistema retornar ao ar novamente, resultando uma inicialização rápida de sistema de arquivos.

Kirk McKusick, desenvolvedor do Berkeley FFS, resolveu este problema com o Soft Updates: todas as atualizações de meta-dados pendentes são mantidas em memória e escritas no disco numa seqüência ordenada ("ordered meta-data updates"). Isto causa o seguinte efeito, no caso de operações pesadas de meta-dados, atualizações posteriores em um item "pegam" as primeiras que ainda estão em memória e não foram escritas no disco. Então todas as operações em, digamos, um diretório, geralmente são feitas em memória antes que a atualização seja escrita no disco (os blocos de dados são ordenado de acordo com seu posicionamento, assim não podem estar no disco antes de seus meta-dados). Se o sistema der pane, a consequência será um "log rewind": todas as operações que não encontraram seu caminho para o disco parecem nunca ter ocorrido. Um estado consistente de sistema de arquivo é mantido como se fosse de 30 a 60 segundos antes. O algoritmo utilizado garante que todos os recursos em uso sejam marcados com seus mapas de bits apropriados: blocos e inodos. Depois de uma pane, a única alocação de erros de recursos que ocorre são os marcados como "used" que na verdade são "free". O [fsck\(8\)](#) reconhece esta situação, e libera os recursos que não estejam em uso. É seguro ignorar o estado sujo do sistema de arquivo após uma pane forçando montá-lo com `mount -f`. Para liberar recursos que podem estar ociosos, o [fsck\(8\)](#) necessita ser executado posteriormente. Esta é a idéia atrás do *background fsck*: no momento da inicialização do sistema, apenas um *snapshot* do sistema de arquivo é gravada. O `fsck` pode ser executado depois. Todos os sistemas de arquivo podem ser montados "sujos", para que a inicialização do sistema prossiga em modo multi-usuário. Então o `fsck` em

segundo plano será agendado para ser executado em todos os sistemas de arquivo onde for necessário, para liberar recursos que podem estar ociosos. (Sistemas de arquivos que não usam Soft Updates ainda necessitam do `fsck` em primeiro plano.)

A vantagem é que as operações de meta-dados são quase tão rápidas quando as atualizações assíncronas (mais rápidas do que com *logging*, que precisa escrever os meta-dados duas vezes). As desvantagem são a complexidade do código (implicando em um alto risco de falhas de código em uma área extremamente sensível no que diz respeito a perda de dados do usuário), e um alto consumo de memória. Além disso existem algumas peculiaridades que temos que lidar. Depois de uma pane, o estado do sistema de arquivos parece estar algo como ``antigo". Em situações onde a opção síncrona pode ter causado alguns arquivos de tamanho zero após o `fsck`, estes arquivos não existem em um sistema de arquivo com Soft Update, pois nem os meta-dados e os conteúdos dos arquivos foram escritos no disco. O espaço em disco não é liberado até que as atualizações tenham sido escritas no disco, que pode acontecer algum tempo depois de executar o comando `rm`. Isto pode causar problemas ao introduzir grandes quantidades de dados em um sistema de arquivo que não possui espaço livre suficiente para abrigar todos os arquivos duas vezes.

---

## 6.13. Ajustando Limites do Kernel

### 6.13.1. Limites de Arquivo/Processo

#### 6.13.1.1. *kern.maxfiles*

A variável *kern.maxfiles* pode ter seu valor aumentado ou diminuído baseado nos requisitos do sistema. Esta variável indica o número máximo de descritores de arquivos no seu sistema. Quando a tabela de descritores de arquivos está cheia, a mensagem ``file: table is full" aparecerá várias vezes no buffer de mensagens do sistema, que pode ser visualizado através do comando `dmesg`.

Cada arquivo aberto, socket, ou fifo usa um descritor de arquivo. Um servidor de produção de larga escala pode facilmente requerer muitos milhares de descritores de arquivos, dependendo do tipo e do número de serviços sendo executados concorrentemente.

O valor padrão da variável *kern.maxfile* é definido pela opção *MAXUSERS* no seu arquivo de configuração de kernel. *kern.max.files* cresce proporcionalmente ao valor de *MAXUSERS*. Ao compilar um kernel customizado, é uma boa idéia modificar esta configuração de acordo com o uso do seu sistema. A partir deste número, o kernel toma base para muitos limites pré-definidos. Embora uma máquina em produção não tenha 256 usuários conectados ao mesmo tempo, os recursos requeridos podem ser similares aos de um servidor de ponta.

**Nota:** Assim como no FreeBSD 4.5, configurar a opção *MAXUSERS* para 0 no seu arquivo de configuração de kernel faz com que um valor padrão razoável seja configurado de acordo com a memória RAM presente no seu sistema.

### 6.13.1.2. *kern.ipc.somaxconn*

A variável `sysctl kern.ipc.somaxconn` limita o tamanho da fila de escuta para aceitação de novas conexões TCP. O valor padrão de 128 é tipicamente baixo para uma manipulação robusta de novas conexões em um ambiente de servidor web com alta carga. Para tais ambientes o aumento deste valor para 1024 ou mais é recomendado. O serviço de daemon pode por si só limitar o tamanho da fila (por exemplo, [sendmail\(8\)](#), ou **Apache**) mas na maioria das vezes existirá uma diretiva em seus arquivos de configuração para ajustar o tamanho da fila. Filas de escuta grandes também podem fazer um bom trabalho evitando ataques de Negação de Serviço (DoS).

### 6.13.2. Limites de Rede

A opção de configuração de kernel *NMBCLUSTERS* dita a quantidade de *Mbufs* de rede disponível para o sistema. Um servidor com muito tráfego e um número pequeno de *Mbufs* limitarão a habilidade do FreeBSD. Cada cluster representa aproximadamente 2 K de memória, então um valor de 1024 representa 2 megabytes de memória de kernel reservada para buffers de rede. Um cálculo simples pode ser feito para saber quantos são necessários. Se você possui um servidor web que chega a um máximo de 1000 conexões simultâneas, e cada conexão consome 16 K de buffer de envio e 16 K de recepção, você precisa de aproximadamente 32 MB de buffers de rede para cobrir seu servidor web. Uma boa regra geral é multiplicar por 2, então  $2 \times 32 \text{ MB} / 2 \text{ KB} = 64 \text{ MB} / 2 \text{ KB} = 32768$ . Recomendamos valores entre 4096 e 32768 para máquinas com grandes quantidades de memória. Sob nenhuma circunstância você deve especificar valores arbitrariamente altos para este parâmetro, pois pode causar travamentos durante a inicialização. A opção *-m* do [netstat\(1\)](#) pode ser usada para observar o uso do cluster de rede.

*kern.ipc.nmbclusters* pode ser usado para ajustar isto no momento da inicialização. Somente versões antigas do FreeBSD irão requerer o uso da opção [config\(8\)](#) de kernel *NMBCLUSTERS*.

Para servidores mais ocupados, que fazem uso extensivo da chamada de sistema [sendfile\(2\)](#), pode ser necessário aumentar o número de buffers [sendfile\(2\)](#) através da opção de configuração de kernel *NSFBUFFS* colocando seu valor no arquivo `/boot/loader.conf` (veja [loader\(8\)](#) para mais detalhes). Um indicador comum que indica que este parâmetro precisa ser ajustado é quando processos são vistos no estado ```_sfbufa```. A variável `sysctl kern.ipc.nsfbufs` oferece uma visão apenas de leitura de como esta variável está configurada no kernel. Este parâmetro aumenta nominalmente com o *kern.maxusers*, entretanto pode ser necessário ajustar de acordo com a necessidade.

**Importante:** Mesmo que o *socket* tenha sido marcado como não bloqueador, invocar o [sendfile\(2\)](#) neste *socket* pode resultar em bloqueamento de chamadas no [sendfile\(2\)](#) até que *struct sf\_bufs* tenham sido disponibilizados.

### 6.13.2.1. *net.inet.ip.porrange.\**

A variável `sysctl net.inet.ip.porrange.*` controla a faixa de número de portas automaticamente ligadas a *sockets* TCP e UDP. Existem três faixas: a baixa, a padrão e a faixa alta. Muitos programas de rede usam a faixa padrão, que é controlada pela variável `net.inet.ip.porrange.first` e `net.inet.ip.porrange.last`, que possuem valores padrão 1024 e 5000, respectivamente. Faixas de porta padrão são usadas para conexões que saem, e é possível ficar sem portas sob certas circunstâncias. Isto ocorre de forma mais comum quando você roda um servidor proxy que tem muita carga. A faixa de portas não é um problema quando se executa servidores cujo papel principal é receber conexões, como um servidor web normal, ou um servidor que possui um número limitado de conexões que saem, como um *relay* de correio. Para situações onde você pode ficar sem portas, é recomendado que se aumente modestamente a variável `net.inet.ip.porrange.last`. Um valor de 10000, 20000 ou 30000 deve ser suficiente. Você também deve considerar os efeitos colaterais que a mudança de faixa de portas pode causar em um firewall. Alguns firewalls podem bloquear grandes faixas de portas (geralmente portas de números baixos) e esperar que os sistemas utilizem faixas de portas altas para conexões que saem -- por esta razão recomenda-se que a variável `net.inet.ip.porrange.first` tenha seu valor diminuído.

### 6.13.2.2. TCP Bandwidth Delay Product

A Limitação do Produto de Atraso de Banda TCP é similar ao TCP/Vegas no **NetBSD**. Pode ser habilitado através da configuração da variável `sysctl net.inet.tcp.inflight_enable` para o valor *1*. O sistema tentará calcular o produto do atraso de banda para cada conexão e limitar a quantidade de dados enfileirados para a rede para apenas a quantidade requerida, com o objetivo de otimizar a quantidade de dados entrando e saindo.

Esta característica é útil se você está servindo dados através de modems, Gigabit Ethernet, ou até mesmo links WAN de alta velocidade (ou qualquer outro link com um alto produto de atraso de banda), especialmente se você também está usando escalamento de janela ou possui uma grande janela de envio configurada. Se você habilitar esta opção, você deve ter certeza de configurar a variável `net.inet.tcp.inflight_debug` para *0* (desabilitar depuração), e para produção configurar `net.inet.tcp.inflight_min` para pelo menos *6144* pode ser benéfico. Entretanto, note que configurar valores mínimos altos pode efetivamente desabilitar a limitação de banda dependendo do link. A característica de limitação reduz a quantidade de dados construídos na roda intermediária e trocar as filas de pacotes assim como reduzir a quantidade de dados construídos na interface de enfileiramento da máquina local. Com poucos pacotes enfileirados, conexões interativas, especialmente sob modems lentos, também serão capazes de operar com tempos reduzidos de *Round Trip*. Entretanto, note que esta característica tem efeito apenas na transmissão de dados (envio de dados / lado do servidor). Não tem efeito na recepção de dados (*download*)

Ajustar o valor de `net.inet.tcp.inflight_stab` não é recomendado. Este parâmetro tem valor 20 como padrão, representando 2 pacotes máximos adicionados ao cálculo de janela de produto de atraso de banda. A janela adicional é requerida para estabilizar o algoritmo e melhorar a resposta em condições de mudança, mas também pode resultar em tempos altos de ping em links lentos (ainda mais lentos do que você teria sem o

algoritmo *inflight*). Nestes casos, você pode querer tentar reduzir este parâmetro para 15, 10 ou 5; e talvez tenha que reduzir a variável *net.inet.tcp.inflight\_min* (por exemplo, para 3500) para obter o efeito desejado. Reduzir estes parâmetros deve ser feito como um último recurso somente.

---

## 6.14. Adicionando Espaço de Swap

Não importa o que você planeje, algumas vezes o sistema não roda como você espera. Se você acha que precisa de mais espaço de swap, é simples adicionar. Você tem três maneiras de fazer isso: adicionar um novo disco, habilitar swap através de NFS, e criar um arquivo de swap em uma partição existente.

### 6.14.1. Swap em um Novo Disco Rígido

A melhor maneira de adicionar swap, claro, é usar isto como desculpa para adicionar outro disco rígido. Você pode sempre usar outro disco rígido afinal de contas. Se você puder fazer isto, releia a discussão [espaço de swap](#) da seção do Handbook [Configuração Inicial](#) para sugestões de quais os melhores meios para arrumar seu swap.

### 6.14.2. Swap através de NFS

Fazer swap através de NFS somente é recomendado se você não possui um disco local para fazer swap. É lento e ineficiente em versões do FreeBSD anteriores a 4.X. É razoavelmente rápido na 4.0-RELEASE e superior. Mesmo com versões mais novas do FreeBSD, swap de NFS será limitado pela largura de banda da rede colocando uma carga extra no servidor NFS.

### 6.14.3. Arquivos de Swap

Você pode criar um arquivo de tamanho específico para utilizar como arquivo de swap. Em nosso exemplo usaremos um arquivo de 64MB chamado `/usr/swap0`. Você pode usar qualquer nome que desejar, é claro.

#### Exemplo 6-1. Criando um Arquivo de Swap no FreeBSD 4.X

1. Tenha certeza de que seu arquivo de configuração de *kernel* inclui o driver *vnode*. Ele *não* está nas versões recentes do arquivo `GENERIC`.
2. `pseudo-device vn 1 #Vnode driver (turns a file into a device)`
3. crie um vn-device:
4. `# cd /dev`
5. `# sh MAKEDEV vn0`
6. crie um arquivo de swap (`/usr/swap0`):
7. `# dd if=/dev/zero of=/usr/swap0 bs=1024k count=64`
8. configure as permissões apropriadas em (`/usr/swap0`):
9. `# chmod 0600 /usr/swap0`
10. habilite o arquivo de swap em `/etc/rc.conf`:

11. `swapfile="/usr/swap0" # Ajuste o nome do arquivo de swap desejado.`
12. Reinicialize a máquina ou habilite o arquivo de swap imediatamente, digite:
13. `# vnconfig -e /dev/vn0b /usr/swap0 swap`

### Exemplo 6-2. Criando um Arquivo de Swap no FreeBSD 5.X

1. Tenha certeza de que seu arquivo de configuração de *kernel* tenha o driver de memória ([md\(4\)](#)). Está no `GENERIC` por padrão.
  2. `device md # Memory "disks"`
  3. crie um arquivo de swap (`/usr/swap0`):
  4. `# dd if=/dev/zero of=/usr/swap0 bs=1024k count=64`
  5. configure as permissões em (`/usr/swap0`):
  6. `# chmod 0600 /usr/swap0`
  7. habilite o arquivo de swap no `/etc/rc.conf`:
  8. `swapfile="/usr/swap0" # Ajuste o nome do arquivo de swap desejado.`
  9. Reinicialize a máquina ou habilite o arquivo de swap imediatamente, digite:
  10. `# mdconfig -a -t vnode -f /usr/swap0 -u 0 && swapon /dev/md0`
- 

## 6.15. Gerenciamento de Energia e Recursos

*Escrito por Hiten Pandya e Tom Rhodes.*

É muito importante utilizar os recursos de hardware de maneira eficiente. Antes do ACPI ser introduzido, era muito difícil e inflexível para os sistemas operacionais gerenciar o uso de energia e propriedades térmicas de um sistema. O hardware era controlado por algum tipo de interface BIOS embarcada, ou *Plug and Play BIOS (PNPBIOS)*, ou *Gerenciamento de Energia Avançado (APM)* e assim por diante. Gerenciamento de Energia e Recursos é um dos componentes chave de um sistema operacional moderno. Por exemplo, você pode querer que o sistema operacional monitore os limites do sistema (e possivelmente alertar você) caso a temperatura do seu sistema aumente inesperadamente.

Nesta seção do Handbook do FreeBSD ofereceremos informações sobre ACPI. Daremos referências de leituras mais completas no final. Por favor, saiba que o ACPI está disponível no FreeBSD 5.X e superiores como um módulo padrão de *kernel*. Para o FreeBSD 4.9, ACPI pode ser habilitado adicionando-se a linha `device acpi` ao arquivo de configuração do *kernel* e recompilando-se o sistema.

### 6.15.1. O que é ACPI?

Configuração Avançada e Interface de Energia (ACPI) é um padrão escrito por um consórcio de fornecedores para oferecer um padrão de interface para recursos de hardware e gerenciamento de energia. É um elemento chave na *Configuração direcionada ao Sistema e Gerenciamento de Energia*, ex.: oferece mais controle e flexibilidade para o sistema operacional (SO). Sistemas modernos, ``arrebentaram" os



limites das interfaces *Plug and Play* atuais (tais como APM, que é usada no FreeBSD 4.X), antes da introdução do ACPI. ACPI é o sucessor direto do APM (Gerenciamento Avançado de Energia).

### 6.15.2. Contratempos do Gerenciamento Avançado de Energia (APM)

O *Gerenciamento Avançado de Energia (APM)* facilita o controle do uso de energia de um sistema baseado em sua atividade. A BIOS APM é oferecida pelo fornecedor (do sistema) e é específica à plataforma de hardware. Um driver APM no SO media o acesso ao *Programa de Interface APM*, que permite o gerenciamento de níveis de energia.

Existem quatro problemas principais no APM. Primeiro, o gerenciamento de energia é feito pela BIOS (específica do fornecedor), e o SO não tem nenhum conhecimento dela. Um exemplo disso é quando o usuário configura valores de tempo ocioso para discos rígidos na BIOS APM que, quando excedidos, ela (BIOS) desliga o disco sem o consentimento do SO. Segundo, a lógica do APM é embarcada na BIOS, e opera fora do escopo do SO. Isto significa que os usuários só podem consertar problemas em suas BIOS APM gravando uma nova ROM; que é um procedimento extremamente perigoso, e se falhar, pode levar o sistema à um estado irreversível. Terceiro, APM é uma tecnologia específica do fornecedor, o que significa que existem muitas paridades (duplicação de esforços) e falhas encontradas nas BIOS destes fornecedores que podem não ser consertadas nas de outros fornecedores. Última mas não menos importante, a BIOS APM não possui espaço suficiente para implementar uma política sofisticada de energia, ou uma que possa se adaptar bem ao propósito da máquina.

*BIOS Plug and Play (PNPBIOS)* era inconfiável em muitas situações. PNPBIOS é uma tecnologia de 16 bits, assim o SO tem que emular 16 bits para fazer "interface" com os métodos PNPBIOS.

O driver APM do FreeBSD está documentado na página de manual do [apm\(4\)](#).

### 6.15.3. Configurando ACPI

O driver `acpi.ko` é carregado por padrão na inicialização pelo [loader\(8\)](#) e não deve ser compilado no *kernel*. A razão por trás disso é que os módulos são mais fáceis de se trabalhar, digamos se trocar para outro `acpi.ko` sem que seja preciso recompilar o *kernel*. Isto tem a vantagem de facilitar os testes. Uma outra razão é que inicializar o ACPI depois do sistema estar no ar não é tão útil, e em alguns casos pode ser fatal. Na dúvida, apenas desabilite o ACPI. Este driver não deve e não pode ser desabilitado pois o barramento do sistema o usa para várias interações de hardware. O ACPI pode ser desabilitado com o utilitário [acpicconf\(8\)](#). Na verdade, a maior parte das interações com o ACPI pode ser feita através do [acpicconf\(8\)](#). Basicamente isto significa que se qualquer coisa a respeito do ACPI estiver na saída do [dmesg\(8\)](#), então provavelmente já estará rodando.

**Nota:** ACPI e APM não podem coexistir e devem ser usados separadamente. O último a ser carregado será finalizado se o driver tiver notícias que o outro está sendo executado.

Numa forma mais simples, o ACPI pode ser usado para colocar o sistema em modo de espera com a opção `-s` e `I-5` do [acpicnf\(8\)](#).

```
halt -p
```

Outras opções estão disponíveis. Confira a página de manual [acpicnf\(8\)](#) para mais informações.

#### 6.15.4. Depurando e Desabilitando o ACPI

Quase tudo no ACPI é transparente, até que não funcione. Você como usuário saberá quando alguma coisa não funcionar de acordo. O driver [acpi\(4\)](#) suporta várias opções de depuração, é até possível seletivamente desabilitar algumas partes do sistema ACPI. Para mais informações a respeito de facilidades de depuração leia a página de manual do [acpi\(4\)](#).

Algumas vezes, por várias razões, o módulo `acpi.ko` deve ser descarregado. Isto só pode ser feito no momento da inicialização pelo [loader\(8\)](#). Você pode digitar no prompt do [loader\(8\)](#) o comando `unset acpi_load` cada vez que você inicializar o sistema, para interromper o carregamento automático do driver [acpi\(4\)](#) adicione a seguinte linha no arquivo `/boot/loader.conf`:

```
exec="unset acpi_load"
```

O FreeBSD 5.1-RELEASE e superiores vem com um menu em tempo de inicialização que controla como o FreeBSD é inicializado. Uma das opções propostas é a de desligar o ACPI. Para desabilitá-lo selecione 2. Boot FreeBSD with ACPI disabled no menu.

---

## Capítulo 7. O processo de inicialização do FreeBSD

### Índice

- 7.1. [Sinótese](#)
- 7.2. [Problema na inicialização](#)
- 7.3. [A MBR, e os estágios de inicialização Um, Dois, e Três](#)
- 7.4. [Interação do Kernel durante o boot](#)
- 7.5. [Dicas de configuração de dispositivos](#)
- 7.6. [Init: Inicialização do controle de processos](#)
- 7.7. [Sequência do Shutdown](#)

### 7.1. Sinótese

O processo de inicialização e carregamento de um sistema operacional refere-se a um "processo de inicializar", ou simplesmente "carregar". O processo de inicialização do FreeBSD fornece uma grande flexibilidade que acontece quando você inicia o sistema,

permitindo que você selecione diferentes operações no sistema instalado no computador, ou igualar diferentes versões do mesmo sistema operacional ou do kernel instalado.

Este capítulo detalha as opções de configuração que você pode ajustar e personalizar o processo de inicialização do FreeBSD. Isto inclui tudo o que acontece desde que o kernel do FreeBSD foi inicializado, teste de dispositivos, e inicialização do [init\(8\)](#). Se você não souber quando isso acontece, isto acontece quando a cor do texto muda de branco brilhante para cinza.

Depois de ler este capítulo, você saberá:

- O que os componentes do sistema de inicialização do FreeBSD são, e como eles interagem.
- As opções que você pode dar a componentes de inicialização do FreeBSD para controlar o processo de inicialização.
- O básico do [device.hints\(5\)](#).

**Somente x86:** Este capítulo descreve somente o processo de inicialização para FreeBSD rodando em sistemas Intel x86.

---

## 7.2. Problema na inicialização

Voltando a um computador inicializando o sistema operacional a atitude é um dilema interessante. Por definição, o computador não sabe qualquer coisa antes que o sistema operacional esteja iniciando. Isto inclui rodar programas do disco. Assim se o computador não pode rodar um programa do disco sem o sistema operacional, e os programas do sistema operacional estão no disco, como o sistema operacional foi inicializado?

Este problema paralelo estão no livro *As aventuras de Baron Munchausen*. O personagem despencou por um boeiro, e puxou a si mesmo pelos cadarços (*bootstraps*). Atualmente na computação, o termo *bootstrap* refere-se ao mecanismo utilizado para carregar o sistema operacional, que abreviado tornou-se o proceso de *booting*.

No hardware x86 o Sistema Básico de Entrada/Saída (BIOS) é responsável por carregar o sistema operacional. Para fazer isto, na BIOS aparece o disco rígido para o Master Boot Record (MBR), que deve ser alocado sobre um local específico do disco. A BIOS tem bastante mudanças que podem ser feitas para carregar e rodar a MBR, e assume que a MBR pode então carregar a saída do resto das tarefas envolvidas na inicialização do sistema operacional.

Se você tiver somente um sistema operacional instalado no seu disco então a MBR padrão será suficiente. Esta MBR irá procurar pela primeira partição inicializável no disco, e entao roda o código nesta partição para carregar o restante do sistema operacional.

Se você tiver instalado vários sistemas operacionais em seus discos então você pode instalar uma MBR diferente, que possa mostrar uma lista de diferentes sistemas operacionais, e permitir a você escolher um para carregar. FreeBSD vem com um MBR que pode ser instalado, e outros sistemas operacionais vendidos fornecem MBRs alternativos.

O restante do sistema *bootstrap* do FreeBSD é dividido em três estágios. O primeiro estágio é executado pela MBR, que sabe apenas iniciar o computador em um estado específico e rodar o segundo estágio. O segundo estágio pode fazer mais um pedaço em pouco tempo, depois roda o terceiro estágio. O terceiro estágio terminará a tarefa de carregar o sistema operacional. O trabalho é dividido dentro destes três estágios porque o padrão do PC impõe limites sobre o tamanho dos programas que podem ser rodados nos estágios um e dois. Acorrentar as tarefas permite que o FreeBSD forneça um carregador mais flexível.

O kernel então é inicializado e começa a testar os dispositivos, e então inicializa estes para o uso. Uma vez que o processo de inicialização do kernel é terminado, o kernel passa o controle do processo para [init\(8\)](#), o qual se certifica que os discos estão em um estado aproveitável. [init\(8\)](#) começa então a configuração a nível-de-usuário que monta o sistema de arquivos, seta e levanta dispositivos de rede para comunicação com a mesma, e normalmente inicia todos os processos que são rodados sobre um sistema FreeBSD na inicialização.

---

## 7.3. A MBR, e os estágios de inicialização Um, Dois, e Três

### 7.3.1. MBR, `/boot/boot0`

A MBR do FreeBSD é localizada no `/boot/boot0`. Esta é uma *cópia* do MBR, por que a real MBR deve ficar em uma parte especial do disco, fora da área do FreeBSD.

`boot0` é muito simples, visto que o programa dentro da MBR pode ter no máximo 512 bytes. Se você tiver instalado na MBR o FreeBSD e instalado múltiplos sistemas operacionais em seu HD então você verá um menu similar a este na hora da inicialização:

#### Exemplo 7-1. `boot0` Screenshot

```
F1 DOS
F2 FreeBSD
F3 Linux
F4 ??
F5 Drive 1

Default: F2
```

Outros sistemas operacionais, em particular Windows® 95, tem conhecimento para sobrescrever suas próprias informações em uma MBR já existente. Se isto acontecer a você, ou você queira substituir sua MBR existente com o FreeBSD MBR então use o seguinte comando:

```
fdisk -B -b /boot/boot0 device
```

Onde *device* é o dispositivo que você carrega, como `ad0` para o primeiro disco IDE, `ad2` para o primeiro disco IDE da segunda controladora IDE, `da0` para o primeiro disco SCSI, e assim por diante.

Se você for um usuário Linux, contudo, e preferir que o **LILO** controle o processo de inicialização, você pode editar o arquivo `/etc/lilo.conf` para FreeBSD, ou selecionar *Leave The Master Boot Record Untouched* durante o processo de instalação do FreeBSD. Se você tiver instalado o gerenciador de boot do FreeBSD, você pode carregar dentro do Linux e modificar o **LILO** no arquivo de configuração `/etc/lilo.conf` e adicionar a seguinte opção:

```
other=/dev/hdXY
table=/dev/hdb
loader=/boot/chain.b
label=FreeBSD
```

o qual permitirá a inicialização do FreeBSD e Linux via **LILO**. Em outro exemplo, nós usamos *XY* para determinar o número do drive e da partição. Se estiver usando um drive SCSI, você irá querer mudar o `/dev/hdXY` para leitura algo similar a `/dev/sdXY`, novamente usando a sintaxe *XY*. O `loader=/boot/chain.b` pode ser omitido se você tiver ambos os sistemas operacionais na mesma partição. Você pode agora rodar `/sbin/lilo -v` para executar suas novas alterações no sistema, este deve ser verificado com mensagens na tela.

### 7.3.2. Estágio Um, `/boot/boot1`, e Estágio Dois, `/boot/boot2`

Conceitualmente o primeiro e segundo estágios são parte do mesmo programa, sobre a mesma área do disco. Devido o espaço contido eles foram divididos em duas partes, mas você deve instalá-los juntos.

Eles são encontrados no setor de inicialização, que é onde `boot0`, ou alguns outros programas na MBR espera encontrar um programa para rodar e continuar o processo de inicialização. O arquivo no diretório `/boot` são cópias dos arquivos reais, que são armazenados fora do sistema de arquivos do FreeBSD.

`boot1` é muito simples, este também pode somente ter o tamanho máximo de 512 bytes, e sabemos bastante sobre o *disklabel* do FreeBSD, que armazena informações no *slice*, para procurar e executar o `boot2`.

`boot2` é ligeiramente mais sofisticado, e compreende bastante o sistema de arquivos do FreeBSD para procurar arquivos nele, e pode fornecer uma relação simples para escolher o kernel ou o carregador para executar.

Desde que o [carregador](#) é muito mais sofisticado, e fornece uma configuração fácil-de-usar agradável de inicialização, `boot2` usualmente executa-a, mas previamente isto era tarefa para executar diretamente no kernel.

### Exemplo 7-2. `boot2` Screenshot

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/kernel
boot:
```

Se você precisar sempre substituir a instalação `boot1` e `boot2` use [disklabel\(8\)](#).

```
disklabel -B diskslice
```

Onde *diskslice* é o disco e *slice* da sua inicialização, tal como `ad0s1` para o primeiro HD sobre a IDE.

**Modalidade Perigosamente Dedicada** Se você usar apenas o nome do disco, tal como `ad0`, no comando [disklabel\(8\)](#) você irá criar um disco perigosamente dedicado, sem partições. Isto não deveria ser o que você quer fazer, assim que você verificar o dobro do [disklabel\(8\)](#) depois você deverá pressionar **Return**.

### 7.3.3. Estágio três, `/boot/loader`

A inicialização e o estágio final dos três-estágios de *bootstrap*, é localizado no sistema de arquivos, usualmente como `/boot/loader`.

A inicialização é usada como um método amigável para configuração, usando um jogo de comandos internos fácil-de-usar, voltado por um interpretador mais poderoso, com um jogo de comandos mais complexo.

#### 7.3.3.1. Fluxo de programa do carregador

Durante a inicialização, o carregador sondará para um console e para discos, e figura de saída de que os discos são carregados. Isto setará variáveis conformemente, e um interpretador é iniciado onde os comandos do usuário podem ser passados de um script ou interativamente.

Então o carregador irá ler o `/boot/loader.rc`, que por padrão lê dentro de `/boot/defaults/loader.conf` que seta padrões para variáveis e lê o `/boot/loader.conf` para as mudança destas variáveis locais. `loader.rc` age então sobre estas variáveis, carregando qualquer módulo e kernel que são selecionados.

Finalmente, por padrão, o carregador emite uma espera de 10 segundos para pressionar alguma tecla, e carregar o kernel se este não for interrompido. Se interrompido, ao usuário é apresentado um alerta no prompt que é de fácil uso para entender o comando setado, onde o usuário pode ajustar variáveis, descarregar todos os módulos, carregar módulos, e eles finalmente carregar ou reiniciar.

#### 7.3.3.2. Comandos Carregando e Compilando

Estes são os comandos mais comuns usados no carregamento do sistema. Para uma completa discussão de todas as avaliações dos comandos, por favor veja [loader\(8\)](#).

`autoboot seconds`

O processo de carregamento do kernel não é interrompido durante o tempo concedido, em segundos. Isto mostra uma contagem regressiva, e o tempo padrão concedido é 10 segundos.

`boot [-options] [kernelname]`

Imediatamente o processo de carregamento do kernel, com as opções dadas, se existir, e com o nome do kernel dado, se for.

`boot-conf`

Ir através da mesma configuração automática dos módulos baseados em variáveis como que acontece no boot. Isto somente faz sentido se você primeiramente `descarregar`, e alterar algumas variáveis, do kernel mais freqüente.

`help [tópico]`

Mensagens de ajuda são lidas em `/boot/loader.help`. Se o tópico dado for *index*, então a lista de tópicos disponíveis é dada.

`include filename ...`

Processa o arquivo com o nome dado ao mesmo. O arquivo é lido, e interpretado linha por linha. Um erro imediatamente para o comando de inclusão.

`load [-t type] filename`

Carrega o kernel, o módulo do kernel, ou arquivo do tipo dado, com o nome dado. Todos argumentos depois do nome do arquivo são passados para o arquivo.

`ls [-l] [path]`

Mostra uma lista de arquivos dentro dos caminhos dados, ou o diretório do root, se o caminho não é especificado. Se `-l`, é especificada o tamanho do arquivo será mostrado também.

`lsdev [-v]`

Lista todos os dispositivos que pode ser possível para carregar módulos. Se `-v` for especificado, mais detalhes serão mostrados.

`lsmod [-v]`

Mostra módulos carregados. Se `-v` for especificado, mais detalhes serão mostrados.

`more filename`

Mostra os arquivos especificados, com uma pausa em cada *LINHA* mostrada.

`reboot`

Imediatamente reiniciará o sistema.

`set variable, set variable=value`

Seta as variáveis do ambiente carregado.

`unload`

Remove todos módulos carregados.

### 7.3.3.3. Exemplos de carregador

Estão aqui alguns exemplos práticos do uso do carregador:

- Para simplesmente carregar seu kernel usual, mas em modo mono-usuário:
- `boot -s`
- Para descarregar seu kernel e módulos usuais, e então carregar apenas seu kernel velho (ou outro):
- `unload`
- `load kernel.old`

Você pode usar o `kernel.GENERIC` para referir ao kernel genérico que vem no disco de instalação, ou `kernel.old` para consultar seu kernel previamente instalado (quando você tiver atualizado ou configurado seu kernel, por exemplo).

**Nota:** Use o seguinte para carregar seus módulos usuais com outro kernel:

```
unload
set kernel="kernel.old"
boot-conf
```

- Para carregar um script de configuração do kernel (um script automatizado qual faça as coisas que você faria normalmente dentro do kernel na configuração do boot-time):
- `load -t userconfig_script /boot/kernel.conf`



## 7.4. Interação do Kernel durante o boot

Uma vez que o kernel é carregado pelo carregador [loader](#) (como usado) ou [boot2](#) (passando o carregador), isto examina suas opção de inicialização, se existir, e ajusta seu comportamento como necessário.

### 7.4.1. Opções de inicialização no Kernel

Estão aqui as opção de boot mais comuns:

-a

durante a inicialização o kernel, pede para o dispositivo para montar um arquivo de sistema do root.

-C

boot do CDROM.

-c

rode UserConfig, o configurador do boot-time kernel

-s

boot em modo mono-usuário

-v

ser mais detalhado durante a inicialização do kernel

**Nota:** Existem outras opções de boot, leia [boot\(8\)](#) para maiores informações sobre elas.

---

## 7.5. Dicas de configuração de dispositivos

*Contribuído por Tom Rhodes.*

**Nota:** Esta é uma característica do FreeBSD 5.0 e posteriores que não existe em versões anteriores.

Durante a inicialização do sistema, o *boot loader(8)* irá ler o arquivo [device.hints\(5\)](#). Este arquivo armazena informações de inicialização do kernel conhecidas como variáveis, sobretudo referenciando as vezes ``device hints''. Estas ``device hints'' são usadas pelos drivers do dispositivo para configuraração do dispositivo.

Dicas de configuração de dispositivos podem também ser especificados no [estágio 3 no prompt de inicialização](#). Variáveis podem ser adicionadas usando comando `set`, removidas com o comando `unset`, e visualizadas com o comando `show`. Variáveis setadas dentro do arquivo `/boot/device.hints` podem ser canceladas aqui também. Device hints incorporadas na inicialização não são permanentes e serão perdidas na próxima inicialização.

Uma vez que o sistema é carregado, o comando [kenv\(1\)](#) pode ser usado para obter todas as variáveis.

A sintaxe para o arquivo `/boot/device.hints` é uma variável por linha, usando a mistura padrão ```#"` como marcadores do comentário. Linhas são construídas como segue:

```
hint.driver.unit.keyword="value"
```

A sintaxe para o estágio 3 do *boot loader* é:

```
set hint.driver.unit.keyword=value
```

*driver* é o nome do dispositivo, *unit* é o número do driver do dispositivo, e *keyword* é a sugestão keyword. O keyword pode consistir nas seguintes opções:

- *at*: especifica o dispositivo para o qual é anexado.
- *port*: especifica o endereço de início do I/O para ser usado.
- *irq*: especifica o número da interrupção requerida para ser usado.
- *drq*: especifica o número do canal DMA.
- *maddr*: especifica o endereço físico da memória ocupado pelo dispositivo.
- *flags*: seta variações de bits de opções para o dispositivos.
- *disabled*: Se setada para *1* o dispositivo é desabilitado.

Os drivers do dispositivo podem ou não aceitar (se requerido) mais hints não listadas aqui, é recomendado que veja a página do manual deles. Para maiores informações, consulte as páginas de manuais [device.hints\(5\)](#), [kenv\(1\)](#), [loader.conf\(5\)](#), e [loader\(8\)](#).

---

## 7.6. Init: Inicialização do controle de processos

Uma vez que o kernel termina a inicialização, isto passa para controle dos processos do usuário [init\(8\)](#), que é localizado no `/sbin/init`, ou o caminho do programa especificado em `init_path` variável dentro da inicialização.

### 7.6.1. Sequência automática da Reinicialização

A sequência da reinicialização automática certifica-se que o sistema de arquivos disponíveis sobre o sistema são consistentes. Se eles não são, [fsck\(8\)](#) e não podem

reparar as inconsistências, [init\(8\)](#) deixe o sistema em [modo mono-usuário](#) para que o administrador do sistema tome cuidado dos problemas diretamente.

### 7.6.2. Modo Mono-usuário

Este modo pode ser acessado [automaticamente na sequência da reinicialização](#), ou pelo usuário na inicialização com a opção `-s` ou setando a variável `boot_single` dentro do loader.

Isto pode ser utilizado chamando [shutdown\(8\)](#) sem utilizar as opções de rebotar (`-r`) ou parar (`-h`), do [modo multi-usuário](#).

Se o *console* do sistema for setada para *insecure* em `/etc/ttys`, então o sistema alerta para a senha de `root` antes de inicializar em modo mono-usuário.

#### Exemplo 7-3. Um Console Insecure em `/etc/ttys`

```
name getty type status
comments
#
If console is marked "insecure", then init will ask for the root
password
when going to single-user mode.
console none unknown off insecure
```

**Nota:** Um console *insecure* significa que você considera sua segurança física para a console ser insecure, e quer certificar que somente alguém que saiba a senha de `root` pode usar o modo mono-usuário, e isto não significa que você pode rodar sua console insecurely. Assim, se você quiser segurança, escolha *insecure*, não *secure*.

### 7.6.3. Modo Multi-Usuário

Se [init\(8\)](#) encontra seus arquivos de sistema em ordem, ou uma vez que o usuário terminou em [modo mono-usuário](#), o sistema entra em modo multi-usuário, onde começa a configuração dos recursos do sistema.

#### 7.6.3.1. Configuração de recursos (rc)

O sistema de configuração de recurso lê as configurações padrões do `/etc/defaults/rc.conf`, e o sistema especifica detalhes do `/etc/rc.conf`, e então procede a montagem do sistema de arquivos mencionados em `/etc/fstab`, iniciando os serviços de rede, inicializando vários *daemons* do sistema, e finalmente executando os scripts de inicialização de pacotes instalados localmente.

A página do manual [rc\(8\)](#) é uma boa referência para a configuração de recursos do sistema, como são examinados os scripts.

## 7.7. Sequência do Shutdown

Sob controle o shutdown, via [shutdown\(8\)](#), [init\(8\)](#) tentará rodar o script `/etc/rc.shutdown`, e então proceder o envio de todos os processos de *TERM*, e subsequente do *KILL* também alguns que não terminarem oportunamente.

Para desligar uma máquina FreeBSD sobre arquiteturas de sistemas que suportam gerenciamento de energia, simplesmente use o comando `shutdown -p now` para desligar imediatamente. Para apenas reinicializar o FreeBSD, use `shutdown -r now`. Você precisa ser `root` ou um membro do grupo `operator` para rodar [shutdown\(8\)](#). Os comandos [halt\(8\)](#) e [reboot\(8\)](#) podem até ser usados, consultando por favor seus manuais de referência e [shutdown\(8\)](#) para maiores informações.

**Nota:** Gerenciamento de energia requer suporte [acpi\(4\)](#) no kernel ou carregar algum módulo para FreeBSD 5.X e suporte [apm\(4\)](#) para FreeBSD 4.X.

---

# Capítulo 8. Usuários e Administração Básica de contas

## Índice

- 8.1. [Sinópsse](#)
- 8.2. [Introdução](#)
- 8.3. [O super usuário](#)
- 8.4. [Contas do sistema](#)
- 8.5. [Contas de usuários](#)
- 8.6. [Modificando contas](#)
- 8.7. [Limitando usuários](#)
- 8.8. [Personalizando usuários](#)
- 8.9. [Grupos](#)

*Contrição de Neil Blakey-Milner.*

## 8.1. Sinópsse

O FreeBSD permite que várias pessoas usem o computador ao mesmo tempo. Obviamente, somente um daqueles usuários pode sentar-se na frente da tela e do teclado ao mesmo tempo. [1], mas qualquer número de usuários pode logar-se através de uma rede. Para usar o sistema, cada usuário deve ter uma conta.

Depois de ler este capítulo, você irá saber:

- A diferença entre várias contas de usuários no sistema FreeBSD.
- Como adicionar uma conta de usuário.
- Como remover uma conta de usuário.

- Como alterar os detalhes de uma conta de usuário, como o nome completo do usuário, ou a *shell* preferida.
- Como ajustar os limites básicos de cada conta, para controlar os recursos tais como memória e tempo de uso da CPU, à que usuários e grupos de podem acessar.
- Como usar grupos para facilitar a administração de contas.

Antes de ler este capítulo, você deve:

- Entender o básico de UNIX®, e FreeBSD ([Capítulo 3](#)).

## Notas

- [1] Bem, a não ser que você conectar múltiplos terminais ao sistema, mas iremos abordar isso no [Capítulo 17](#).
- 

## 8.2. Introdução

Todo o acesso ao sistema é feito através das contas, e todos os processos são executados por usuários, dessa forma, administração de contas e usuários são integralmente importantes em sistemas FreeBSD.

Todas as contas em um sistema FreeBSD tem determinadas informações associada à mesma, de forma à identifica-la.

### Nome do usuário

O nome do usuário será digitado no terminal de `login:`. Os nomes de usuários devem ser unicos em um computador, não podendo haver dois usuários com o mesmo nome. Há uma série de regras para criar nomes válidos de usuários, documentadas na [passwd\(5\)](#); você normalmente poderiam ser utilizados nomes de usuários com oito caracteres todos minúsculos.

### Senha

Cada conta tem uma senha associada com ela. A Senha pode ser vazia, neste caso nenhuma senha será solicitada para acessar o sistema. Isso normalmente é uma idéia muito ruim; todas as contas devem ter uma senha.

### ID do Usuário (UID)

O UID é um número que varia de 0 à 65536 usado para identificar de forma única os usuários no sistema. Internamente, o FreeBSD usa o UID para identificar usuários-- qualquer comando do FreeBSD será liberado para um usuário específico convertendo seu UID depois de utilizar o comando. Isso significa que você pode ter contas com diferentes nomes de usuários mas com o mesmo UID. Ambos, para o FreeBSD são um só usuário. É improvável que você tenha necessidade de fazer isso um dia.

## ID de Grupo (GID)

O GID é um número que varia de 0 à 65536, usado para identificar de forma única o grupo primário ao qual o usuário pertence. Os grupos são um mecanismo para controlar o acesso aos recursos baseados em GID ao invés de UID. Isto pode reduzir significativamente o tamanho de arquivos de configurações. Um usuário pode também pertencer a mais de um grupo.

## Classe de Login

Classe de login são uma extensão para o mecanismo de grupos que fornecem flexibilidade adicional para adaptar o sistema à diferentes usuários.

## Tempo de alteração da senha

Por padrão o FreeBSD não força os usuários a trocar as suas senhas periodicamente. Você pode fazer isto para cada usuário, forçando alguns ou todos usuários terem que mudar suas senhas após um determinado período de tempo.

## Tempo de expiração da conta

Por padrão o FreeBSD não expira as contas. Se você estiver criando uma conta que você sabe que tem um tempo limitado de uso, por exemplo, em uma escola onde você tem contas para os estudantes, você pode especificar quando a conta deve expirar. Após a expiração deste tempo o cliente não poderá logar-se no sistema, embora diretórios e arquivos do usuário em questão, permanecerão intocados.

## Nome completo dos usuários

O nome único do usuário identifica a conta do mesmo, internamente, para o FreeBSD, mas não reflete necessariamente o nome real do usuário. Esta informação pode, opcionalmente, ser associada com a conta.

## Diretório *Home*

O Diretório *Home* é o caminho completo para o diretório do sistema em que o usuário logará. Uma convenção comum indica que todos diretórios *home* dos usuários devam ficar sob `/home/usuário`. O usuário armazenaria seus arquivos pessoais em seu diretório *home*, e todos os diretórios que puder criar dentro deste.

## *Shell* do usuário

A *shell* fornece o ambiente padrão para o usuário interagir com o sistema. Existem muitos tipos distintos de *shell*, e os usuários experientes terão suas próprias preferências, que podem ser refletidas junto aos ajustes de cada conta.

Existem três tipos de contas: o [super usuário](#), [usuários do sistema](#), e [usuários](#). A conta ``super usuário'', normalmente chamada de `root`, é usada para administrar o sistema sem limitação de privilégios. Os ``usuários do sistema'' executam serviços. Finalmente, contas de ``usuários'' são usadas por pessoas, que poderão logar-se ao sistema, ler e-mail, e assim por diante.

---

## 8.3. O super usuário

A conta ``super usuário'', normalmente chamada `root`, é usada para configurar e facilitar a administração do sistema, e não deve ser usada para tarefas cotidianas como envio e recebimento de e-mail, exploração geral do sistema, ou para programação.

Isto é porque o super usuário, ao contrário dos usuários normais, podem operar sem limites, e o emprego errado dos privilégios de super usuário pode resultar em um espetacular desastre. As contas de usuários comuns são incapazes de destruir o sistema através de um erro, de forma que é geralmente melhor usar um usuário normal quando possível, a menos que você necessite privilégios a mais.

Você deve sempre dobrar e triplicar a atenção com os comandos que você emite com o super usuário, desde que um espaço extra ou um caracter faltante podem significar a perda irreparável dos dados.

Assim, a primeira coisa que você deve fazer após ter lido este capítulo é criar um usuário sem privilégios para você mesmo para uso geral, se ainda não o tiver. Isto aplica-se igualmente se você está utilizando o sistema em ambiente multi-usuários ou mono-usuário. Mais tarde neste capítulo, nós discutiremos como criar usuários adicionais, e como alternar entre o usuário e o super usuário.

---

## 8.4. Contas do sistema

Contas de ``usuários do sistema'' são utilizadas para executar serviços como DNS, correio, servidores web, e assim por diante. A razão para isso é segurança, se todos os serviços rodassem com os privilégios de super usuários, poderiam agir sem restrições.

Exemplos de usuários de sistema são `daemon`, `operator`, `bind` (para o Serviço de Nomes de Domínio), e `news`. Frequentemente os administradores de sistemas criam a conta `httpd` para rodar um servidor de web.

`nobody` é o usuário genérico sem privilégios do sistema de usuários. Entretanto, é importante manter na sua mente que quanto mais serviços usarem `nobody`, mais arquivos e processos irão associar-se a ele, sendo assim, este usuário se tornará cada vez mais privilegiado, se muitos serviços utilizarem este usuário como padrão.

## 8.5. Contas de usuários

As "contas de usuários" são a primeira etapa para os usuários, pessoas reais, acessarem o sistema. Estas contas isolam o usuário e seu ambiente, impedindo que os mesmos danifiquem o sistema ou prejudiquem outros usuários, e permite que customizem seu ambiente sem afetar outros.

Cada pessoa que acessa seu sistema deve ter uma conta de usuário. Isto permite que você procure quem está fazendo o que, e previna que pessoas alterem configurações de outros usuários, leiam e-mails alheios, e assim por diante.

Cada usuário pode ajustar seu próprio ambiente para acomodar-se ao sistema, para utiliza rum interpretador de comandos alternativo, editor padrão distinto, atalhos de teclados e até mesmo utilizar outra língua além da padrão.

---

## 8.6. Modificando contas

Existe uma variedade de comandos diferentes disponíveis no ambiente UNIX® para manipular contas de usuários. Os comandos mais comuns estão sumariados abaixo, seguidos por exemplos mais detalhados de seu uso.

Comando	Sumário
<a href="#">adduser(8)</a>	O comando recomendado para adicionar novos usuários.
<a href="#">rmuser(8)</a>	O comando recomendado para excluir usuários.
<a href="#">chpass(1)</a>	Uma ferramenta flexível para alterar informações de usuários.
<a href="#">passwd(1)</a>	Um simples comando para alterar a senha do usuário
<a href="#">pw(8)</a>	Uma poderosa e flexível ferramenta para alterar todos os aspectos das contas de usuários.

### 8.6.1. adduser

[adduser\(8\)](#) é um programa simples para adicionar novos usuários. Cria entradas nos arquivos `passwd` e `group`. Criará também um diretório `home` para o usuário novo, copia os arquivos de configuração padrão ("dotfiles") de `/usr/share/skel`, e pode opcionalmente enviar ao usuário novo uma mensagem de boas vindas.

No FreeBSD 5.0, o [adduser\(8\)](#) foi reescrito de rotinas em linguagem Perl para um "shell script" que trabalha em conjunto com o [pw\(8\)](#), assim seu uso é ligeiramente diferente no FreeBSD 4.X e no FreeBSD 5.X.



Para criar a configuração inicial, use `adduser -s -config_create`. [1] Em seguida, configuramos os padrões do `adduser(8)`, e criamos a nossa primeira conta de usuário, uma vez que utilizar a conta `root` para uso normal é um péssimo hábito.

### Exemplo 8-1. Configurando `adduser` e adicionando um usuário no FreeBSD 4.X

```
adduser -v
Use option ``-silent'' if you don't want to see all warnings and
questions.
Check /etc/shells
Check /etc/master.passwd
Check /etc/group
Enter your default shell: csh date no sh tcsh zsh [sh]: zsh
Your default shell is: zsh -> /usr/local/bin/zsh
Enter your default HOME partition: [/home]:
Copy dotfiles from: /usr/share/skel no [/usr/share/skel]:
Send message from file: /etc/adduser.message no
[/etc/adduser.message]: no
Do not send message
Use passwords (y/n) [y]: y

Write your changes to /etc/adduser.conf? (y/n) [n]: y

Ok, let's go.
Don't worry about mistakes. I will give you the chance later to
correct any input.
Enter username [a-z0-9_-]: jua
Enter full name []: J. Usuario Aleatorio
Enter shell csh date no sh tcsh zsh [zsh]:
Enter home directory (full path) [/home/jua]:
Uid [1001]:
Enter login class: default []:
Login group jua [jua]:
Login group is ``jua''. Invite jua into other groups: guest no
[no]: wheel
Enter password []:
Enter password again []:

Name: jua
Password: ****
Fullname: J. Usuario Aleatorio
Uid: 1001
Gid: 1001 (jua)
Class:
Groups: jua wheel
HOME: /home/jua
Shell: /usr/local/bin/zsh
OK? (y/n) [y]: y
Added user ``jua''
Copy files from /usr/share/skel to /home/ua
Add another user? (y/n) [y]: n
Goodbye!
#
```

Em resumo, nós mudamos a shell padrão para **zsh** (uma shell adicional encontrada na coleção de ports), e desligamos o envio de um e-mail de boas vindas para o usuário adicionado. Nós salvamos as configurações, criamos uma conta com o nome de `jua`, e

certificamos que `jua` faz parte do grupo `wheel` (de modo que possa assumir os privilégios do usuário `root` com o comando [su\(1\)](#).)

**Nota:** A senha que você digita não aparece, nem são mostrados asteriscos. Certifique-se que você não digitou duas vezes a senha erroneamente.

**Nota:** Use o [adduser\(8\)](#) sem argumentos de agora em diante, e você não terá que redefinir as opções padrão. Se o programa solicitar que os padrões sejam modificados, saia dele, e tente a opção `-s`.

### Exemplo 8-2. Adicionando um usuário em FreeBSD 5.X

```
adduser
Username: jua
Full name: J. Usuario Aleatorio
Uid (Leave empty for default):
Login group [jua]:
Login group is jua. Invite jua into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh zsh nologin) [sh]: zsh
Home directory [/home/jua]:
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username : jua
Password : ****
Full Name : J. Usuario Aleatorio
Uid : 1001
Class :
Groups : jua wheel
Home : /home/jua
Shell : /usr/local/bin/zsh
Locked : no
OK? (yes/no): yes
adduser: INFO: Successfully added (jua) to the user database.
Add another user? (yes/no): no
Goodbye!
#
```

#### 8.6.2. `rmuser`

Você pode utilizar o [rmuser\(8\)](#) para remover completamente um usuário do sistema. [rmuser\(8\)](#) executa as seguintes etapas:

1. Remova as entradas [crontab\(1\)](#) do usuário (se houver alguma).
2. Remova todos os trabalhos pendentes no [at\(1\)](#), que pertencem ao usuário.
3. Mata todos os processos controlados pelo usuário.
4. Remova o usuário do arquivo de senha local do sistema
5. Remova o diretório home do usuário (se este for de autoridade do usuário)
6. Remova o arquivo de armazenamento de mensagens de correio eletrônico que pertence ao usuário, em `/var/mail`.

7. Remove todos os arquivos de posse do usuário nas áreas temporárias de gravação de arquivos, como `/tmp`.
8. Finalmente, remove o nome do usuário de todos os grupos que ele pertence, em `/etc/group`.

**Nota:** Se um grupo se tornar vazio e o nome do grupo for o mesmo que o nome do usuário, o grupo será removido; isto inclui os grupos originais que é opcionalmente criado para cada usuário pelo [adduser\(8\)](#).

[rmuser\(8\)](#) não pode ser usado para remover contas de super usuários, pois isso aponta indícios de modificações prejudiciais no sistema.

Por padrão, a aplicação funciona em modo interativo, e tenta confirmar que você está certo do que está fazendo.

### Exemplo 8-3. `rmuser` Remoção interativa de contas

```
rmuser jua
Matching password entry:
jua:*:1001:1001:::0:0:J. Usuario Aleatorio:/home/jru:/usr/local/bin/zsh
Is this the entry you wish to remove? y
Remove user's home directory (/home/jua)? y
Updating password file, updating databases, done.
Updating group file: trusted (removing group jua -- personal group is
empty) done.
Removing user's incoming mail file /var/mail/jua: done.
Removing files belonging to jua from /tmp: done.
Removing files belonging to jua from /var/tmp: done.
Removing files belonging to jua from /var/tmp/vi.recover: done.
#
```

### 8.6.3. `chpass`

O [chpass\(1\)](#) modifica informações da base de dados do usuário tais como senhas, interpretadores de comandos e informações pessoais.

Apenas administradores do sistema, como o super usuário, podem alterar informações e senhas de outros usuários com [chpass\(1\)](#).

Quando passado sem opções, com exceção de um usuário opcional, [chpass\(1\)](#) abrirá um editor contendo as informações do usuário. Quando o usuário sair do editor, a base de dados do usuário será atualizada com as novas informações.

**Nota:** No FreeBSD 5.X, você será questionado sobre a senha depois que sair do editor, caso você não seja um super usuário.

### Exemplo 8-4. Usando o `chpass` de forma interativa pelo super usuário

```
#Changing user database information for jua.
Login: jua
Password: *
```

```
Uid [#]: 1001
Gid [# or name]: 1001
Change [month day year]:
Expire [month day year]:
Class:
Home directory: /home/jua
Shell: /usr/local/bin/zsh
Full Name: J. Usuario Aleatorio
Office Location:
Office Phone:
Home Phone:
Other information:
```

O usuário normal pode alterar somente um subconjunto pequeno destas informações, e somente para ele mesmo.

### Exemplo 8-5. Usando o `chpass` de forma interativa por um usuário normal

```
#Changing user database information for jua.
Shell: /usr/local/bin/zsh
Full Name: J. Usuario Aleatorio
Office Location:
Office Phone:
Home Phone:
Other information:
```

**Nota:** [chfn\(1\)](#) e [chsh\(1\)](#) são sinônimos de [chpass\(1\)](#). São links simbólicos para esta aplicação. Da mesma forma, [ypchpass\(1\)](#), [ypchfn\(1\)](#), e [ypchsh\(1\)](#) também o são. O suporte à NIS é automático, de forma que especificar `yp` antes do comando torna-se dispensável. Se isso parece confuso para você, não se preocupe pois o assunto NIS será abordado em [Capítulo 19](#).

#### 8.6.4. `passwd`

O comando [passwd\(1\)](#) é a maneira usual de modificar sua própria senha, ou a senha de um outro usuário, caso você tenha privilégios de administrador.

**Nota:** Os usuários comuns devem digitar sua senha original, antes de definir a nova senha, para impedir que uma pessoa desautorizada modifique-a, quando o verdadeiro dono estiver ausente mas manteve o console aberto.

### Exemplo 8-6. Alterando sua senha

```
% passwd
Changing local password for jua.
Old password:
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

### Exemplo 8-7. Alterando a senha de outro usuário, tendo os privilégios do super usuário

```
passwd jua
```

```
Changing local password for jua.
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

**Nota:** O [chpass\(1\)](#), bem como [yppasswd\(1\)](#), também é um link simbólico para [passwd\(1\)](#), de forma que o NIS possa trabalhar com um ou outro comando.

### 8.6.5. pw

[pw\(8\)](#) é um comando utilitário cuja função é criar, remover, alterar e apresentar usuários e grupos. Ele funciona como um *front end* para os arquivos de usuários e grupos do sistema. O [pw\(8\)](#) tem um conjunto muito completo de opções e comandos que o torna apropriado para o uso em scripts shell, mas os usuários novos podem achá-lo mais complexo do que os outros comandos apresentados aqui.

### Notas

[1] A opção `-s` faz o [adduser\(8\)](#) por padrão ficar em modo silencioso. Usaremos `-v` mais tarde quando quisermos modificar estes padrões.

---

## 8.7. Limitando usuários

A habilidade de limitar seus usuários no sistema pode fatalmente já ter vindo em sua mente. FreeBSD fornece diversas maneiras de um administrador limitar a quantidade de recursos do sistema que um indivíduo pode usar. Estes limites são divididos em duas seções: quotas do disco, e outros limites de recursos.

As quotas de disco limitam o uso do disco aos usuários, e fornecem uma maneira de verificar rapidamente esse uso sem calculá-lo todas as vezes. As quotas são discutidas em [Seção 12.12](#).

Os outros limites de recurso incluem maneiras de limitar a quantidade de uso do CPU, memória e outros recursos que um usuário pode consumir. Estes limites são definidos usando classes de sessão e discutidos aqui.

As classes de login são definidas em `/etc/login.conf`. A semântica precisa está além do escopo abordado nesta seção, mas é descrita em detalhe na página de manual do [login.conf\(5\)](#). É suficiente dizer que cada usuário é relacionado a uma classe de contas, a partir do início de sua sessão (*default*, por padrão), e que cada classe dispõe de um conjunto de recursos que são associados a ela. Uma definição de recurso por seção é composta de um par de opções, no formato *nome =valor*, onde *nome* é um valor previamente conhecido e *valor* é um conjunto de caracteres arbitrário processada de acordo com o *nome*. O ajuste das classe de *login* bem como cada um dos recursos que pode ser controlados, são detalhadamente descritos na página de manual do [login.conf\(5\)](#)

Os limites de recurso são distintos de cada definição de ``login" de uma sessão tradicional em dois pontos. Primeiro, para cada limite existe o limite suave (o atual) e o árduo. O limite suave pode ser definido pelo usuário ou pela aplicação, mas nunca poderá ser maior que o limite árduo. O segundo, pode ser revogado pelo usuário e ter seu valor diminuído, mas nunca pode ser aumentado. Segundo ponto, a maioria destes limites é contada por processo, e não pelo usuário como um todo. Note que, apesar destas diferenças entre a estrutura de recursos e limitações por sessão, este exemplo limitado não ilustra *realmente* o uso efetivo das características de limitação por classe de login.

Dessa forma, seguem abaixo os limites mais comuns, normalmente utilizados em classes de utilização de recursos (as outras opções podem ser encontradas na página de manual do [login.conf\(5\)](#)).

#### *coredumpsize*

Trata-se do limite de tamanho de um arquivo *core* gerado por um programa, por razões óbvias, subordinado a outros limites no uso do disco (ex: *filesize*, ou quota de disco). Não obstante, é usado frequentemente como um método menos severo de controlar o consumo do espaço de disco: uma vez que usuários não geram os arquivos de core conta própria, e logo, também não os deletam (por não saber que existem) configurar esta opção pode evitar que fiquem sem espaço em disco, devido à falha de alguma aplicação com grande demanda por memória por exemplo (ex: **emacs**).

#### *cputime*

Indica a quantidade máxima de tempo do processador central que o processo de um usuário pode consumir. Os processos que desrespeitarem serão mortos pelo kernel.

**Nota:** Este é um limite de *tempo* no processador central, não porcentagem do processador central como indicado do [top\(1\)](#) e [ps\(1\)](#). Limitação por porcentagem, no momento que este documento foi escrito, não é possível, e seria inviável limitar o fato de que um compilador pode facilmente consumir quase 100% de um processador por algum tempo.

#### *filesize*

É o tamanho máximo de um arquivo que um usuário pode possuir. Ao contrário da [quota de disco](#), este limite é imposto a arquivos individuais, e não a todo o conjunto de arquivos que o usuário possui.

#### *maxproc*

Este é o número máximo de processos que um usuário executar simultaneamente. Isto inclui processos e primeiro plano e segundo plano. Por razões óbvias, este valor não pode ser maior do que o limite do sistema,

especificado pela variável [sysctl\(8\)](#) *kern.maxproc*. Note também que configurar este ítem para um valor pequeno vai prejudicar a produtividade do usuário: é frequentemente útil logar-se em múltiplos terminais e executar processos. Algumas tarefas, tais como compilar um programa grande, também geram múltiplos processos (ex: [make\(1\)](#), [cc\(1\)](#) e outros intermediários).

### *memorylocked*

Trata-se da quantidade máxima de memória que um processo pode solicitar que seja reservado - veja [mlock\(2\)](#). Alguns programas críticos do sistema, tais como [amd\(8\)](#), reservam esta memória em espaço de troca (SWAP) não contribuindo com a degradação do sistema em tempo real.

### *memoryuse*

Trata-se da quantidade máxima de memória que um processo pode consumir durante todo seu tempo de execução. Inclui o uso da memória principal e a *swap*. Não trata-se de controle total de restrição de consumo de memória, mas é um bom começo

### *openfiles*

Trata-se da quantidade máxima de arquivos que um processo pode ter aberto. No FreeBSD, arquivos são usados também para representar *sockets* e canais IPC; portanto tenha cuidado para não ajustar este valor demasiadamente baixo. O limite aqui é definido pela [sysctl\(8\)](#) *kern.maxfiles*.

### *sbsize*

Esta é a quantidade máxima de memória a ser utilizada em rede, somatizados em quantos mbufs um usuário pode consumir. Foi originalmente criado em resposta a um velho ataque DoS que criava muitos *sockets*, mas pode usualmente ser usado para limitar a comunicação da rede.

### *stacksize*

Trata-se do tamanho máximo que uma pilha de processos pode vir a ter. Sozinho, este valor não é suficiente para limitar a quantidade de memória que um programa pode usar, consequentemente, deve ser usado em conjunto com outros limites.

Existem outros pontos que devem ser lembrados ao ajustar limites de recursos. A seguir serão apresentados alguns pontos gerais, sugestões, e comentários variados.

- Os processos que são iniciados no sistema pelo */etc/rc* são atribuídos à classe de login *daemon*.
- Embora o */etc/login.conf* que acompanha o sistema seja uma fonte boa de valores razoáveis para a maioria dos limites, apenas você, administrador, pode saber o que é apropriado para o seu sistema. Ajustar um limite demasiadamente

elevado pode abrir seu sistema até o abuso, enquanto o ajustar demasiadamente baixo pode colocar a produtividade em tensão.

- Os usuários do sistema de janela X (X11) devem provavelmente ter direito de utilizarem mais recursos que outros usuários. O X11 por si só usa poucos recursos, mas incentiva os usuários à rodarem programas simultâneos.
- Recorde que muitos limites se aplicam aos processos individuais, não usuário como um todo. Por exemplo, ajustar *openfiles* para 50 significa que cada processo que o usuário rodar, pode abrir até 50 arquivos. Assim, a quantidade bruta de arquivos que um usuário pode abrir é o valor *openfiles* multiplicado pelo *maxproc*. Isto também se aplica ao consumo de memória.

Para mais informações sobre limites de recursos e classes de login em geral, consulte por favor as páginas de manuais relevantes: [cap\\_mkdb\(1\)](#), [getrlimit\(2\)](#), [login.conf\(5\)](#).

---

## 8.8. Personalizando usuários

A localização é um ambiente ajustado pelo administrador ou pelo usuário do sistema para oferecer línguas diferentes, conjunto de caracteres, data e padrões de tempo diferentes, e assim por diante. Isto é discutido no capítulo sobre [localização](#).

---

## 8.9. Grupos

Um grupo é simplesmente uma lista de usuários. Os grupos são identificados por nome e GID (ID de Grupo). No FreeBSD (e na maioria de outros sistemas UNIX®), os dois fatores que o kernel usa para decidir se um processo está tem permissão para algo é seu ID de usuário e a lista dos grupos à que ele pertence. Diferente de um ID de usuário, um processo tem uma lista dos grupos associados à ele. Você eventualmente pode precisar consultar o "ID de grupo" de um usuário ou de um processo; na maioria das vezes seus direitos implicam apenas no primeiro grupo da lista.

O nome do grupo para ser identificado no mapa de ID de grupos está em `/etc/group`. Trata-se de um arquivo de texto plano com quatro campos delimitadores. O primeiro campo é o nome do grupo, o segundo é a senha encriptada, o terceiro é o ID de grupo, e o quarto é a lista de membros daquele grupo, delimitado por vírgula. Pode seguramente ser editado na mão (supondo naturalmente, que você não vá cometer erros de sintaxe!). Para uma descrição mais completa da sintaxe, veja a página de manual [group\(5\)](#).

Se você não quiser editar manualmente o `/etc/group`, pode utilizar o comando [pw\(8\)](#) para adicionar e editar grupos. Por exemplo, para adicionar um grupo chamado `timedois` e em seguida confirmar que ele foi criado, você pode usar:



### Exemplo 8-8. Adicionando um grupo por meio do pw(8)

```
pw groupadd timedois
pw groupshow timedois
timedois:*:1100:
```

O número *1100* acima identifica o ID do grupo *timedois*. Agora, *timedois* não tem nenhum membro, e portanto é inútil. Vamos modificar aquele usuário *jua* e coloca-lo no grupo *timedois*.

### Exemplo 8-9. Adicionando alguém a um determinado grupo por meio do pw(8)

```
pw groupmod timedois -M jua
pw groupshow timedois
timedois:*:1100:jua
```

A opção *-M* deve ser uma lista de usuários que serão membros do grupo em questão, delimitados por vírgula. Com base nas outras seções deste capítulo, sabemos que o arquivo de senha também contém o grupo de cada usuário. Se o usuário fizer parte apenas de um grupo, o valor identificado no campo em questão é válido. Do contrário o sistema inclui automaticamente o usuário à lista de membro de cada grupo, e o grupo primário não será apresentado com o argumento *groupshow* do [pw\(8\)](#), ou qualquer ferramenta similar. Ou seja [pw\(8\)](#) manipula somente o arquivo */etc/group*; nunca tentará ler também dados do */etc/passwd*.

### Exemplo 8-10. Usando o id(1) para determinar grupos associados aos usuários

```
% id jua
uid=1001(jua) gid=1001(jua) groups=1001(jua), 1100(timedois)
```

Como pode-se notar, *jua* é membro dos grupos *jua* e *timedois*.

Para maiores informações sobre o [pw\(8\)](#), consulte sua página de manual e para maiores informações sobre o formato do */etc/group*, consulte a página de manual [group\(5\)](#).

---

## Capítulo 9. Configurando o *kernel* do FreeBSD

### Índice

- 9.1. [Sinópse](#)
- 9.2. [Porque compilar um \*kernel\* customizado?](#)
- 9.3. [Compilando e instalando um \*kernel\* customizado](#)
- 9.4. [O arquivo de configuração](#)
- 9.5. [Criando Dispositivos de Controle](#)
- 9.6. [Se algo der errado](#)

*Atualizado e reestruturado por Jim Mock. Contribuição original de Jake Hamby.*

## 9.1. Sinótese

O *kernel* é o coração do sistema FreeBSD. Ele é responsável por gerenciar a memória, reforçar controles de segurança, pela rede, pelo acesso a disco, e muito mais. À medida que mais e mais o FreeBSD torna-se dinamicamente configurável, ocasionalmente faz-se necessário configurar e recompilar seu *kernel*.

Depois de ler este capítulo, você saberá:

- Porque você poderá precisar compilar um *kernel* customizado.
- Como escrever um arquivo de configuração do *kernel*, ou alterar um arquivo de configuração existente.
- Como usar o arquivo de configuração do *kernel* para criar e compilar um novo *kernel*.
- Como instalar o novo *kernel*.
- Como criar quaisquer entradas no `/dev` caso isso venha a ser necessário.
- Como resolver problemas se algo der errado.

---

## 9.2. Porque compilar um *kernel* customizado?

Tradicionalmente, o FreeBSD teve o que foi chamado de *kernel* "monolítico". Isto significa que o *kernel* era um grande programa, contendo uma lista fixa de dispositivos suportados, e se você quisesse alterar seu comportamento então seria necessário compilar um novo *kernel*, e então reiniciar seu computador com ele.

Hoje, o FreeBSD está movendo-se rapidamente para um modelo onde muitas das funcionalidades do *kernel* estão contidas em módulos que são carregados e descarregados dinamicamente conforme eles são necessários. Isto permite que o *kernel* imediatamente se adapte a um novo *hardware* tornando-o disponível (tais como cartões PCMCIA para *laptop*), ou a novas funcionalidades à serem inseridas no *kernel* que não eram necessárias quando o ele foi originalmente compilado. Isso é conhecido como *kernel* modular. Coloquialmente eles são chamados KLDs.

Apesar disto, ainda é necessário carregar algumas configurações estáticas no *kernel*. Em alguns casos isto está assim porque estas funcionalidades estão tão amarradas ao *kernel* que elas não podem ser carregadas dinamicamente. Em outros casos, isto pode simplesmente estar assim porque ninguém ainda conseguiu tempo para escrever um módulo de carga dinâmico do *kernel* para cada uma delas.

Compilar um *kernel* personalizado é um dos rituais mais importantes que cada usuário de UNIX® deve enfrentar. Este processo, embora consuma tempo, trará muitos benefícios para o seu sistema FreeBSD. Diferentemente do *kernel* GENERIC, o qual necessita de suporte a uma abrangente lista de equipamentos, um *kernel* personalizado

somente conterà suporte para o *hardware* do seu PC. Isto tem inúmeros benefícios, tais como:

- Tempo de inicialização mais rápido. Visto que o *kernel* apenas testará o equipamento que você tiver em seu sistema, o tempo de testes ao inicializar a máquina diminuirá drasticamente.
- Menor uso da memória. Um *kernel* personalizado frequentemente utiliza menos memória que o *kernel* `GENERIC`, o que é importante porque o *kernel* deve sempre estar presente na memória física. Por esta razão, um *kernel* personalizado é especialmente útil em um sistema com pouca quantidade de RAM.
- Suporte adicional de *hardware*. Um *kernel* personalizado permite a você adicionar suporte a dispositivos, tais como placas de som, que não estão presentes no *kernel* `GENERIC`.

---

## 9.3. Compilando e instalando um *kernel* customizado

Primeiramente, deixe-nos apresentar rapidamente o diretório de compilação do *kernel*. Todos os diretórios mencionados serão relativos ao diretório principal, `/usr/src/sys`, que também poderá ser acessado diretamente através de `/sys`. Há um número de subdiretórios que representam partes diferentes do *kernel*, mas os mais importantes, para nossos objetivos, são `arch/conf`, onde você editará sua configuração de *kernel* personalizado, e `compile`, que é a área de preparação onde seu *kernel* irá ser compilado. `arch` representa uma das arquiteturas suportadas pelo FreeBSD como `i386`, `alpha`, ou `pc98` (uma ramo de desenvolvimento alternativo de *hardware* do PC, popular no Japão). Tudo que está dentro de um diretório de uma arquitetura específica diz respeito somente a esta arquitetura; o resto do código do FreeBSD é comum para todas as plataformas às quais ele poderá ser potencialmente convertido. Observe a organização lógica da estrutura de diretório, com cada dispositivo suportado, sistema de arquivos, e opções em seu próprio subdiretório. As versões do FreeBSD 5.X e superiores tem suporte para `sparc64`, e algumas outras arquiteturas em desenvolvimento.

**Nota:** Se não existir o diretório `/usr/src/sys` no seu sistema, então o fonte do *kernel* não foi instalado. A maneira mais fácil para fazer isso é rodando `/stand/sysinstall` como `root`, escolher `Configure`, `Distributions`, `src`, e então `sys`. Se você tiver alguma aversão ao `sysinstall` e possuir acesso a um FreeBSD CDROM "oficiais", então você poderá também instalar o fonte através da linha de comando:

```
mount /cdrom
mkdir -p /usr/src/sys
ln -s /usr/src/sys /sys
cat /cdrom/src/ssys.[a-d]* | tar -xzvf -
```

Depois, vá até o diretório `arch/conf` e copie o arquivo de configuração `GENERIC` com o nome que você quiser para o seu *kernel*. Por exemplo:

```
cd /usr/src/sys/i386/conf
```

```
cp GENERIC MYKERNEL
```

Tradicionalmente, este nome tem todas letras em maiúsculas, se você mantém múltiplas máquinas FreeBSD com diferentes *hardware*, é uma boa idéia nomeá-lo prefixando o *hostname* da sua máquina. Para a finalidade deste exemplo, será usado o nome MYKERNEL.

**Dica:** Guardar os arquivos de configuração do seu *kernel* diretamente abaixo do `/usr/src` pode ser uma má idéia. Se você tiver problemas poderá ser tentado a apenas apagar o `/usr/src` e iniciar tudo novamente. Cinco segundos depois disto feito, você poderá perceber que removeu o seu arquivo de configuração de *kernel* personalizado. Não edite diretamente o arquivo `GENERIC`, pois ele pode ser sobrescrito da próxima vez que você for [atualizar a árvore do código fonte](#); suas modificações do *kernel* serão perdidas.

Você pode querer manter em outro local seu arquivo de configuração do *kernel*, e então criar um link simbólico deste arquivo para o diretório `i386`.

Por exemplo:

```
cd /usr/src/sys/i386/conf
mkdir /root/kernels
cp GENERIC /root/kernels/MYKERNEL
ln -s /root/kernels/MYKERNEL
```

**Nota:** Você deverá executar todos estes comandos sob a conta do `root` senão ocorrerão erros de permissão negada.

Agora, edite o `MYKERNEL` com seu editor de texto favorito. Se você for começar a configurar, provavelmente somente o `vi` esteja disponível, o qual é muito complexo para explicar aqui, mas é coberto em muitos livros indicados na [bibliografia](#). Entretanto, o FreeBSD oferece um editor mais fácil chamado `ee` que, se você for um novato, deve ser o editor de sua escolha. Sinta-se livre para mudar as linhas de comentários, principalmente para identificar sua configuração e as alterações que diferenciam seu arquivo do `GENERIC`.

Se você tiver que compilar um kernel sobre SunOS™ ou qualquer outro Sistema Operacional BSD, muito destes arquivos serão muito familiar a você. Se você vier de outro sistema operacional tal como DOS, o arquivo de configuração `GENERIC` pode impressionar você, sendo assim siga as descrições do [Arquivo de Configuração](#) lentamente e com cuidado.

**Nota:** É importante você sempre verificar o arquivo `/usr/src/UPDATING`, antes de executar todas as etapas da atualização, No caso você pode [sincronizar sua árvore de código](#) com o código fonte do último projeto do FreeBSD. Neste arquivo todas as importantes alterações para atualização do FreeBSD são digitadas. Ajuste sempre sua versão do código fonte do FreeBSD de acordo com `/usr/src/UPDATING`, pois estas notas são mais precisas para novas informações que o handbook.

Você deve agora compilar o código fonte do kernel. Estão aqui dois procedimentos que você pode usar para isto, e qual você usará, vai depender de como quer recompilar seu kernel, e a versão do FreeBSD que você está rodando.

- Se você tiver instalado *somente* o código fonte do kernel, use o procedimento 1.
- Se você tiver rodando um FreeBSD versão anterior ao 4.0, e você *não* tiver atualização para o 4.0 ou mais atual use o procedimento `make world`, procedimento 1.
- Se você estiver recompilando um kernel novo sem atualizar o código fonte (talvez para adicionar uma nova opção, tal como *IPFIREWALL*) você teria que usar o outro procedimento.
- Se você estiver recompilando o kernel como parte de um processo de um `make world`, use o procedimento 2.

### Procedimento 1. Compilando um Kernel da Maneira ``Tradicional''

1. Execute [config\(8\)](#) para gerar o código fonte do kernel.
2. `# /usr/sbin/config MYKERNEL`
3. Mude o diretório de configuração. Isto é mostrado após rodar o comando acima mencionado.
4. `# cd ../compile/MYKERNEL`

Para FreeBSD anterior a 5.0, usa-se preferivelmente:

```
cd ../../compile/MYKERNEL
```

5. Compilando o kernel.
6. `# make depend`
7. `# make`
8. Instalando o novo kernel.
9. `# make install`

### Procedimento 2. Compilando um Kernel da ``Nova'' maneira

1. Mude de diretório para `/usr/src`.
2. `# cd /usr/src`
3. Compilando o kernel.
4. `# make buildkernel KERNCONF=MYKERNEL`
5. Instalando o novo kernel.
6. `# make installkernel KERNCONF=MYKERNEL`

**Nota:** No FreeBSD 4.2 e anteriores troque `KERNCONF=` por `KERNEL=`. Após o 4.2-STABLE esta opção já foi atualizada, e antes de Fevereiro de 2001 o sistema não reconhece `KERNCONF=`.

Se você *não* atualizou seu código de alguma maneira (as formas possíveis incluem **CVSup**, **CTM**, ou **anoncvs**), então deve-se usar os comandos `config`, `make depend`, `make`, `make install` nesta sequência.

O novo kernel será copiado para o diretório do root como `/kernel` e o velho kernel será movido para `/kernel.old`. Agora, desligue o sistema e reinicie-o, para carregar seu

novo kernel. Caso algo saia errado, há algumas instruções para [resolver problemas](#) no final deste capítulo. É interessante que você leia a seção de como recuperar o sistema caso seu novo kernel [não carregue](#).

**Nota:** No FreeBSD 5.0, o kernel é instalado junto com os módulos em `/boot/kernel`, e os kernel velhos são colocados em `/boot/kernel.old`. Outros arquivos relacionados ao processo de carregamento, tal como o carregamento do sistema por meio do [loader\(8\)](#) e suas configurações são armazenadas também em `/boot`. Módulos personalizados de terceiros normalmente são colocados em `/boot/modules`, embora os usuários devam estar cientes que manter os módulos em sincronização com o kernel compilado é muito importante. Se os módulos pretendidos não rodarem com o kernel compilado, isso pode resultar em instabilidade ou inconformidades no sistema.

**Nota:** Se você adicionar qualquer dispositivo novo (tal como Placa de Som) e estiver rodando FreeBSD 4.X ou versões anteriores, pode ser necessário adicionar algum dispositivo de controle em seu diretório `/dev` antes que eles possam ser utilizados corretamente. Para mais informações, dê uma olhada na seção [Criando dispositivos de controle](#) posteriormente neste mesmo capítulo.

---

## 9.4. O arquivo de configuração

O formato geral do arquivo de configuração é bastante simples. Cada linha contém uma palavra-chave e mais algum argumento. Para simplificar, a maioria das linhas contém somente um argumento. Qualquer coisa que segue um `#` é considerado um comentário e é ignorado. As seguintes seções descrevem cada palavra-chave, geralmente na ordem que são listadas no `GENERIC`, embora algumas palavras-chaves relacionadas sejam agrupadas em uma única seção (tal como `rede`) mesmo que sejam descartados durante todo o arquivo `GENERIC`. Uma lista completa de opções mais detalhadas, a explicação de cada um dos dispositivos presentes no arquivo de configuração do kernel, estão disponíveis no arquivo `LINT`, localizado no mesmo diretório do `GENERIC`. Se você tiver na dúvida a respeito da necessidade de uma linha, verifique primeiro no `LINT`.

**Nota:** No FreeBSD 5.X e superiores o `LINT` não existe. Veja o arquivo `NOTES` para opções dependentes de arquitetura. Algumas opções, principalmente independentes da arquitetura, são armazenadas no arquivo `/usr/src/sys/conf/NOTES`. É aconselhável revisar as opções dispostas neste arquivo também.

A seguir um exemplo de arquivo de configuração `GENERIC` do kernel com vários comentários adicionais, onde eles forem necessários, para esclarecimentos. Este exemplo deve se equivaler uma cópia que você tem disponível em `/usr/src/sys/i386/conf/GENERIC`. Para obter detalhes sobre todas as opções possíveis do kernel, veja `/usr/src/sys/i386/conf/LINT`.

```

#
GENERIC -- Generic kernel configuration file for FreeBSD/i386
#
For more information on this file, please read the handbook section
on
Kernel Configuration Files:
#
http://www.FreeBSD.org/doc/en_US.ISO8859-
1/books/handbook/kernelconfig-config.html
#
The handbook is also available locally in /usr/share/doc/handbook
if you've installed the doc distribution, otherwise always see the
FreeBSD World Wide Web server (http://www.FreeBSD.org/) for the
latest information.
#
An exhaustive list of options and more detailed explanations of the
device lines is also present in the ../../conf/NOTES and NOTES
files.
If you are in doubt as to the purpose or necessity of a line, check
first
in NOTES.
#
$FreeBSD: src/sys/i386/conf/GENERIC,v 1.380 2003/03/29 13:36:41
mdodd Exp $

```

Em seguida estão as palavra-chaves obrigatoriamente necessárias em *todos* os kernels que você for compilar:

```
machine i386
```

Está é a arquitetura da máquina. Deve ser algum dos seguintes valores: *i386*, *pc98*, *sparc64*, *alpha*, *ia64*, *amd64*, or *powerpc*.

```
cpu I486_CPU
cpu I586_CPU
cpu I686_CPU
```

A opção acima especifica o tipo da CPU em seu sistema. É possível manter múltiplas entradas desta definição, em casos onde não souber ao certo que classes de CPU devem ser suportadas em seu kernel, nesse caso, as opções escolhidas são *I586\_CPU* ou *I686\_CPU*, entretanto, para personalizar um kernel, é melhor especificar apenas a CPU que você dispõe, quando esta puder ser prevista. Se você estiver inseguro quanto à classe da sua CPU, pode verificar o arquivo `/var/run/dmesg.boot` para verificar as mensagens do carregamento do sistema.

O suporte à *I386\_CPU* ainda está disponível no fonte do FreeBSD, mas esta opção agora é desativada por padrão em ambos `-STABLE` e `-CURRENT`. Isto significa que para instalar o FreeBSD com um processador de classe 386, você terá que optar por uma das seguintes alternativas:

- Instale uma versão mais antiga do FreeBSD e recompile o sistema como descrito na [Seção 9.3](#).
- Compile as aplicações em nível de usuário e o kernel em outra máquina e instale-as no 386 em questão, usando os arquivos previamente compilados, armazenados em `/usr/obj` (para mais detalhes veja a [Seção 21.5](#)).

- Gere sua própria versão do FreeBSD incluindo suporte a *I386\_CPU* no kernel de instalação via CD-ROM.

A primeira destas opções é provavelmente a mais fácil, mas você necessitará de muito espaço em disco em uma máquina de classe 386, o que pode ser difícil de encontrar.

```
ident GENERIC
```

Está é a identificação do kernel. Você pode mudar o nome do seu kernel para qualquer um, mas como exemplo da configuração do nosso kernel customizado, o *MYKERNEL*, você deve acompanhar as instruções dos próximos exemplos. O nome que você colocou no *ident* serão mostrados quando o kernel for carregado, de forma que a opção é utilizada para dar um nome novo ao kernel, ou um nome diferente se você quiser distinguir seu novo kernel do kernel atual (especialmente útil quando quiser compilar um kernel experimental).

```
maxusers n
```

A opção *maxusers* ajusta uma série de tabelas importantes do sistema. Esta opção não deve ser utilizada de forma a se igualar ao número de usuários simultâneos esperados em sua máquina.

Desde o FreeBSD 4.5, o sistema ajusta automaticamente esta configuração, se seu valor for modificado para *0*[\[1\]](#). No FreeBSD 5.X, *maxusers* é definido por padrão com o valor *0* quando for omitido. Se sua versão do FreeBSD for anterior ao 4.5, ou você deve definir esta variável manualmente sempre, e o valor de *maxusers* deve ser no mínimo 4, especialmente se for usando o Sistema de Janelas X, ou compilando programas com frequência. A razão é que a tabela mais importante ajustada por meio do *maxusers* é o número máximo de processos, que é definido como  $20 + 16 * maxusers$ , de forma que, se você ajustar *maxusers* para o valor 1, poderão existir no máximo 36 processos simultâneos, incluindo aí os 18 processos que o sistema carrega na hora da inicialização, e os 15 que você provavelmente irá criar quando iniciar o sistema de interface gráfica. Mesmo tarefas simples como a leitura de uma página de manual será iniciada levantando nove processos para filtro, decompactação e visualização do manual. Colocando *maxusers* em 64 permitirá que você tenha 1044 processos simultâneos carregados, o que deve ser o suficiente para a maioria de suas necessidades. Entretanto, se você se deparar com o erro `proc table is full` ao tentar iniciar outro programa, ou se o sistema em questão tratar de um servidor com um grande número de usuários simultâneos (digamos, como o `ftp.FreeBSD.org` por exemplo), é possível aumentar o valor da variável *maxusers* e recompilar o kernel do sistema.

**Nota:** *maxusers* não limita o número de usuários que podem logar em sua máquina. Ele apenas define o valor de várias tabelas à quantias razoáveis considerando o número máximo de usuários e processos que você provavelmente terá rodando em seu sistema. A opção que possibilita limitar o número de logins *remotos simultâneos e de terminais gráficos* é [pseudo-device pty 16](#).

```
Floating point support - não desabilite.
device npx0 at nexus? port IO_NPX irq 13
```



*npx0* é a opção que disponibiliza suporte lógico à ponto flutuante em unidades matemáticas do FreeBSD, que faz o papel de equipamento co-processador, ou simplesmente emulador matemático lógico. Esta definição *não* é opcional.

```
Pseudo devices - o número indica quantas unidades serão alocadas.
pseudo-device loop # Network loopback
```

Trata-se do dispositivo genérico de comunicação TCP/IP interna - via *loopback*. Por exemplo, se você fizer uma conexão telnet ou FTP para localhost (ou seja, 127.0.0.1) a conexão voltará à você através deste pseudo-dispositivo. Este comportamento é mandatório.

Todas as outras opções no decorrer do arquivo são consideravelmente optativas. Veja as notas em cada uma das opções abaixo, para obter mais informações.

```
#To statically compile in device wiring instead of /boot/device.hints
#hints "GENERIC.hints" #Default places to look for devices.
```

No FreeBSD 5.X e versões posteriores, o [device.hints\(5\)](#) é usado para configurar algumas opções de dispositivos de controle. Por padrão o [loader\(8\)](#) verifica as opções em `/boot/device.hints`. Usando a opção *hints* estas definições podem ser estaticamente compiladas dentro do seu kernel, de forma que a necessidade de criar o arquivo `device.hints` dentro de `/boot` seja suprimida.

```
#makeoptions DEBUG=-g #Build kernel with gdb(1) debug symbols
```

O processo normal de compilação do FreeBSD não inclui informações sobre problemas com a compilação do kernel, e remove a maioria dos símbolos depois que o kernel é instalado, para economizar algum espaço em disco. Caso você esteja fazendo testes com kernel na série -CURRENT, ou alterando opções de desenvolvimento de seu próprio kernel FreeBSD, pode ser interessante descomentar esta linha. Isto irá habilitar o uso da opção *-g* que habilita a adição de informações de depuração do programa, realizados pelo [config\(8\)](#), caso você esteja compilando seu kernel da maneira ``tradicional" (veja a [Seção 9.3](#) para obter mais informações.).

```
options MATH_EMULATE #Support for x87 emulation
```

Esta linha permite que o kernel simule um co-processador matemático se seu computador não tiver um disponível (386 ou 486SX). Caso seu PC seja um 486DX, ou um 386 ou 486SX (com um chip 387 ou 487 separado), ou superior (Pentium®, Pentium II, etc.), esta opção poderá ser comentada.

**Nota:** As rotinas normais para emulação matemáticas do co-processador que vem com o FreeBSD *não* são muito precisas. Se você não dispõe de um co-processador matemático mas tem necessidade de uma emulação mais precisa, substitua a opção anterior por *GPL\_MATH\_EMULATE*, de forma a usar o suporte à co-processamento matemático GNU que não é habilitado por padrão em razão de questões legais da licença GPL.

No FreeBSD 5.X, a emulação matemática é desabilitada por padrão, em outras CPUs que não tem suporte a ponto flutuante matemático nativo, a emulação é menos comum,

e em muitos casos não são sequer suportados pelo kernel `GENERIC` sem que outras opções adicionais sejam incluídas.

```
options INET #InterNETworking
```

Suporte a Rede. Mantenha esta opção ainda que você não planeje se conectar a uma rede. Muitos programas tem necessidade mínimas que necessitam de suporte à rede, como o uso da interface de *loopbacl* (isto é, conexões internas de rede), de forma que esta opção é mandatória.

```
options _INET6 #IPv6 communications protocols
```

Trata-se do protocolo de comunicação IPv6.

```
options FFS #Berkeley Fast Filesystem
options FFS_ROOT #FFS usable as root device [keep this!]
```

Trata-se do sistema de arquivos essencial do FreeBSD. Mantenha esta opção se você planeja controlar algum disco rígido.

**Nota:** À partir do FreeBSD 5.X, *FFS\_ROOT* não é mais necessário.

```
options UFS_ACL #Support for access control lists
```

Esta opção, está presente apenas no FreeBSD 5.X; ela habilita o suporte do kernel à listas de controle de acesso. Seu uso requer as opções de atributos estendidos e UFS2, e os detalhes são descritos na [Seção 10.13](#). ACLs são habilitadas por padrão, e não devem ser desabilitadas no kernel se tiverem sido previamente utilizadas no sistema de arquivos, pois removerá a lista de controle de acessos, modificando os atributos dos arquivos protegidos para valores imprevisíveis.

```
options UFS_DIRHASH #Improve performance on big directories
```

Esta opção inclui recursos para aumentar a velocidade de operações de disco em diretórios grandes, ao custo de usar um pouco de memória adicional. Normalmente é interessante usufruir desta opção em um grande servidor, ou em situações onde haja grande demanda de interação com estações de trabalho; remova esta opção quando estiver usando o FreeBSD em um pequeno sistema onde memória a é prioritária e velocidade de acesso a disco é menos importante, como um firewall.

```
options SOFTUPDATES #Enable FFS Soft Updates support
```

Esta opção habilita o método *Soft Updates* no kernel, isto irá ajudar a aumentar a velocidade de escrita ao disco. Mesmo sendo habilitada no kernel, estes recursos ainda precisam ser atribuídos à cada partição de disco. Revise a saída do comando [mount\(8\)](#) para obter detalhes sobre que sistemas de arquivos estão com *Soft Updates* habilitados. Caso não esteja observando a opção, é necessário ativa-la usando o [tunefs\(8\)](#) (para sistemas de arquivos existentes) ou [newfs\(8\)](#) (para novo sistema de arquivos).

```
options MFS #Memory Filesystem
options MD_ROOT #MD is a potential root device
```

Trata-se basicamente de um espaço em memória mapeado para armazenamento rápido de arquivos temporários, se você tiver uma boa quantidade de memória real e virtual, pode forçar a utilização deste recurso em outros pontos também. Uma idéia interessante é montar como MFS o diretório `/tmp`, se houver necessidade de muito desempenho nesta unidade. Neste caso, adicione por exemplo a seguinte linha no arquivo

```
/etc/fstab:
```

```
/dev/ad1s2b /tmp mfs rw 0 0
```

Agora basta reiniciar seu sistema ou rodar o comando `mount /tmp`.

**Nota:** No FreeBSD 5.X, a opção [md\(4\)](#) é usada com grande vantagem sobre MFS. Mais informações sobre a configuração de sistemas de arquivo em memória, podem ser obtidas nos manuais [mdconfig\(8\)](#) e [mdmfs\(8\)](#), e também na [Seção 12.10](#). Consequentemente, a opção *MFS* não é mais suportada na série 5.X.

```
options NFS #Network Filesystem
options NFS_ROOT #NFS usable as root device, NFS required
```

Trata-se do sistema de arquivos de rede. A menos que você não planeje montar partições UNIX® sob rede TCP/IP, comente esta linha.

```
options MSDOSFS #MSDOS Filesystem
```

Trata-se do sistema de arquivos MS-DOS®. Se você não pretende montar partições formatadas em DOS, pode seguramente retirar esta opção. Se você tiver necessidade de montar sistemas DOS posteriormente, o módulo será automaticamente carregado. O ótimo conjunto de programas **mtools** (disponível na Coleção de Ports) permite que você acesse disquetes DOS sem ter que montá-los e desmontá-los (e não requer *MSDOSFS*).

```
options CD9660 #ISO 9660 Filesystem
options CD9660_ROOT #CD-ROM usable as root, CD9660 required
```

O sistema de arquivos ISO 9660 é utilizado em CDRoms. Pode ser interessante comentar esta opção se você não tem um dispositivo óptico de leitura de CDRom, ou se tem a necessidade ocasional de montar CDs de dados (uma vez que o módulo correspondendo pode ser carregado dinamicamente quando for necessário montar um CD de dados). CDs de áudio não necessitam deste suporte.

```
options PROCFS #Process filesystem
```

Trata-se do sistema de arquivo de processos. Este sistema de arquivo tem a função de criar e montar o ambiente `/proc` que permite o uso de programas como [ps\(1\)](#) para você ter mais informações sobre quais processos está rodando. No FreeBSD 5.X, o uso do *PROCFS* não é mais necessário por vários motivos, visto que a maioria das ferramentas de monitoramento tem que ser adaptadas para rodar sobre *PROCFS*. Em adição, o kernel 5.X-CURRENT que usa *PROCFS* deve agora também incluir suporte para *PSEUDOSFS*:

```
options PSEUDOSFS #Pseudo-filesystem framework
```

*PSEUDOS* não está disponível sob FreeBSD 4.X. Ao contrário do FreeBSD 4.X, as novas instalações do FreeBSD 5.X não montam, por padrão, o sistema de arquivos de processos.

```
options COMPAT_43 #Compatible with BSD 4.3 [KEEP THIS!]
```

Compatibilidade com BSD 4.3. Deixe isto descomentado, alguns programas agirão estranhamente se você comentar isto.

```
options COMPAT_FREEBSD4 #Compatible with FreeBSD4
```

Esta opção é necessária no FreeBSD 5.X i386™ e Sistemas Alpha para suporte à aplicativos compilados em outras versões do FreeBSD, que usa outro sistema de interfaces de chamadas. Recomenda-se que esta opção seja usada sobre todos i386 e sistemas Alpha que podem rodar outros aplicativos; plataformas que ganham suporte somente em 5.X, tal como ia64 e Sparc64®, não tem esta necessidade.

```
options SCSI_DELAY=15000 #Delay (in ms) before probing SCSI
```

Isso faz com que o kernel pare por 15 segundos antes de procurar pelos dispositivos SCSI em seu sistema. Se você dispor apenas de discos IDE, pode ignorar esta opção; em discos SCSI, pode ser interessante baixar à até 5 segundos, para acelerar a inicialização. Naturalmente, se depois disso o FreeBSD tiver problemas para reconhecer algum dispositivo SCSI, suba este valor novamente.

```
options UCONSOLE #Allow users to grab the console
```

Permite que a console usada por aplicações gráficas sejam acessadas por outros usuários. Por exemplo, se você criar um terminal **xterm** com `xterm -C`, tudo que estiver sendo feito pelo [write\(1\)](#), [talk\(1\)](#), e todas as outras mensagens que você receber, assim como algumas mensagens de console emitidas pelo kernel, serão capturadas.

**Nota:** No FreeBSD 5.X, *UCONSOLE* não é mais necessária.

```
options USERCONFIG #boot -c editor
```

Esta opção permite carregar e editar as configurações do menu de inicialização.

```
options VISUAL_USERCONFIG #visual boot -c editor
```

Esta opção permite carregar e editar as configurações do menu de inicialização em modo gráfico.

**Nota:** À partir do FreeBSD 5.0, a opção *USERCONFIG* foi substituída pelo novo método [device.hints\(5\)](#). Para maiores informações sobre [device.hints\(5\)](#) por gentileza, visite a [Seção 7.5](#).

```
options KTRACE #ktrace(1) support
```

Esta opção permite rastrear os processos do kernel, que são usados em rotinas de verificação e tratamento de erros.

```
options SYSVSHM #SYSV-style shared memory
```

Esta opção fornece compartilhamento de memória no estilo *System V*. O uso mais comum deste, são as extensões XSHM do X, com muitos programas de gráficos-intensivos que obterão automaticamente vantagens e acréscimo de velocidade. Se você usa X, mantenha esta opção ativa.

```
options SYSVSEM #SYSV-style semaphores
```

Suporte à semáforos padrão *System V*. Geralmente são pouco usados, e adicionam algumas centenas de bytes no tamanho total do kernel.

```
options SYSVMSG #SYSV-style message queues
```

Suporte mensagens interprocessos em padrão *System V*. Mais uma vez, também requer poucas centenas de bytes no kernel.

**Nota:** O comando [ipcs\(1\)](#) listará algum processo que estiver fazendo uso de alguma funcionalidade padrão System V.

```
options P1003_1B #Posix P1003_1B real-time extensions
options _KPOSIX_PRIORITY_SCHEDULING
```

Extensões em tempo real adicionadas em 1993 ao padrão POSIX®. Determinadas aplicações da coleção do ports usam estes recursos (tal como **StarOffice™**).

**Nota:** No FreeBSD 5.X, todas estas funcionalidades agora são fornecidos pela opção `_KPOSIX_PRIORITY_SCHEDULING`, e `P1003_1B` não é mais necessária.

```
options ICMP_BANDLIM #Rate limit bad replies
```

Esta opção permite limitar a quantidade de resposta de erros de ICMP. Normalmente esta opção auxilia a proteção do sistema contra ataques do tipo degaço de serviço por pacotes ICMP. É recomendado manter esta opção.

**Nota:** No FreeBSD 5.X, esta característica é habilitada por padrão e a opção `ICMP_BANDLIM` não é mais necessária.

```
To make an SMP kernel, the next two are needed
#options SMP # Symmetric MultiProcessor Kernel
#options APIC_IO # Symmetric (APIC) I/O
```

Ambas opções acima são necessárias para suporte à multiprocessamento simétrico (SMP).

```
device isa
```

Todos os PCs com FreeBSD suportam esta opção. Caso seu computador seja um IBM PS/2 (Arquitetura Micro Canal), o FreeBSD fornece um suporte um pouco limitado. Para maiores informações sobre este, veja a opção MCA, em `/usr/src/sys/i386/conf/LINT`.

```
device eisa
```

Inclua esta opção caso sua placa-mãe seja EISA. Esta opção habilita a auto-deteção e suporte a configuração de todos os dispositivos sobre o barramento EISA.

```
device pci
```

Inclua esta opção se sua placa-mãe suporta PCI. Esta opção habilita a auto-deteção e suporte a configuração de placas PCI e os que estiverem em transição do barramento ISA para PCI.

```
device agp
```

Inclua esta opção se você dispôr de uma placa de AGP em seu sistema. Esta opção habilita suporte à AGP e AGP GART em placas com estes recursos.

```
Floppy drives
device fdc0 at isa? port IO_FD1 irq 6 drq 2
device fd0 at fdc0 drive 0
device fd1 at fdc0 drive 1
```

Trata-se do controlador de disco flexível. *fd0* é o drive flexível A:, e *fd1* é o drive B:.

```
device ata
```

Este driver suporta todos dispositivos ATA e de ATAPI. Você necessitará somente uma linha com a entrada *device ata* para que o kernel detecte todos os dispositivos PCI ATA/ATAPI em uma máquina moderna.

```
device atadisk # ATA disk drives
```

Esta opção é necessário em conjunto com a *device ata* para dispositivos de controle de discos ATA.

```
device atapicd # ATAPI CDROM drives
```

Esta opção é necessário em conjunto com a *device ata* para dispositivos de controle de drives de CD ATAPI.

```
device atapifd # ATAPI floppy drives
```

Esta opção é necessário em conjunto com a *device ata* para controle dispositivos de unidades de disquetes ATAPI.

```
device atapist # ATAPI tape drives
```

Esta opção é necessário em conjunto com a *device ata* para dispositivos de controle de dispositivos de fitas ATAPI.

```
options ATA_STATIC_ID #Static device numbering
```

Esta opção faz o controle estático de número identificador ATA, no padrão antigo, ou os números de dispositivos que são alocados dinamicamente.

```
ATA and ATAPI devices
device ata0 at isa? port IO_WD1 irq 14
device ata1 at isa? port IO_WD2 irq 15
```

É usado em sistemas mais antigos, que não suportam padrão PCI.

```
SCSI Controllers
device ahb # EISA AHA1742 family
device ahc # AHA2940 and onboard AIC7xxx devices
device amd # AMD 53C974 (Teckram DC-390(T))
device dpt # DPT Smartcache - See LINT for options!
device isp # Qlogic family
device ncr # NCR/Symbios Logic
device sym # NCR/Symbios Logic (newer chipsets)

device adv0 at isa?
device adw
device bt0 at isa?
device aha0 at isa?
device aic0 at isa?
```

Controladoras SCSI. Caso alguma destas controladoras não estejam disponíveis em seu sistema, comente a linha em questão. Caso você tenha apenas discos IDE, todas opções aqui podem ser completamente removidas.

```
SCSI peripherals
device scbus # SCSI bus (required)
device da # Direct Access (disks)
device sa # Sequential Access (tape etc)
device cd # CD
device pass # Passthrough device (direct SCSI
access)
```

Periféricos SCSI. Mais uma vez, se algum destes periféricos não estiverem disponíveis em seu sistema, comente a linha em questão. Caso você tenha apenas discos IDE, todas opções aqui podem ser completamente removidas.

```
RAID controllers
device ida # Compaq Smart RAID
device amr # AMI MegaRAID
device mlx # Mylex DAC960 family
```

Suporte à controladoras RAID. Você pode manter apenas suporte aos equipamentos que você tem disponível, removendo o suporte às controladoras RAID não disponíveis.

```
atkbd0 controls both the keyboard and the PS/2 mouse
device atkbd0 at isa? port IO_KBD
```

O controlador de teclado (*atkbd*) fornece recursos de controle à operações de E/S para o teclado AT e teclados e dispositivos de apontamento PS/2 (como *mouse* e canetas). Este dispositivo é necessário ao suporte de teclado (*atkbd*) bem como ao dispositivo de controle PS/2 (*psm*).

```
device atkbd0 at atkbdc? irq 1
```

O dispositivo de controle *atkbd*, em conjunto com o *atkbdc*, provém acesso ao teclado AT 84 ou à teclados AT, modelos aprimorados e convencionais.

```
device psm0 at atkbdc? irq 12
```

Use este dispositivo caso seu mouse seja do tipo que requer conectores PS/2.

```
device vga0 at isa?
```

Dispositivo de controle de vídeo.

```
splash screen/screen saver
pseudo-device splash
```

Possibilita que seja adicionada uma tela de sobreposição (bitmap) na inicialização do sistema. Protetores de tela também exigem suporte a este dispositivo.

```
syscons is the default console driver, resembling an SCO console
device sc0 at isa?
```

*sc0* é o dispositivo padrão de controle do console do sistema, e se assemelha ao console de sistema SCO. A maioria dos terminais requerem este suporte mesmo terminais que fazem uso do mapeamento *termcap*, mas não deve ser relevante quando o dispositivo *vt0* for utilizado, uma vez que este é compatível com o console *VT220*. Neste caso, ao se logar, basta definir a variável de ambiente *TERM* como *scoansi* especialmente alguma aplicação que ocupe a tela inteira apresentar inconformidades no console.

```
Enable this and PCVT_FREEBSD for pcvt vt220 compatible console
driver
#device vt0 at isa?
#options XSERVER # support for X server on a vt console
#options FAT_CURSOR # start with block cursor
If you have a ThinkPAD, uncomment this along with the rest of the
PCVT lines
#options PCVT_SCANSET=2 # IBM keyboards are non-std
```

Trata-se de um dispositivo de controle VT220 compatível com o console, padrões VT100/102. Esta opção é aconselhada para computadores móveis que tenham problemas de compatibilidade com *sc0*. Ajuste também o valor da variável *TERM* para *vt100* ou *vt220* logar-se no terminal. Este dispositivo pode ser usado para interconectar um grande número de diferentes máquinas em um ambiente de rede, onde *termcap* ou *terminfo* contenham entradas para o dispositivo *sc0* -- *vt100* deve estar virtualmente disponível em todas estações.

```
Power management support (see LINT for more options)
device apm0 at nexus? disable flags 0x20 # Advanced Power
Management
```

Suporte à Gerenciamento Avançado de Energia. Usado especialmente em laptops.

```
PCCARD (PCMCIA) support
```



```
device card
device pcic0 at isa? irq 10 port 0x3e0 iomem 0xd0000
device pcic1 at isa? irq 11 port 0x3e2 iomem 0xd4000 disable
```

Suporte PCMCIA. Especialmente útil em laptop.

```
Serial (COM) ports
device sio0 at isa? port IO_COM1 flags 0x10 irq 4
device sio1 at isa? port IO_COM2 irq 3
device sio2 at isa? disable port IO_COM3 irq 5
device sio3 at isa? disable port IO_COM4 irq 9
```

Suporte à portas de comunicação serial. Estes dispositivos são análogamente referentes às portas COM1 até COM4, em ambientes MS-DOS/Windows®.

**Nota:** Se você tiver um modem interno na COM4 e uma porta serial na COM2, você terá que mudar a IRQ do modem para 2 (pelas razões técnicas  $IRQ2 = IRQ9$ ) de forma que o FreeBSD compreenda estas definições. Se você tiver um cartão serial de múltiplas interfaces seriais, verifique a página de manual [sio\(4\)](#) para obter maiores informações sobre os valores apropriados para esta linha de opção do kernel. Algumas placas vídeo (especialmente aquelas com chip baseado em S3) usam endereço de E/S no formato  $0x*2e8$ , e como várias placas seriais simplesmente ignoram qualquer possibilidade de convivência com uma placa de vídeo destas, seu endereçamento de E/S 16-bit costumam conflitar com estes dispositivos, que querem operar na porta COM4 - normalmente indisponível.

Cada porta serial requer uma IRQ exclusiva (a não ser que você esteja usando um cartão multi-portas onde as interrupções compartilhadas são suportadas), sendo assim as IRQs padrão para COM3 e COM4 não são habilitadas automaticamente.

```
Parallel port
device ppc0 at isa? irq 7
```

Trata-se da porta paralela para a interface de barramento ISA.

```
device ppbus # Parallel port bus (required)
```

Suporta o barramento de porta paralela.

```
device lpt # Printer
```

Suporte à porta paralela de impressoras.

**Nota:** Todos os três acima são necessários para habilitar suporte à impressora paralela.

```
device plip # TCP/IP over parallel
```

Dispositivo de controle para a interface de rede via porta paralela.

```
device ppi # Parallel port interface device
```

Uma porta paralela de propósito geral ("geek port") padrão de E/S IEEE1284.

```
#device vpo # Requires scbus and da
```

Trata-se do suporte à ``Zip Drive" da Iomega. Requer ainda suporte à *scbus* e *da*.  
Melhor performance garantida com portas em modo EPP 1.9.

```
PCI Ethernet NICs.
```

```
device de # DEC/Intel DC21x4x (``Tulip'')
device fxp # Intel EtherExpress PRO/100B (82557, 82558)
device tx # SMC 9432TX (83c170 ``EPIC'')
device vx # 3Com 3c590, 3c595 (``Vortex'')
device wx # Intel Gigabit Ethernet Card (``Wiseman'')
```

Suporte à uma série de placas de rede PCI. Você pode comentar ou remover os dispositivos que controlam placas de rede que você não disponha em seu sistema.

```
PCI Ethernet NICs that use the common MII bus controller code.
device miibus # MII bus support
```

O suporte ao barramento MII é necessário para o correto funcionamento de uma série de placas de rede PCI 10/100, especificamente as placas com transceivers MII-compatíveis ou então os que implementam a interface de controle em questão, operando como MII. Adicionar a opção *device miibus* provê suporte à uma API de barramento MII genérica, e a todos os dispositivos de controle PHY, incluindo um genérico para PHYs não declarados no momento da negociação dos recursos.

```
device dc # DEC/Intel 21143 and various workalikes
device rl # RealTek 8129/8139
device sf # Adaptec AIC-6915 (``Starfire'')
device sis # Silicon Integrated Systems SiS 900/SiS 7016
device ste # Sundance ST201 (D-Link DFE-550TX)
device tl # Texas Instruments ThunderLAN
device vr # VIA Rhine, Rhine II
device wb # Winbond W89C840F
device xl # 3Com 3c90x (``Boomerang'', ``Cyclone'')
```

Suporte à placas de rede que usam o código controlador do barramento MII.

```
ISA Ethernet NICs.
```

```
device ed0 at isa? port 0x280 irq 10 iomem 0xd8000
device ex
device ep
WaveLAN/IEEE 802.11 wireless NICs. Note: the WaveLAN/IEEE really
exists only as a PCMCIA device, so there is no ISA attachment needed
and resources will always be dynamically assigned by the pccard
code.
device wi
Aironet 4500/4800 802.11 wireless NICs. Note: the declaration below
will
work for PCMCIA and PCI cards, as well as ISA cards set to ISA PnP
mode (the factory default). If you set the switches on your ISA
card for a manually chosen I/O address and IRQ, you must specify
those parameters here.
device an
The probe order of these is presently determined by
i386/isa/isa_compat.c.
device ie0 at isa? port 0x300 irq 10 iomem 0xd0000
device fe0 at isa? port 0x300
```

```
device le0 at isa? port 0x300 irq 5 iomem 0xd0000
device lnc0 at isa? port 0x280 irq 10 drq 0
device cs0 at isa? port 0x300
device sn0 at isa? port 0x300 irq 10
requires PCCARD (PCMCIA) support to be activated
#device xe0 at isa?
```

Tratam-se de dispositivos de controle de interfaces Ethernet padrão ISA Ethernet drivers. Veja a lista de placas suportadas por cada um destes dispositivos em `/usr/src/sys/i386/conf/LINT`.

```
pseudo-device ether # Ethernet support
```

*ether* é necessário caso você tenha alguma placa de rede disponível. Este dispositivo inclui suporte ao código genérico do protocolo Ethernet.

```
pseudo-device sl 1 # Kernel SLIP
```

*sl* é o suporte à SLIP. Esta tecnologia foi quase completamente substituída pelo PPP, que é mais fácil de configurar, mais flexível e melhor suporte à conexão modem-a-modem, e mais é mais poderoso. O *valor* na frente da entrada *sl* indica quantas sessões simultâneas da conexãoes SLIP serão suportadas.

```
pseudo-device ppp 1 # Kernel PPP
```

Trata-se do suporte PPP no kernel, usado especialmente para conexões discadas. Há também uma outra versão do PPP implementada em nível de usuário com o dispositivo *tun* e oferece mais flexibilidade e características tais como exigem as conexões discadas. O *valor* de conexões após a entrada *ppp*, define a quantidade de conexões simultâneas suportadas pelo PPP.

```
pseudo-device tun # Packet tunnel.
```

É usada pelo programa que implementa o suporte PPP em nível de usuário. O *valor* na frente da entrada *tun* indica o número de sessões simultâneas suportadas pelo PPP. Veja a [PPP](#) deste livro para obter maiores informações.

```
pseudo-device pty # Pseudo-ttys (telnet etc)
```

Trata-se de um "pseudo-terminal" que simula a interface de login no sistema. É usado pelo telnet bem como por seções *rlogin*, **xterm**, e algumas outras aplicações como **Emacs**. O *valor* após a entrada *pty* indica o número de dispositivos *pty* que devem ser criados. Se você tem a necessidade de mais de 16 sessões simultâneas de **xterm** ou acessos remotos, pode aumentar este valor até o máximo de 256.

```
pseudo-device md # Memory ``disks''
```

É o pseudo-dispositivo para discos de memória.

```
pseudo-device gif
```

ou

```
pseudo-device gif 4 # IPv6 and IPv4 tunneling
```

Implementa tunelamento do protocolo IPv6 sobre IPv4, bem como tunelamento IPv4 sobre IPv4, e IPv6 sobre. À partir do FreeBSD 4.4 o dispositivo *gif* pode ser criado por demanda, e a primeira entrada (sem o identificador numérico após o *gif*) pode ser usada. Versões anteriores do FreeBSD requerem esta definição.

```
pseudo-device faith 1 # IPv6-to-IPv4 relaying (translation)
```

Este pseudo-dispositivo captura os pacotes que são emitidos e desvia-os ao daemon da tradução IPv4/IPv6.

```
pseudo-device bpf # Berkeley packet filter
```

Trata-se do Filtro de Pacotes de Berkeley. Este pseudo-dispositivo permite que as interfaces da rede sejam colocadas em modo promíscuo, capturando cada pacote que trafegar publicamente na rede. Estes pacotes podem ser capturados e salvos em disco ou examinados com o programa [tcpdump\(1\)](#).

**Nota:** A opção *pseudo-device bpf* é usada também pelo [dhclient\(8\)](#) para obter o endereço IP do roteador padrão (gateway) e assim por diante. Se você usa DHCP, mantenha esta opção descomentada.

```
Suporte USB
#device uhci # UHCI PCI->USB interface
#device ohci # OHCI PCI->USB interface
#device usb # USB Bus (required)
#device ugen # Generic
#device uhid # ``Human Interface Devices''
#device ukbd # Keyboard
#device ulpt # Printer
#device umass # Disks/Mass storage - Requires scbus and da
#device ums # Mouse
USB Ethernet, requires mii
#device aue # ADMtek USB ethernet
#device cue # CATC USB ethernet
#device kue # Kawasaki LSI USB ethernet
```

Estas opções adicionam suporte à vários dispositivos USB.

Para obter mais informações sobre cada dispositivo USB suportado pelo FreeBSD por gentileza, veja `/usr/src/sys/i386/conf/LINT`.

### 9.4.1. Grandes Configurações de Memória (PAE)

Máquinas com grandes configurações de memória tem a necessidade de que sejam acessados endereços de memória que superam o limite de 4 Gigabytes de espaço no ambiente de usuário e do kernel (KVA). Devido à esta limitação, a Intel adicionou suporte à espaço de endereçamento físico de 36-bit nos processadores no Pentium em sua linha de CPUs Pro e posteriores.

A extensão de endereço físico (PAE) possibilita que as CPUs Intel® Pentium Pro e posteriores possam ter até 64 gigabytes de memória. O FreeBSD oferece suporte à este recurso por meio da opção *PAE* na configuração do kernel, disponível na série 4.X à partir da versão 4.9-RELEASE e no 5.X à partir do 5.1-RELEASE. Devido à limitações da arquitetura de memória Intel, não é possível distinguir o que é endereçado acima ou abaixo dos 4 gibabytes. A memória além dos 4GB é meramente inclusa e mapeada ao *pool* de memória disponível.

Para habilitar suporte ao PAE, basta adicionar ao arquivo de configuração do kernel, a seguinte linha:

```
options PAE
```

**Nota:** O suporte à PAE no FreeBSD limita-se à processadores Intel IA-32. Deve-se também notar, que o suporte à PAE no FreeBSD ainda não foi totalmente testado, e é considerado de qualidade beta, quando comparado com outros recursos estáveis do FreeBSD.

O suporte à PAE no FreeBSD tem algumas limitações:

- Um mesmo processo não pode alocar mais do que 4 gigabytes do espaço da VM.
- Módulos KLD não podem ser carregados dentro em um kernel com PAE habilitado, devido a diferenças na estrutura da configuração do módulo de kernel.
- Dispositivos de controle não usam o a interface [bus\\_dma\(9\)](#) podem causar corrupção de dados em um kernel com PAE, portanto não são recomendados para uso em conjunto com PAE. Por esta razão um arquivo específico para configuração de um kernel com PAE está disponível separadamente no FreeBSD 5.X, onde os dispositivos de controle conflitante estão ausentes.
- Alguns valores ajustáveis no kernel do sistema são baseados na quantidade de memória que foi configurada no kernel, estaticamente, no momento da compilação deste. Alguns destes valores podem ter necessidade de serem realocados devido à natureza do suporte à muita memória disponível em sistemas com PAE. Um bom exemplo é a MIB ajustável via [sysctl\(8\)](#), *kern.maxvnodes*, que controla o número máximo de vnodes que podem ser alocados no kernel. É recomendado que este e outros valores ajustáveis sejam redefinidos à um valor razoável, em sistemas com PAE.
- Pode ser necessário aumentar o espaço virtual de endereçamento do kernel (KVA) ou diminuir a quantidade de recursos utilizados com maior demanda no kernel, objetivando evitar a exaustão do KVA. A opção *KVA\_PAGES* pode ser usada na configuração do kernel para aumentar o espaço disponível para o KVA.

Para mais dicas sobre performance e estabilidade, é recomendado a leitura da página de manual [tuning\(7\)](#). A página de manual [pae\(4\)](#) contém informação recentes sobre o suporte PAE no FreeBSD.

## Notas

- [1] O algoritmo de ajuste automático do *maxuser* define o valor igual a quantidade de memória do sistema, respeitando o valor mínimo de 32, e o máximo de 384.
- 

# 9.5. Criando Dispositivos de Controle

**Nota:** Se você estiver usando o FreeBSD versão 5.0 ou posterior, pode seguramente pular esta seção. Estas versões usam o [devfs\(5\)](#) para criar os dispositivos de controle de forma completamente transparente para o usuário.

Quase todos dispositivo no kernel usam um "dispositivo de controle" que fica no diretório `/dev`. O usuário normal visualiza estes dispositivos como arquivos regulares, mas na realidade são entradas especiais no kernel do sistema, que os programas usam para acessar os recursos do sistema. Existe uma rotina *shell*, o `/dev/MAKEDEV`, que é executado quando você instala o sistema operacional pela primeira vez; esta rotina cria quase todos os dispositivos de controle suportados no sistema. Entretanto, não cria *todos*, de forma que ao adicionar suporte para novos dispositivos, esta rotina deve ser usada para criar a entrada apropriada neste diretório e acima dele. Veja um exemplo simples:

Vamos supor que você adicionou um CD-ROM IDE e requer que ele seja suportado pelo kernel. A linha no kernel deve ser a seguinte:

```
device acd0
```

Ou seja, a entrada `acd0` deve estar disponível no `/dev`, possivelmente seguida por uma letra, tal como `c`, ou precedido por uma letra `r`, que significa tratar-se de um dispositivo crú ("raw"). Existe contudo uma grande possibilidade destes arquivos não estarem presentes sob este diretório, neste caso você pode entrar no diretório `/dev`:

```
sh MAKEDEV acd0
```

Quando a rotina terminar de ser executada, deverão existir as entradas `acd0c` e `racd0c` sob `/dev`.

Para placas de som, o seguinte comando cria as entradas apropriadas:

```
sh MAKEDEV snd0
```

**Nota:** Ao criar dispositivos de controle tais como Placas de Som, se outras pessoas tiverem acesso à sua máquina, pode ser necessário proteger os dispositivos contra acesso externo adicionando os arquivos no `/etc/fstab`. Veja o manual do [fstab\(5\)](#) para obter maiores informações.

Siga este simples procedimento quando quiser adicionar dispositivos de controle à outros equipamentos não GENÉRICOS que não tiverem entradas apropriadas.

**Nota:** Todas as controladoras SCSI usam a mesma entrada sob o `/dev`, de forma que estes não requerem que os dispositivos sejam manualmente criados. As placas de rede e SLIP/PPP não tem dispositivos de controle sob o `/dev`, de forma que você não deve se preocupar também com estes equipamentos.

---

## 9.6. Se algo der errado

Há 5 categorias de problemas que podem ocorrer ao compilar um kernel personalizado. Elas são:

Falhas na configuração:

Se a rotina [config\(8\)](#) falhar, é provável que você tenha cometido um erro bastante simples. Felizmente, o [config\(8\)](#) indicará o número da linha que está com problema, de forma que você pode tranquilamente ir diretamente à esta linha, com um editor de texto como o `vi`. Por exemplo, se você encontrar o seguinte erro:

```
config: line 17: syntax error
```

Você vai identificar o problema, usando o `vi`, com o comando `17G` no modo de comandos do editor. Certifique-se que o a opção em questão está corretamente configurada, comparando o kernel GENÉRICO com sua outra referência de arquivo de kernel.

Falha no `make`:

Se o comando `make` falhar, é indicação de que existem erros no seu arquivo de configuração do kernel, mas estes erros não são suficientemente claros para o [config\(8\)](#) indentifica-lo. Mas uma vez, dê uma olhada em seu arquivo de configuração do kernel, e se não conseguir solucionar a questão, envie uma mensagem de correio eletrônico para [lista de discussão FreeBSD de perguntas genéricas](#) com a configuração do seu kernel, de forma que o problema seja diagnosticado de forma rápida.

Falha na instalação do kernel:

Se o kernel estiver perfeitamente compilado, e não foi instalado corretamente (no caso do comando `make install` ou `make installkernel` ter

falhado), a primeira providência a se tomar é verificar se seu sistema não está sendo executado com nível de segurança positivo (*securelevel* 1 ou superior - veja [init\(8\)](#)). O processo de instalação do kernel tenta modificar a região de memória que armazena informações sobre o kernel, e coloca esta memória em uma outra região. Se o nível de segurança for 1 ou superior, o acesso direto à memória não é permitido. A instalação do kernel precisa ser feita no nível de segurança 0 ou inferior.

O kernel não carrega:

Se o kernel novo não carregar, ou falhar no momento de reconhecer os dispositivos instalados em seu sistema, não se apavore! Felizmente o FreeBSD tem um excelente mecanismo de recuperação de kernel incompatível. Escolha o kernel que você deseja carregar e inicie o FreeBSD com ele. Para fazer isso, quando o sistema de inicialização estiver na contagem regressiva de 10 segundos, aperte qualquer tecla, exceto a tecla **Enter**, use o comando `unload` e então mande iniciar o kernel antigo, com o comando `boot kernel.old`, ou o nome de qualquer outro kernel que você achar conveniente e que exista em seu FreeBSD. É sempre boa idéia manter um kernel confiável (ou uma cópia dele) quando começar o processo de recompilação e instalação de um novo.

Depois de carregar um kernel funcional, verifique o arquivo de configuração do seu kernel customizado (o que falhou), corrija as possíveis deficiências e recompile-o. Um recurso muito útil que pode ajudar, é o fato do arquivo `/var/log/messages` gravar, entre outras coisas, todas as mensagens dos kernel até o momento em que este foi carregado com sucesso. O comando [dmesg\(8\)](#) imprimirá sempre as últimas mensagens atualmente geradas pelo kernel.

**Nota:** Se você estiver tendo problemas para compilar um kernel, tome uma precaução, e mantenha um `GENERIC`, ou qualquer outro kernel funcional, disponível. Você não pode confiar sempre no `kernel.old` pois no caso de um segundo kernel problemático ser instalado, o problemático anterior será o `kernel.old`, sobrescrevendo o arquivo funcional. Lembre-se de corrigir o kernel e ter um novo kernel funcional o quanto antes, ou de mover o kernel de segurança para o nome original de `kernel`, pois comandos como o [ps\(1\)](#) podem não funcionar corretamente. O comando `destrava` as opções de imutabilidade do kernel, e possibilita que ele seja renomeado ou sobreposto.

```
chflags noschg /kernel
```

Se você encontrar qualquer problema relacionado à esta etapa do procedimento de instalação, é provável que esteja usando o FreeBSD em nível de segurança ([securelevel\(8\)](#)) maior que zero. Neste caso edite a linha `kern_securelevel` no arquivo `/etc/rc.conf` e ajuste seu valor para `-1`, depois reinicie o FreeBSD. Esta configuração pode ser feita antes do início da configuração do novo kernel.



Se você quiser assegurar seu kernel e ``trava-lo" em algum local, de forma que seu kernel funcional não possa ser modificado ou alterado, faça o seguinte:

```
chflags schg /kernel
```

No FreeBSD 5.X, o kernel não é instalado em uma região da memória que não pode ser modificado, de forma que este problema especificamente, seja improvável como causa das dificuldades que você está tendo.

O kernel funciona, mas o [ps\(1\)](#) não funciona mais:

Se você instalou uma versão do kernel diferente da versão dos utilitários do sistema, por exemplo, um kernel 4.X em um sistema 3.X, vários comandos de estatísticas do sistema como [ps\(1\)](#) e [vmstat\(8\)](#) não funcionarão mais. A biblioteca `libkvm` deve ser recompilada, bem como estes utilitários todos. É por essa razão que normalmente não é uma boa idéia usar um kernel de versão diferente de todo o resto do sistema operacional.

---

## Capítulo 10. Segurança

### Índice

- 10.1. [Sinopse](#)
- 10.2. [Introdução](#)
- 10.3. [Protegendo FreeBSD](#)
- 10.4. [DES, MD5, e Crypt](#)
- 10.5. [Senhas de uso único](#)
- 10.6. [KerberosIV](#)
- 10.7. [Kerberos5](#)
- 10.8. [Firewalls](#)
- 10.9. [OpenSSL](#)
- 10.10. [VPN sobre IPsec](#)
- 10.11. [OpenSSH](#)
- 10.12. [Controle de Acesso Obrigatório \(MAC-Mandatory Access Control\)](#)
- 10.13. [File System Access Control Lists](#)
- 10.14. [Recomendações de Segurança do FreeBSD](#)

*Muito deste capítulo foi obtido da página de manual [security\(7\)](#) Matthew Dillon.*

### 10.1. Sinopse

Este capítulo irá fornecer uma instrução básica sobre os conceitos de segurança do sistema, Algumas regras comuns de manuseio, e alguns tópicos avançados sobre FreeBSD. Uma boa quantidade dos tópicos abordados aqui pode ser aplicada a segurança do sistema e da Internet de um modo geral. A Internet não é mais o local ``amigável" no qual todos querem ser seus bons vizinhos. Garantir a segurança do seu

sistema tornou-se imperativo para proteger seus dados, propriedade intelectual, tempo, e muito mais das mãos de hackers e afins.

FreeBSD dispõe de uma estrutura de utilitários e mecanismos para assegurar a integridade e a segurança do seu sistema e de sua rede.

Lendo este capítulo, você irá aprender sobre:

- Conceitos básicos de segurança do sistema, relacionados ao FreeBSD.
- Diversos mecanismos de criptografia, disponíveis no FreeBSD, como o DES e o MD5.
- Como configurar autenticação com senha de uso único.
- Como configurar **KerberosIV** no FreeBSD em versões anteriores a 5.0.
- Como configurar **Kerberos5** em versões posteriores a 5.0.
- Como criar firewalls usando IPFW.
- Como configurar IPsec e criar uma VPN entre máquinas FreeBSD/Windows®.
- Como configurar e usar **OpenSSH**, a implementação do SSH no FreeBSD.
- Como configurar e carregar módulos de extensão de controle de acesso usando o Sistema MAC do TrustedBSD.
- O que é uma Lista de Controle de Acesso do sistema de arquivos e como usa-la.
- Como utilizar as publicações do Conselho de Segurança do FreeBSD.

Antes de ler este capítulo você precisa:

- Entender os conceitos básicos de FreeBSD e Internet.

---

## 10.2. Introdução

Segurança é uma função que começa e termina com o administrador do sistema. Enquanto todo sistema BSD UNIX® multi-usuário tem alguma segurança nativa, o trabalho de criar e manter mecanismos adicionais de segurança para controlar aqueles usuários "honestos" é provavelmente uma das grandes e solitárias tarefas do "sysadmin-Administrador do sistema". Máquinas estão seguras somente quando você as faz assim, e o interesse por segurança está sempre competindo com a necessidade humana por conveniência. Sistemas UNIX, em geral, são capazes de executar um grande número de processos simultâneos e muitos destes processos operam como servidores - Significando que entidades externas podem conectar e conversar com eles. A medida que os mini-computadores e *mainframes* de ontem tornaram-se os micro-computadores de hoje, e a medida que os computadores foram se conectando em redes e as redes se conectaram entre elas, segurança passou a ser um assunto sempre mais importante.

A melhor maneira de implementar segurança é através de camadas, como se fosse uma "cebola". Em resumo, o que você precisa fazer é tantas camadas de proteção quanto lhe for conveniente e então monitorar cuidadosamente o sistema contra criar invasões. Você não precisa exagerar em suas configurações de proteção, ou irá interferir com a possibilidade de detecção, e a detecção é isoladamente um dos aspectos mais

importantes de qualquer mecanismo de segurança. Por exemplo, faria pouco sentido definir o flag *schg* (veja [chflags\(1\)](#)) em cada binário do sistema, porque enquanto isto pode temporariamente proteger os binários, irá impedir que um intruso que obteve acesso ao seu sistema faça uma alteração que seria facilmente detectada e levar seus mecanismos de segurança a nunca detectarem o intruso ou a falha que foi explorada.

Segurança de sistema também está relacionada com várias formas de ataque, incluindo ataques que tentam travar, ou de alguma forma impedir o uso ou funcionamento do sistema, mas que não tentam comprometer a conta `root` ("break root"). Preocupações com segurança podem ser divididas em diversas categorias:

1. Ataque de negação de serviços (*DoS-Denial of service attacks*).
2. Comprometimento de contas de usuário.
3. Comprometimento do `root` através de servidores acessíveis.
4. Comprometimento do `root` via contas de usuários.
5. Criação de portas de acesso clandestinas (*Backdoor*).

Um ataque de negação de serviço é uma ação que torna indisponíveis à máquina os recursos necessários. Tipicamente, um ataque DoS é um mecanismo de força bruta que tenta travar ou então tornar a máquina temporariamente inutilizável entupindo um serviço ou a capacidade da rede. Alguns ataques DoS tentam tirar vantagem de falhas na camada de rede para travar a máquina com um único pacote. Este último somente pode ser corrigido aplicando a correção da falha ao kernel. Ataques a servidores frequentemente podem ser corrigidos por uma configuração adequada dos limites da carga que os servidores impõe ao sistema sob condições adversas. Ataques de força Bruta por rede são muito difíceis de tratar. Um ataque por pacote falsificado-*spoofed-packet*, por exemplo, é quase impossível de parar, colocando seu sistema repentinamente fora da internet. Ele pode não ser capaz de colocar seu equipamento fora de operação, mas ele pode saturar sua conexão com a internet.

Um ataque às contas de usuário é até mais comum que um ataque DoS. Muitos administradores de sistema ainda executam servidores padrão de **telnetd**, **rlogind**, **rshd**, and **ftpd** em suas máquinas. Estes Servidores, por padrão, não operam dentro de conexões encriptadas. o resultado é que se você tem qualquer base de usuários de tamanho moderado, um ou mais dos seus usuários que conectam em seu sistema de um local remoto (o qual é o mais comum e conveniente meio de conectar a um sistema) terão a sua senha descoberta. O administrador de sistema atencioso irá analisar seus logs de acesso remoto buscando por endereços de origem suspeita mesmo para autenticações bem sucedidas.

Deve-se sempre assumir que uma vez o intruso tem acesso a conta de usuário, ele pode tomar a conta de `root`. De qualquer forma, a realidade é que em um sistema bem seguro e com boa manutenção, acessar uma conta e usuário não irá necessariamente dar ao intruso acesso ao `root`. Esta distinção é importante porque sem o acesso a conta de `root` o intruso geralmente não poderá esconder suas pistas e talvez, melhor ainda, esteja apto somente a nada mais que bagunçar os arquivos do usuário, ou travar a máquina. o Comprometimento da conta de usuário é muito comum porque usuários tendem a não tomar as precauções que um administrador de sistema toma.

Administradores de sistema precisam ter em mente que ali existem potencialmente muitas maneiras de violar uma conta `root` em uma máquina. O intruso pode conhecer a senha de `root`, o intruso pode encontrar uma falha em um servidor rodando com o usuário `root` e obter acesso através de uma conexão de rede para este servidor, ou o intruso pode conhecer uma falha em um programa com privilégios de `root` que permite ao intruso violar a conta `root` uma vez que ele tenha invadido o sistema através de uma conta de usuário. Se um intruso encontrou uma maneira de violar a conta de `root` em uma máquina, ele pode não ter a necessidade de instalar uma porta de acesso clandestina-*backdoor*. Muitas das falhas em contas `root` encontradas e corrigidas até hoje envolveram um considerável esforço do intruso para limpar suas pistas, então a maioria dos intrusos instala uma porta de acesso clandestina-*backdoor*. Uma porta de acesso clandestina-*backdoor* fornece ao intruso uma maneira fácil e conveniente retomar o acesso pela conta de `root` do sistema, mas isto permite ao administrador de sistema uma maneira conveniente para detectar a intrusão. Making it impossible for an attacker to install a backdoor may actually be detrimental to your security, because it will not close off the hole the attacker found to break in the first place.

Security remedies should always be implemented with a multi-layered ``onion peel'' approach and can be categorized as follows:

1. Securing `root` and staff accounts.
2. Securing `root` - root-run servers and `suid/sgid` binaries.
3. Securing user accounts.
4. Securing the password file.
5. Securing the kernel core, raw devices, and filesystems.
6. Quick detection of inappropriate changes made to the system.
7. Paranoia.

The next section of this chapter will cover the above bullet items in greater depth.

---

## 10.3. Protegendo FreeBSD

**Comando vs. Protocolo:** Durante todo este documento, nós vamos usar o texto em **negrito** para referenciar a um comando ou aplicação. Será usado para exemplos como `ssh`, desde que seja um protocolo bem como um comando.

As seções que seguem vão cobrir os métodos para proteger seu sistema FreeBSD que foram mencionadas na [última seção](#) deste capítulo.

### 10.3.1. Protegendo a conta `root` e contas do sistema

Primeiramente, não perca tempo protegendo as contas do sistema se você não protegeu a conta `root`. A maioria dos sistemas tem uma senha atribuída a conta `root`. A primeira coisa que você faz é assumir que a senha está *sempre* comprometida. Isso não significa que você deve remover a senha. A senha é quase sempre necessária para o acesso ao console da máquina. O que isso significa é que você não deve possibilitar o uso da

senha fora do console ou possivelmente com o comando [su\(1\)](#). Por exemplo, certifique-se de que seus terminais *pty* estão especificados como sendo *insecure* no arquivo `/etc/ttys` de modo que logins diretos como `root` via `telnet` ou `rlogin` serão rejeitados. Se estiver usando outros serviços de login tais como `sshd`, certifique-se de que logins diretos como `root` estão desabilitados também. Você pode fazer isso editando seu arquivo `/etc/ssh/ssh_config`, e certificando-se de que `PermitRootLogin` está ajustada para *NO*. Considere cada método de acesso - serviços tais como FTP often fall through the cracks. Logins diretos como `root` somente devem ser permitidos através do console do sistema.

Certamente, como um administrador do sistema você tem que ser capaz de pegar `root`, assim nós abrimos alguns furos. Mas nos certificamos que estes furos exijam uma senha adicional para operar. Uma maneira para fazer o `root` acessível é adicionar contas apropriadas ao grupo `wheel` (em `/etc/group`). Os membros colocados no grupo `wheel` são permitidos usar `su` para `root`. Você nunca deve dar aos membros do sistema acesso nativo ao `wheel` colocando-os no grupo `wheel` em sua entrada no arquivo de senhas. Contas do sistema devem ser colocadas em um grupo `staff`, e então ser adicionados ao grupo `wheel` através do arquivo `/etc/group`. Somente aqueles membros que realmente necessitam ter acesso `root` devem ser colocados no grupo `wheel`. É possível também, quando usar um método de autenticação tal como Kerberos, usar o arquivo `.k5login` do Kerberos na conta `root` para permitir um [ksu\(1\)](#) para `root` sem ter que colocar qualquer um no grupo `wheel`. Esta pode ser a melhor solução visto que o mecanismo `wheel` ainda permite que um intruso viole o `root` se conseguir seu arquivo de senhas e invadir uma conta do sistema. Enquanto ter o mecanismo `wheel` é melhor do que nada, esta não é necessariamente a opção mais segura.

Uma maneira indireta de proteger contas do sistema, e finalmente acesso `root` é usar um método de login de acesso alternativo e fazer o que é conhecido como "colocar estrela" em senhas criptografadas para as contas do sistema. Usando o comando [vipw\(8\)](#), pode-se substituir cada instância de uma senha criptografada com um único carácter ```*"`. Este comando atualizará o arquivo `/etc/master.passwd` e a base de dados de usuários/senhas para desabilitar logins autenticados por senha.

Uma entrada de contas do sistema como:

```
foobar:R9DT/Fa1/LV9U:1000:1000::0:0:Foo
Bar:/home/foobar:/usr/local/bin/tcsh
```

Deveria ser trocada para:

```
foobar:*:1000:1000::0:0:Foo Bar:/home/foobar:/usr/local/bin/tcsh
```

Esta mudança impedirá que logins normais ocorram, visto que a senha criptografada nunca combina com ```*"`. Com isto feito, os membros do sistema devem usar um outro mecanismo para se autenticar tais como [kerberos\(1\)](#) ou [ssh\(1\)](#) usando um par de chaves pública/privada. Quando usar algo como o Kerberos, geralmente deve-se proteger as máquinas que executam os servidores Kerberos e sua estação de trabalho *desktop*. Quando usar um par de chaves pública/privada com `ssh`, geralmente deve proteger a máquina usada para fazer login *de* (tipicamente uma estação de trabalho). Uma camada adicional de proteção pode ser adicionada ao par de chaves pela senha que protege o par

de chaves ao criá-las com [ssh-keygen\(1\)](#). Sendo capaz de ``colocar estrela" nas senhas para contas também garante que os membros do sistema podem apenas fazer login através de métodos de acesso seguros que você criou. Isto força todos os membros do sistema a usar conexões seguras, criptografadas para todas as seções, que fecham um furo importante usado por muitos invasores: *sniffing* na rede de uma máquina menos segura.

Os mecanismos mais indiretos de proteção também assumem que você está logando de um servidor mais restritivo para um servidor menos restritivo. Por exemplo, se sua máquina principal está executando todos os tipos de serviços, sua estação de trabalho não deve executar algum. Para que sua estação de trabalho seja razoavelmente segura você deve executar o mínimo de serviços possíveis, incluindo até nenhum serviço, e você deve executar uma proteção de tela protegida por senha. Naturalmente, dando acesso físico a uma estação de trabalho para um atacante pode quebrar qualquer tipo de segurança que você colocou. Isto é definitivamente um problema que você deve considerar, mas você também deve considerar o fato de que a grande maioria das violações ocorrem remotamente, através da rede, de pessoas que não têm acesso físico a sua estação de trabalho ou servidores.

Usando algo como o Kerberos também te dá a habilidade de desativar ou trocar a senha para as contas do sistema em um lugar, e ter efeito imediatamente em todas as máquinas em que os membros do sistema podem ter uma conta. Se uma conta de um membro do sistema for comprometida, a habilidade de trocar imediatamente sua senha em todas as máquinas não deve ser desconsiderada. Com senhas separadas, trocar a senha em N máquinas pode ser um problema. Você também pode impor restrições de troca de senha com Kerberos: não somente um ticket Kerberos pode ser feito para timeout após um tempo, mas o sistema Kerberos pode exigir que o usuário escolha uma nova senha após um certo período de tempo (digamos, uma vez ao mês).

### 10.3.2. Protegendo serviços que executam como root e Binários SUID/SGID

O administrador de sistemas prudente somente executa serviços que necessita, nem mais, nem menos. Esteja prevenido que serviços de terceiros são frequentemente os mais sujeitos a falhas. Por exemplo, executando uma versão antiga do **imapd** ou **popper** é como dar um ticket `root` universal para o mundo inteiro. Nunca execute um serviço que você não verificou com cuidado. Muitos serviços não precisam ser executados como `root`. Por exemplo, os daemons **ntalk**, **comsat**, e **finger** podem ser executados em *sandboxes* especiais de usuário. Um sandbox não é perfeito, a menos que você atravesse por uma quantidade grande de problemas, mas a aproximação da cebola à segurança ainda permanece: Se alguém puder entrar através de um servidor executando em um sandbox, ele ainda tem que quebrar o sandbox. Quanto mais camadas o atacante deve quebrar, mais baixa a probabilidade de seu sucesso. Furos envolvendo `root` foram encontrados historicamente em todos os serviços que sempre executam como `root`, incluindo serviços básicos do sistema. Se você está usando uma máquina através da qual pessoas somente fazem login via **sshd** e nunca via **telnetd** ou **rshd** ou **rlogind**, então desligue estes serviços!

O FreeBSD agora por padrão executa **ntalkd**, **comsat**, e **finger** em um sandbox. Um outro programa que pode ser candidato para executar em um sandbox é o [named\(8\)](#). O

`/etc/defaults/rc.conf` inclui os argumentos necessários para executar o **named** em um sandbox em um formulário comentado. Dependendo se você está instalando um sistema novo ou atualizando um sistema existente, contas especiais de usuário usadas por estes sandboxes podem não estar instaladas. O administrador de sistemas prudente pesquisa e implementa sandboxes para serviços sempre que possível.

Há um número de outros serviços que tipicamente não executam em sandboxes: **sendmail**, **popper**, **imapd**, **ftpd**, e outros. Há alternativas para alguns destes, mas instalá-las pode exigir mais trabalho do que você está disposto a executar (o fator conveniência ataca outra vez). Você pode ter que executar estes serviços como `root` e confiar em outros mecanismos para detectar violações que podem ocorrer através deles.

Os outros furos com grande potencial `root` em um sistema são os binários `suid-root` e `sgid` instalados no sistema. A maioria destes binários, tais como **rlogin**, residem em `/bin`, `/sbin`, `/usr/bin`, ou `/usr/sbin`. Enquanto nada é 100% seguro, os binários `suid` e `sgid` padrões do sistema podem ser considerados razoavelmente seguros. Apesar disso, furos `root` são encontrados ocasionalmente nestes binários. Um furo `root` foi encontrado no *Xlib* em 1998 que fez o **xterm** (que é tipicamente `suid`) vulnerável. É melhor estar seguro do que lamentar-se e o administrador de sistemas prudente restringirá binários `suid`, somente aqueles que usuários do sistema devem executar, a um grupo especial que somente estes usuários podem acessar, e começa a se livrar (`chmod 000`) de todos os binários `suid` que ninguém usa. Um servidor sem monitor geralmente não necessita um binário **xterm**. Binários `sgid` podem ser quase sempre perigosos. Se um intruso puder violar um binário `sgid-kmem`, o intruso provavelmente será capaz de ler o `/dev/kmem` e assim ler o arquivo de senhas criptografadas, comprometendo potencialmente qualquer conta com senha. Alternativamente um intruso que viole o grupo `kmem` pode monitorar as teclas pressionadas enviadas através dos terminais `pty`, incluindo `pty's` usados por usuários que fazem login através de métodos seguros. Um intruso que viole o grupo `tty` pode escrever para quase todos os `tty` de usuário. Se um usuário estiver executando um programa ou emulador no terminal com característica de simulação de teclado, o intruso pode gerar potencialmente um data stream que ocasiona o terminal do usuário a executar um comando, que é então executado como esse usuário.

### 10.3.3. Protegendo Contas de Usuários

Contas de usuário geralmente são as mais difíceis de proteger. Enquanto você pode impor restrições de acesso Draconianas em contas do sistema e colocar ``estrela" nas suas senhas, você não pode ser capaz de fazer algo assim com todas as contas gerais de usuário que você pode ter. Se você tiver controle suficiente, então você pode conseguir e ser capaz de proteger as contas de usuário corretamente. Se não, você simplesmente tem que ser mais vigilante na monitoração daquelas contas. O uso do `ssh` e `Kerberos` para contas de usuário é mais problemática, devido a administração extra e suporte técnico requeridos, mas ainda uma solução muito boa comparada a um arquivo de senhas criptografadas.

### 10.3.4. Protegendo o Arquivo de Senhas

A única maneira certa é colocar \* em quantas senhas você puder e usar ssh ou Kerberos para o acesso daquelas contas. Mesmo que o arquivo de senhas criptografadas (`/etc/spwd.db`) possa somente ser lido pelo `root`, pode ser possível para um intruso obter acesso de leitura a esse arquivo mesmo se o atacante não pode obter acesso de escrita `root`.

Seus scripts de segurança devem sempre verificar e relatar mudanças ao arquivo de senhas (veja a seção [Verificando integridade de arquivos](#) abaixo).

### 10.3.5. Protegendo o Kernel Core, Raw Devices, e sistemas de arquivos

Se um atacante violar o `root` ele pode fazer quase que qualquer coisa, mas há certas conveniências. Por exemplo, a maioria dos *kernel*s modernos tem um controlador de dispositivo de farejar (*sniffing*) pacotes embutido. Sob FreeBSD é chamado dispositivo `bpf`. Um intruso tentará geralmente executar um farejador (*sniffer*) de pacotes em uma máquina comprometida. Você não necessita dar ao intruso esta capacidade e a maioria dos sistemas não tem a necessidade para o dispositivo `bpf` compilado.

Mas mesmo se você desligar o dispositivo `bpf`, você ainda tem `/dev/mem` e `/dev/kmem` para preocupar-se. De qualquer forma, o intruso pode ainda escrever em raw disk devices. Também, há outra característica no *kernel* chamada carregador de módulo, [kldload\(8\)](#). Um intruso ousado pode usar um módulo KLD para instalar seu próprio dispositivo `bpf`, ou outro dispositivo de farejar (*sniffing*), em um *kernel* funcionando. Para evitar estes problemas você tem que executar o kernel em um nível de segurança elevado, pelo menos *securelevel* 1. O *securelevel* pode ser ajustado com `sysctl` na variável `kern.securelevel`. Uma vez que você ajustou o *securelevel* para 1, acesso de escrita a raw devices estarão negados e flags especiais `chflags`, tais como *schg*, serão reforçadas. Você deve também assegurar que a flag *schg* esteja ajustada em binários críticos de inicialização, diretórios, e arquivos *script* - tudo que começa a executar até o ponto onde o *securelevel* é ajustado. Isto pode ser um exagero, e atualizar o sistema é muito mais difícil quando você opera em um nível de segurança elevado. Você pode acertar e executar o sistema em um nível de segurança elevado mas não ajustar a flag *schg* para cada arquivo e diretório do sistema. Outra possibilidade é simplesmente montar o `/e/usr` apenas leitura. Deve-se notar que ser Draconiano demais no que você tenta proteger pode impedir toda importante detecção de uma invasão.

### 10.3.6. Verificando Integridade de Arquivo: Binários, Arquivos de Configuração, Etc.

Quando vier abaixo, você pode somente proteger sua configuração core do sistema e arquivos de controle muito antes do fator conveniência elevar-se a sua cabeça. Por exemplo, usando `chflags` para ajustar o bit *schg* na maioria dos arquivos em `/e/usr` é provavelmente improdutivo, porque enquanto pode proteger os arquivos, fecha também uma janela de detecção. A última camada de sua cebola de segurança é talvez a mais importante - detecção. O resto da sua segurança é inútil (ou, pior, oferece a você um falso senso de segurança) se você não puder detectar invasões potenciais. A metade do



trabalho da cebola é atrasar o atacante, preferencialmente do que para-lo, a fim de dar ao lado da detecção da equação uma oportunidade de pegá-lo no ato.

A melhor maneira de detectar uma invasão é procurar por arquivos modificados, perdidos ou inesperados. A melhor maneira de procurar por arquivos modificados é de outro sistema (frequentemente centralizado) com acesso limitado. Escrevendo seus scripts de segurança no sistema com acesso limitado extra-seguro fá-los na maioria das vezes invisíveis para atacantes potenciais, e isto é importante. A fim de conseguir vantagem máxima você geralmente tem que dar a máquina com acesso limitado, acesso significativo as outras máquinas no local, geralmente fazendo a exportação NFS apenas leitura das outras máquinas para a máquina com acesso limitado, ou criando o par de chaves ssh para permitir a máquina com acesso limitado fazer ssh para outras máquinas. A exceção do seu tráfego de rede, o NFS é o método menos visível - permitindo que você monitore o sistema de arquivos em cada máquina cliente não detectada virtualmente. Se seu servidor com acesso limitado esta conectado as máquinas clientes através de um *switch*, o método NFS é frequentemente a melhor escolha. Se seu servidor com acesso limitado esta conectado as máquinas clientes através de um *hub*, ou através de diversas camadas de roteamento, o método NFS pode ser inseguro demais (network-wise) e usar ssh pode ser a melhor escolha mesmo com audit-trail tracks que o ssh coloca.

Uma vez que você da a máquina com acesso limitado, ao menos acesso leitura aos sistemas clientes supostos a monitorar, você deve escrever scripts para fazer a monitoração real. Dado uma montagem NFS, você pode escrever scripts de simples utilitários do sistema tais como [find\(1\)](#) e [md5\(1\)](#). É melhor fazer fisicamente o md5 dos arquivos da máquina cliente ao menos uma vez por dia, e para testar arquivos de controle tais como aqueles encontradas em */etc* e */usr/local/etc* até mais frequentemente. Quando diferenças são encontradas, relativo a base de informação md5 a máquina com acesso limitado sabe, e deve mostrar para um administrador do sistema para verificar isso. Um bom script de segurança verificará também por binários *suid* inapropriados e por arquivos novos ou deletados em partições do sistema tais como */* e */usr*.

Quanto usar ssh em vez do NFS, escrever o script de segurança é muito mais difícil. Você essencialmente tem que copiar com *scp* os scripts para a máquina cliente a fim de executar eles, fazendo-os visíveis, e por segurança você necessita também copiar com *scp* os binários (tais como *find*) que aqueles scripts usam. O cliente **ssh** na máquina cliente pode já estar comprometido. De modo geral, usar o ssh pode ser necessario quando estiver usando links inseguros, mas ele é também muito difícil de se lidar.

Um bom script de segurança verificará também por mudanças aos arquivos de configuração de acesso dos usuários e membros do sistema: *.rhosts*, *.shosts*, *.ssh/authorized\_keys* e assim por diante... arquivos que puderam ficar fora do alcance da verificação *MD5*.

Se você tiver uma grande quantidade de espaço em disco, isto pode demorar muito para executar em cada arquivo naquela partição. Neste caso, ajustando as flags da montagem para desabilitar binários *suid* e dispositivos naquelas partições são uma boa idéia. As opções *nODEV* e *nosuid* (veja [mount\(8\)](#)) são as que você precisa procurar. Você deve

provavelmente fazer a varredura de qualquer maneira, ao menos uma vez por semana, desde que o objeto desta camada detecte uma invasão se a invasão for eficaz ou não.

Contabilidade de processos (veja [accton\(8\)](#)) é uma característica relativamente de baixo overhead do sistema operacional que pode ajudar como um mecanismo de avaliação post-break-in. É especialmente útil em rastrear como um intruso violou realmente um sistema, assumindo que o arquivo ainda esteja intacto depois que a invasão ocorrer.

Finalmente, scripts de segurança devem processar os arquivos de log, e os logs deles mesmos devem ser gerados em maneira tão segura como for possível - syslog remoto pode ser muito útil. Um intruso tenta cobrir seus rastros, e o arquivos de log são críticos ao administrador do sistema que tentam rastrear a hora e método da invasão inicial. Uma maneira de manter um registro permanente do arquivo de log é executar o console do sistema em uma porta serial e coletar as informações em uma base contínua através de uma máquina segura monitorando os consoles.

### 10.3.7. Paranoia

Um pouco de paranoia nunca machuca. Em geral, um administrador do sistema pode adicionar qualquer número de características de segurança, as long as they do not effect convenience, and can add security features that *do* effect convenience with some added thought. Até mais importante, um administrador de segurança deve misturar uma parte - se você usar recomendações tais como aquelas dadas por este documento literalmente, você da suas metodologias para a perspectiva do atacante que também tem acesso a este documento.

### 10.3.8. Ataques de Negação de Serviço

Esta seção cobre Ataques de Negação de Serviço. Um ataque DoS é tipicamente um ataque de pacote. Enquanto não há muito o que você pode fazer sobre ataques modernos de pacotes falsificados *spoofed* que saturam sua rede, você pode geralmente limitar os danos assegurando que os ataques não possam derrubar seus servidores.

1. Limitando server forks.
2. Limitando ataques springboard (ataques de resposta ICMP, ping broadcast, etc.).
3. Kernel Route Cache.

Um ataque DoS comum é contra a forking server que tenta fazer com que o servidor coma processos, descritores de arquivos e memória, até que a máquina morra. O **inetd** (veja [inetd\(8\)](#)) tem diversas opções para limitar este tipo de ataque. Deve-se notar que enquanto é possível impedir que uma máquina caia, não é geralmente possível impedir que um serviço seja interrompido pelo ataque. Leia a página de manual **inetd** cuidadosamente e preste atenção específica para as opções *-c*, *-C*, e *-R*. Note que ataques de IP falsificados *spoofed* vão enganar a opção *-C* para o **inetd**, assim tipicamente uma combinação de opções deve ser usada. Alguns serviços standalone tem parâmetros self-fork-limitation.

O **sendmail** tem sua opção *-OMaxDaemonChildren*, que tende a funcionar muito melhor do que tentando usar opções de limite de carga do sendmail devido ao atraso de carga. Você deve especificar um parâmetro *MaxDaemonChildren*, quando você inicia o

**sendmail**, alto o bastante para segurar sua carga esperada, mas não tão alto que o computador não pode segurar esse número de **sendmails** sem cair. É também prudente executar o sendmail no modo de fila (*-ODeliveryMode=queued*) e executar o daemon (`sendmail -bd`) separado da execução de filas (`sendmail -q15m`). Se você ainda quiser entrega em tempo real você pode executar a fila em um intervalo mais baixo, como *-q1m*, mas esteja certo de especificar uma opção *MaxDaemonChildren* razoável para *esse* sendmail para impedir falhas em cascata.

O **syslogd** pode ser atacado diretamente e recomenda-se fortemente que você use a opção *-s* sempre que possível, e a opção *-a* de qualquer forma.

Você deve também ser bastante cauteloso com serviços connect-back tais como o indent reverso do **tcpwrapper**, que pode ser atacado diretamente. Você geralmente não precisa usar a característica indent reverso do **tcpwrappers** por esta razão.

É uma idéia muito boa proteger serviços internos do acesso externo por firewalling mantendo-os fora em seus roteadores de borda. A idéia aqui é impedir ataques de saturação de fora da sua LAN, não para proteger serviços internos de um comprometimento `root` baseado em rede. Configure sempre um firewall exclusivo, isto é, ``firewall em tudo *exceto* nas portas A, B, C, D, e M-Z". Desta maneira você pode usar firewall em todas as suas portas baixas exceto para determinados serviços específicos tais como **named** (se você é primario para uma zona), **ntalkd**, **sendmail**, e outros serviços acessíveis via Internet. Se você tentar configurar o firewall de outra maneira - como um firewall inclusivo ou permissivo, há uma boa possibilidade que você esqueça de ``fechar" um par de serviços, ou que você adicione um novo serviço interno e esqueça de atualizar o firewall. Você pode ainda abrir um range de portas altas no firewall, para permitir operações como permissivas, sem comprometer suas portas baixas. Também tome nota que o FreeBSD permite você controlar o range de portas usado por conexões dinâmicas, através de varios *net.inet.ip.portrange* do `sysctl` (`sysctl -a | fgrep portrange`), que pode também facilitar a complexidade da configuração do seu firewall. Por exemplo, você pode usar um range normal first/last de 4000 a 5000, e um range hiport de 49152 a 65535, e bloquear tudo abaixo de 4000 no seu firewall (exceto para determinadas portas de acesso específicas a Internet, naturalmente).

Um outro ataque DoS comum é chamado um ataque springboard - para atacar um servidor de uma maneira que faça o servidor gerar respostas que sobrecarregue o servidor, a rede local, ou alguma outra máquina. O ataque mais comum desta natureza é o *ICMP ping broadcast attack*. O atacante falsifica *spoofs* pacotes ping enviados para o endereço de broadcast da sua LAN com o endereço IP de origem atribuido a máquina real que ele deseja atacar. Se seus roteadores de borda não estiverem configurados para parar ping para endereços de broadcast, sua LAN enrosca gerando respostas suficientes ao endereço de origem falsificado *spoofed* para saturar a vítima, especialmente quando o atacante usa o mesmo truque em diferentes dúzias de endereços broadcast em cima de diferentes dúzias de redes de uma vez. Ataques de broadcast sobre cento e vinte megabits foram medidos. Um segundo ataque springboard comum é contra os sistemas de relatórios de erro do ICMP. Construindo pacotes que geram respostas de erro ICMP, um atacante pode saturar a rede de entrada do servidor e fazer com que o servidor sature sua rede de saída com respostas ICMP. Este tipo de ataque pode também derrubar o servidor executando mbuf's, especialmente se o servidor não pode tratar as respostas

ICMP geradas rápidas o bastante. O kernel do FreeBSD kernel tem uma nova opção de compilação chamada *ICMP\_BANDLIM* que limita a eficácia destes tipos de ataques. A última classe principal de ataques springboard esta relacionada a certos serviços **inetd** internos tais como o serviço echo udp. Um atacante simplesmente falsifica *spoofs* um pacote UDP com o endereço de origem sendo a porta echo do servidor A, e o endereço de destino sendo a porta echo do servidor B, onde o servidor A e B estão ambos em sua LAN. Os dois servidores então lançam este pacote para frente e para trás entre si. O atacante pode sobrecarregar ambos os servidores e suas LANs simplesmente injetando alguns pacotes nesta maneira. Problemas similares existem com a porta **chargen** interna. Um administrador de sistemas competente desligará todos estes serviços internos de teste do inetd.

Ataques de pacotes falsificados *spoofed* podem também ser usados para sobrecarregar o cache de rotas do kernel. Consulte os parâmetros *net.inet.ip.rtxpire*, *rtminexpire* e *rtmaxcache* do *sysctl*. Um ataque de pacotes falsificados que usa uma origem IP aleatória fara o kernel gerar um cache de rotas temporário na tabela de rotas, visível com `netstat -rna | fgrep W3`. Estas rotas tipicamente terminam em 1600 segundos ou menos. Se o kernel detectar que a tabela de rotas cacheadas começou a crescer ele dinamicamente reduz o *rtxpire* mas nunca diminui menos do que o *rtminexpire*. Há dois problemas:

1. O kernel não reage rápido o suficiente quando um servidor levemente carregado é atacado repentinamente.
2. O *rtminexpire* não é baixo o bastante para o kernel sobreviver a um ataque prolongado.

Se seus servidores estiveram conectados a Internet via um T3 ou melhor, pode ser prudente cancelar manualmente o *rtxpire* e *rtminexpire* via [sysctl\(8\)](#). Nunca ajuste o parâmetro de um dos dois para zero (a menos que você queira derrubar a máquina). Ajustar ambos os parâmetros para 2 segundos deve ser suficiente para proteger a tabela de rotas de um ataque.

### 10.3.9. Casos de Acesso com Kerberos e SSH

Há alguns casos com Kerberos e ssh que precisam ser dirigidos se você pretende usa-los. Kerberos V é um protocolo excelente de autenticação, mas há falhas nas aplicações kerberizadas **telnet** e **rlogin** que as fazem inadequadas para relacionamento com streams binários. Também, por padrão o Kerberos não criptografa uma sessão a menos que você use a opção *-x*. O **ssh** criptografa tudo por padrão.

O ssh funciona perfeitamente bem em cada respect exceto que ele envia chaves criptografadas por padrão. O que isto significa é que se você tiver uma estação de trabalho segura guardando chaves que lhe dão acesso ao resto do sistema, e você faz ssh para uma máquina insegura, suas chaves são aproveitadas. As chaves reais não são expostas, mas o ssh instala uma porta de envio para o duração do seu login, e se um atacante violar o `root` na máquina insegura ele pode utilizar essa porta para usar suas chaves para ganhar acesso a qualquer outra máquina que suas chaves abrem.

Nós recomendamos que você use o ssh em combinação com Kerberos sempre que for possível para logins do sistema. O **ssh** pode ser compilado com suporte a Kerberos. Isto

reduz sua segurança em chaves ssh potênciamente expostas enquanto ao mesmo tempo protege senhas via Kerberos. Chaves ssh devem somente ser usadas para tarefas automatizadas de máquinas seguras (algo que o Kerberos é adequado a fazer). Nós também recomendamos que você desligue o envio de chaves na configuração do ssh, ou que você faça uso da opção *from=IP/DOMAIN* que o ssh permite em seu arquivo *authorized\_keys* fazer a chave somente aproveitáveis a entidades que fazem login de máquinas específicas.

---

## 10.4. DES, MD5, e Crypt

*Parcialmente reescrito e atualizado por Bill Swingle.*

Cada usuário em um sistema UNIX® tem sua senha associada com sua conta. Parece óbvio que estas senhas precisam ser conhecidas apenas pelo usuário e o sistema operacional atual. A fim de manter estas senhas em segredo, são criptografadas com o que é conhecido como "one-way hash", isto é, podem somente ser criptografadas facilmente mas não descriptografadas. Em outras palavras, o que nós lhe dissemos a pouco, era óbvio, não é mesmo verdade: o próprio sistema operacional não sabe *realmente* a senha. Sabe somente o formulário de *criptografia* da senha. A única maneira de obter a senha em "texto puro" é procurar por força bruta no espaço de senhas possíveis.

Infelizmente a única maneira segura de criptografar senhas foi quando UNIX veio a ser baseado em DES, Data Encryption Standard. Este não era um problema para usuários residentes nos EUA, mas desde que o código fonte para o DES não pudesse ser exportado para fora dos EUA, o FreeBSD teve que encontrar uma maneira para cumprir com as leis dos EUA e manter compatibilidade com todos as outras variantes do UNIX que ainda usam DES.

A solução devia dividir as bibliotecas de criptografia de modo que os usuários dos EUA pudessem instalar as bibliotecas DES e usar o DES, mas usuários internacionais tiveram ainda um método de criptografia que pudesse ser exportado. Isto é como o FreeBSD veio a usar o MD5 como seu método padrão de criptografia. O MD5 acredita-se ser mais seguro do que o DES, assim instalando o DES é oferecido principalmente por razões de compatibilidade.

### 10.4.1. Reconhecendo Seu Mecanismo de Criptografia

Antes do FreeBSD 4.4 a `libcrypt.a` era um link simbólico apontando para a biblioteca que era usada para criptografia. O FreeBSD 4.4 mudou a `libcrypt.a` para fornecer uma biblioteca hash configuravel para autenticação de senha. Atualmente a biblioteca suporta as funções hash DES, MD5 e Blowfish. Por padrão o FreeBSD usa MD5 para senhas criptografadas.

É consideravelmente fácil identificar que método de criptografia o FreeBSD esta configurado para usar. Examinando as senhas criptografadas no arquivo

`/etc/master.passwd` é uma maneira. Senhas criptografadas com o hash MD5 são mais longas do que aquelas criptografadas com o hash DES e começam também com os caracteres `$1$`. Senhas começando com `$2$` estão criptografadas com funções hash Blowfish. Strings de senhas DES não tem nenhum detalhe identificando suas características, mas são muito mais curtas do que senhas MD5, e são codificadas em alfabeto de 64-caracteres que não inclui o carácter `$`, assim uma string relativamente pequena que não começa com um sinal de dólar é muito provável uma senha DES.

O formato de senha usado para senhas novas é controlado pela capacidade de login `passwd_format` em `/etc/login.conf`, que aceita os valores de `des`, `md5` ou `blf`. Veja as páginas de manual [login.conf\(5\)](#) para mais informação sobre capacidades de login.

## 10.5. Senhas de uso único

S/Key é um esquema de senha de uso único baseado em uma função hash de one-way. O FreeBSD usa o hash MD4 para compatibilidade mas outros sistemas tem usado MD5 e DES-MAC. O S/Key foi parte da base do sistema FreeBSD desde a versão 1.1.5 e é também usado em um número crescente de outros sistemas operacionais. S/Key é marca registrada da Bell Communications Research, Inc.

Da versão 5.0 do FreeBSD, o S/Key foi substituído com o funcionalmente equivalente OPIE (One-time Passwords In Everything). OPIE usa o hash MD5 por padrão.

Há três diferentes tipos de senhas que nós discutiremos abaixo. O primeiro é seu estilo UNIX® usual ou senha Kerberos; nós chamaremos isto de uma ``senha UNIX". O segundo tipo é a senha de uso único que é gerada pelo programa `key` S/Key ou pelo programa OPIE [opiekey\(1\)](#) e acertada pelos programas `keyinit` ou [opiepasswd\(1\)](#) e o prompt de login; nós chamaremos isto uma ``senha de uso único". O tipo final de senha é a senha secreta que você dá aos programas `key/opiekey` (e às vezes aos programas `keyinit/opiepasswd`) que se usa para gerar senhas de uso único; nós chamaremos ela de uma ``senha secreta" ou inadequadamente apenas ``senha".

A senha secreta não tem qualquer coisa com sua senha UNIX; podem ser as mesmas mas isto não é recomendado. Senhas secretas S/Key e OPIE não são limitadas a 8 caracteres como senhas antigas UNIX [\[1\]](#), podem ser tão longas o quanto você gostar. Senhas de seis ou sete frases longas são razoavelmente comuns. Para a maior parte, os sistema S/Key ou OPIE operam completamente independentes do sistema de senhas UNIX.

Além da senha, há outras duas partes de dados que são importantes para o S/Key e OPIE. Um que é conhecido como ``semente" ou ``key", consistindo em duas letras e cinco dígitos. O outro que é chamado ``contagem de iteração", um número entre 1 e 100. O S/Key cria a senha de uso único concatenando a semente e a senha secreta, então aplicando o hash MD4/MD5 tantas vezes como especificadas pela contagem de interação e voltando o resultado em seis palavras curtas em inglês. Estas seis palavras em inglês é sua senha de uso único. O sistema de autenticação (primeiramente PAM) mantem-se a par de última senha de uso único usada, e o usuário é autenticado se o hash da senha fornecida pelo usuário for a igual a senha anterior. Porque um hash de uso único é usado é impossível gerar senhas de uso único futuras se uma usada com sucesso

for capturada; a contagem de interação é decrementada após cada login bem sucedido para manter o usuário e o programa de login em sincronismo. Quando a contagem de interação passar de 1, o S/Key e OPIE devem ser reinicializados.

Há três programas envolvidos em cada sistema que nós discutiremos abaixo. Os programas `key` e `opiekey` aceitam uma contagem de interação, uma semente, e uma senha secreta, e geram uma senha de uso único ou uma lista consecutiva de senhas de uso único. Os programas `keyinit` e `opiepasswd` são usados para inicializar o S/Key e OPIE respectivamente, e trocar senhas, contagem de interação, ou sementes; aceitam um passphrase secreto, ou uma contagem de interação, semente, e senha de uso único. Os programas `keyinfo` e `opieinfo` examinam os arquivos relevantes as credenciais (`/etc/skeykeys` ou `/etc/opiekeys`) e imprimem invocando a contagem de interação atual do usuário e a semente.

Há quatro tipos diferentes de operações que nós cobriremos. O primeiro é usando `keyinit` ou `opiepasswd` sobre uma conexão segura para criar senhas de uso único pela primeira vez, ou para trocar sua senha ou semente. A segunda operação é usando `keyinit` ou `opiepasswd` sobre uma conexão insegura, em conjunto com `key` ou `opiekey` sobre uma conexão segura, para fazer o mesmo. O terceiro é usando `key/opiekey` para fazer login sobre uma conexão insegura. O quarto é usando `key` ou `opiekey` para gerar um número de chaves que podem ser escritas ou impressas para carregar com você quanto ir a algum local sem conexão segura para qualquer lugar.

### 10.5.1. Inicialização de Conexão Segura

Para inicializar o S/Key pela primeira vez, troque sua senha, ou troque sua semente enquanto logado sobre uma conexão segura (e.g., no console de uma máquina ou via `ssh`), use o comando `keyinit` sem nenhum parâmetro enquanto logado:

```
% keyinit
Adding unfurl:
Reminder - Only use this method if you are directly connected.
If you are using telnet or rlogin exit with no password and use
keyinit -s.
Enter secret password:
Again secret password:

ID unfurl s/key is 99 to17757
DEFY CLUB PRO NASH LACE SOFT
```

Para OPIE, `opiepasswd` é usado:

```
% opiepasswd -c
[grimreaper] ~ $ opiepasswd -f -c
Adding unfurl:
Only use this method from the console; NEVER from remote. If you are
using
telnet, xterm, or a dial-in, type ^C now or exit with no password.
Then run opiepasswd without the -c parameter.
Using MD5 to compute responses.
Enter new secret pass phrase:
Again new secret pass phrase:
ID unfurl OTP key is 499 to4268
```

MOS MALL GOAT ARM AVID COED

Nos prompts `Enter new secret pass phrase:` ou `Enter secret password:`, você deve entrar uma senha ou frase. Lembre-se, isto não é a senha que você usará para fazer login, é usada para gerar suas chaves login de uso único. A linha ```ID"` dá os parâmetros da sua instância particular: seu nome de login, a contagem de iteração, e semente. Ao logar o sistema recordará estes parâmetros e apresenta-os para você assim você não tem que lembra-los. A última linha dá a senha de uso único particular que corresponde a estes parâmetros e sua senha secreta; se você fosse re-logar imediatamente, esta senha de uso único é a que você usaria.

### 10.5.2. Inicialização da Conexão Insegura

Para inicializar ou trocar sua senha secreta sob uma conexão insegura, você precisará já tem uma conexão segura para algum lugar onde você pode executar o `key` ou `opiekey`; isto pode ser na forma de um acessório de desktop em um Macintosh®, ou um prompt shell em uma máquina que você confia. Você precisará também fazer uma contagem de iteração (100 é provavelmente um bom valor), e você pode fazer sua própria semente ou usar uma gerada randomicamente. Sob a conexão insegura (para a máquina que você esta inicializando), use o comando `keyinit -s`:

```
% keyinit -s
Updating unfurl:
Old key: to17758
Reminder you need the 6 English words from the key command.
Enter sequence count from 1 to 9999: 100
Enter new key [default to17759]:
s/key 100 to 17759
s/key access password:
s/key access password:CURE MIKE BANE HIM RACY GORE
```

Para o OPIE, você precisa usar `opiepasswd`:

```
% opiepasswd

Updating unfurl:
You need the response from an OTP generator.
Old secret pass phrase:
 otp-md5 498 to4268 ext
 Response: GAME GAG WELT OUT DOWN CHAT
New secret pass phrase:
 otp-md5 499 to4269
 Response: LINE PAP MILK NELL BUOY TROY

ID mark OTP key is 499 gr4269
LINE PAP MILK NELL BUOY TROY
```

Para aceitar a semente padrão (que o programa `keyinit` confusingly chama de *key*), pressione **Return**. Então antes de entrar uma senha de acesso, vá para sua conexão segura ou acessório de desk S/Key, e aplique os mesmos parâmetros:

```
% key 100 to17759
Reminder - Do not use this program while logged in via telnet or
rlogin.
```



```
Enter secret password: <secret password>
CURE MIKE BANE HIM RACY GORE
```

Ou para o OPIE:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Agora troque de volta para sua conexão insegura, e copie a senha de uso único gerada over to the relevant program.

### 10.5.3. Gerando uma Senha de Uso Único

Uma vez que você tenha inicializado o S/Key ou OPIE, quando você fizer login será apresentado com um prompt como este:

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.

FreeBSD/i386 (example.com) (ttypa)

login: <username>
s/key 97 fw13894
Password:
```

Ou para o OPIE:

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.

FreeBSD/i386 (example.com) (ttypa)

login: <username>
otp-md5 498 gr4269 ext
Password:
```

As a side note, os prompts do S/Key e OPIE tem uma característica útil (não mostrada aqui): se você pressionar **Return** no prompt de senha, o prompt irá ligar o echo, assim você pode ver o que você está digitando. Isto pode ser extremamente útil se você estiver tentando digitar uma senha manualmente, como de um impresso.

Neste momento você precisa gerar sua senha de uso único para responder este prompt de login. Isto deve ser feito em um sistema confiável em que você pode executar `key` ou `opiekey`. (Há versões destes para DOS, Windows® e Mac OS® também.) Necessitam ambos a contagem de iteração e a semente como opções de linha de comando. Você pode copiar-e-colar estes do prompt de login na máquina a que você está logando.

No sistema confiável:

```
% key 97 fw13894
Reminder - Do not use this program while logged in via telnet or
rlogin.
Enter secret password:
WELD LIP ACTS ENDS ME HAAG
```

Para OPIE:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Agora que você tem sua senha de uso único você pode continuar logando:

```
login: <username>
s/key 97 fw13894
Password: <return to enable echo>
s/key 97 fw13894
Password [echo on]: WELD LIP ACTS ENDS ME HAAG
Last login: Tue Mar 21 11:56:41 from 10.0.0.2 ...
```

#### 10.5.4. Gerando Múltiplas Senhas de Uso Único

As vezes você tem que ir a logares onde você não tem acesso a uma máquina confiável ou uma conexão segura. Neste caso, é possível usar os comandos `key` e `opiekey` para gerar de antemão um número de senhas de uso único para serem impressas e levadas com você. Por exemplo:

```
% key -n 5 30 zz99999
Reminder - Do not use this program while logged in via telnet or
rlogin.
Enter secret password: <secret password>
26: SODA RUDE LEA LIND BUDD SILT
27: JILT SPY DUTY GLOW COWL ROT
28: THEM OW COLA RUNT BONG SCOT
29: COT MASH BARR BRIM NAN FLAG
30: CAN KNEE CAST NAME FOLK BILK
```

Ou para o OPIE:

```
% opiekey -n 5 30 zz99999
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase: <secret password>
26: JOAN BORE FOSS DES NAY QUIT
27: LATE BIAS SLAY FOLK MUCH TRIG
28: SALT TIN ANTI LOON NEAL USE
29: RIO ODIN GO BYE FURY TIC
30: GREW JIVE SAN GIRD BOIL PHI
```

O `-n 5` pede cinco chaves em sequência, o `30` especifica o que o último número de iteração deve ser. Note que estes são impressos em ordem *reversa* do uso eventual. Se

you may still find it handy to scratch off passwords as you use them.

### 10.5.5. Restringindo o Uso de Senhas UNIX®

S/Key pode colocar restrições no uso de senhas UNIX baseadas no nome de host, nome de usuário, porta de terminal, ou endereço IP de uma seção de login. Estas restrições podem ser encontradas no arquivo de configuração `/etc/skey.access`. A página de manual [skey.access\(5\)](#) tem mais informação no formato completo do arquivo e também detalha alguns cuidados com segurança para se estar ciente antes dependendo deste arquivo para a segurança.

Se não existir o arquivo `/etc/skey.access` (este é o padrão em sistemas FreeBSD 4.X), então todos os usuários estarão permitidos a usar senhas UNIX. Se o arquivo existir, entretanto, então todos os usuários estarão requeridos a usar S/Key a menos que permitidos explicitamente para fazer de outra maneira por declarações de configuração no arquivo `skey.access`. Em todos os casos, senhas UNIX são permitidas no console.

Aqui esta uma amostra do arquivo de configuração `skey.access` que ilustra os três tipos mais comuns de declarações de configuração:

```
permit internet 192.168.0.0 255.255.0.0
permit user fnord
permit port ttyd0
```

A primeira linha (*permit internet*) permite os usuários cujo endereço IP de origem (que é vulnerável a spoofing) combina com o valor e máscara especificados, para usar senhas UNIX. Isto não deve ser considerado um mecanismo de segurança, mas preferivelmente, um meio de lembrar usuários autorizados que eles estão usando uma rede insegura e necessitam usar S/Key para autenticação.

A segunda linha (*permit user*) permite o nome de usuário especificado, neste caso `fnord`, usar senhas UNIX a qualquer hora. Geralmente falando, isto deve somente ser usado para pessoas que são incapazes de usar o programa `key`, como aquelas com terminais burros, ou aquelas que são mal-educadas.

A terceira linha (*permit port*) permite todos os usuários que fizerem login no terminal especificado usar senhas UNIX; isto seria usado para conexões discadas.

O OPIE pode restringir o uso das senhas UNIX baseado no endereço IP de uma seção de login como o S/Key faz. O arquivo relevante é o `/etc/opieaccess`, que é atualmente padrão no FreeBSD 5.0 e em novos sistemas. Por favor verifique [opieaccess\(5\)](#) para mais informação deste arquivo e de considerações de segurança que você deve estar ciente ao usar.

Aqui esta uma amostra do arquivo `opieaccess`:

```
permit 192.168.0.0 255.255.0.0
```

Esta linha permite os usuários cujo endereço IP de origem (que é vulnerável a spoofing) combina com o valor e a máscara especificados, a usar senhas UNIX em qualquer hora.

Se nenhuma regra no `opieaccess` combinar, o padrão é negar logins não OPIE.

## Notas

[1] Sob FreeBSD as senhas de login padrão podem ter até 128 caracteres de comprimento.

---

# 10.6. KerberosIV

*Contribuição de Mark Murray. Baseado em uma contribuição de Mark Dapoz.*

Kerberos é um sistema/protocolo adicional de rede que permite usuários se autenticar através de serviços de um servidor seguro. Serviços tais como login remoto, cópia remota, cópia segura de arquivos entre sistemas e outras tarefas de alto risco são feitas consideravelmente mais seguras e controláveis.

As seguintes instruções podem ser usadas como um guia em como configurar o Kerberos distribuído para FreeBSD. Entretanto, você deve consultar as páginas de manual relevantes para uma descrição completa.

## 10.6.1. Instalando KerberosIV

Kerberos é um componente opcional do FreeBSD. A maneira mais fácil de instalar este software é selecionando a distribuição `krb4` ou `krb5` no **sysinstall** durante a instalação inicial do FreeBSD. Isto instalará a implementação ``eBones" (KerberosIV) ou ``Heimdall" (Kerberos5) do Kerberos. Estas implementações estão incluídas porque são desenvolvidas fora dos EUA/Canadá e estavam portanto disponíveis aos proprietários de sistemas fora daqueles países durante a era de controles restritivos de exportação de código criptográfico dos EUA.

Alternativamente, a implementação do Kerberos do MIT está disponível na coleção de *ports* como [security/krb5](https://www.freebsd.org/ports/security/krb5).

## 10.6.2. Criando a Base de Dados Inicial

Isto é feito somente no servidor Kerberos. Certifique-se primeiramente que você não tem nenhuma base de dados antiga. Você pode entrar no diretório `/etc/kerberosIV` e certificar-se de que apenas os seguintes arquivos estejam presentes:

```
cd /etc/kerberosIV
ls
README krb.conf krb.realms
```

Se alguns arquivos adicionais (tais como `principal.*` ou `master_key`) existir, então use o comando `kdb_destroy` para destruir a base de dados antiga do Kerberos, ou se o Kerberos não estiver executando, simplesmente delete os arquivos extras.

Você deve agora editar os arquivos `krb.conf` e `krb.realms` para definir seu reino Kerberos. Neste caso o reino será *EXAMPLE.COM* e o servidor é `grunt.example.com`. Nós editamos ou criamos o arquivo `krb.conf`:

```
cat krb.conf
EXAMPLE.COM
EXAMPLE.COM grunt.example.com admin server
CS.BERKELEY.EDU okeeffe.berkeley.edu
ATHENA.MIT.EDU kerberos.mit.edu
ATHENA.MIT.EDU kerberos-1.mit.edu
ATHENA.MIT.EDU kerberos-2.mit.edu
ATHENA.MIT.EDU kerberos-3.mit.edu
LCS.MIT.EDU kerberos.lcs.mit.edu
TELECOM.MIT.EDU bitsy.mit.edu
ARC.NASA.GOV trident.arc.nasa.gov
```

Neste caso, os outros reinos não precisam estar aí. Eles estão aqui como um exemplo de como uma máquina pode ser feita de múltiplos reinos. Você pode desejar não incluí-los para simplificar.

A primeira linha nomeia o reino em que este sistema trabalha. As outras linhas contêm entradas reino/host. O primeiro item em uma linha é um reino, e o segundo é um host nesse reino que está agindo como um "centro de distribuição de chave". As palavras *admin server* depois de um nome de host significa que o host fornece também um servidor administrativo de base de dados. Para uma explanação adicional destes termos, consulte por favor as páginas de manual do Kerberos.

Agora nós temos que adicionar `grunt.example.com` ao reino *EXAMPLE.COM* e também adicionar uma entrada para colocar todos os hosts do domínio `.example.com` no reino *EXAMPLE.COM*. O arquivo `krb.realms` seria atualizado como segue:

```
cat krb.realms
grunt.example.com EXAMPLE.COM
.example.com EXAMPLE.COM
.berkeley.edu CS.BERKELEY.EDU
.MIT.EDU ATHENA.MIT.EDU
.mit.edu ATHENA.MIT.EDU
```

Novamente, os outros reinos não precisam estar aí. Eles estão aqui como um exemplo de como uma máquina pode ser feita de múltiplos reinos. Você pode desejar removê-los para simplificar as coisas.

A primeira linha coloca o sistema *específico* no reino nomeado. O resto das linhas mostra como to default systems of a particular subdomain to a named realm.

Now we are ready to create the database. This only needs to run on the Kerberos server (or Key Distribution Center). Issue the `kdb_init` command to do this:

```
kdb_init
```

```
Realm name [default ATHENA.MIT.EDU]: EXAMPLE.COM
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
```

```
Enter Kerberos master key:
```

Now we have to save the key so that servers on the local machine can pick it up. Use the `kstash` command to do this:

```
kstash
Enter Kerberos master key:
Current Kerberos master key version is 1.
Master key entered. BEWARE!
```

This saves the encrypted master password in `/etc/kerberosIV/master_key`.

### 10.6.3. Making It All Run

Two principals need to be added to the database for *each* system that will be secured with Kerberos. Their names are `kpasswd` and `rcmd`. These two principals are made for each system, with the instance being the name of the individual system.

These daemons, **kpasswd** and **rcmd** allow other systems to change Kerberos passwords and run commands like [rccp\(1\)](#), [rlogin\(1\)](#) and [rsh\(1\)](#).

Now let us add these entries:

```
kdb_edit
Opening database...

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered. BEWARE!
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: passwd
Instance: grunt

<Not found>, Create [y] ? y

Principal: passwd, Instance: grunt, kdc_key_ver: 1
New Password: <---- enter RANDOM here
Verifying password

New Password: <---- enter RANDOM here

Random password [y] ? y

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [2000-01-01] ?
Max ticket lifetime (*5 minutes) [255] ?
```

```

Attributes [0] ?
Edit O.K.
Principal name: rcmd
Instance: grunt

<Not found>, Create [y] ?

Principal: rcmd, Instance: grunt, kdc_key_ver: 1
New Password: <---- enter RANDOM here
Verifying password

New Password: <---- enter RANDOM here

Random password [y] ?

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [2000-01-01] ?
Max ticket lifetime (*5 minutes) [255] ?
Attributes [0] ?
Edit O.K.
Principal name: <---- null entry here will cause an exit

```

#### 10.6.4. Creating the Server File

We now have to extract all the instances which define the services on each machine. For this we use the `ext_srvtab` command. This will create a file which must be copied or moved *by secure means* to each Kerberos client's `/etc/kerberosIV` directory. This file must be present on each server and client, and is crucial to the operation of Kerberos.

```

ext_srvtab grunt
Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered. BEWARE!
Generating 'grunt-new-srvtab'....

```

Now, this command only generates a temporary file which must be renamed to `srvtab` so that all the servers can pick it up. Use the [mv\(1\)](#) command to move it into place on the original system:

```
mv grunt-new-srvtab srvtab
```

If the file is for a client system, and the network is not deemed safe, then copy the `client-new-srvtab` to removable media and transport it by secure physical means. Be sure to rename it to `srvtab` in the client's `/etc/kerberosIV` directory, and make sure it is mode 600:

```
mv grumble-new-srvtab srvtab
chmod 600 srvtab
```

#### 10.6.5. Populating the Database

We now have to add some user entries into the database. First let us create an entry for the user `jane`. Use the `kdb_edit` command to do this:

```

kdb_edit
Opening database...

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered. BEWARE!
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: jane
Instance:

<Not found>, Create [y] ? y

Principal: jane, Instance: , kdc_key_ver: 1
New Password: <---- enter a secure password here
Verifying password

New Password: <---- re-enter the password here
Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [2000-01-01] ?
Max ticket lifetime (*5 minutes) [255] ?
Attributes [0] ?
Edit O.K.
Principal name: <---- null entry here will cause an exit

```

### 10.6.6. Testing It All Out

First we have to start the Kerberos daemons. Note that if you have correctly edited your `/etc/rc.conf` then this will happen automatically when you reboot. This is only necessary on the Kerberos server. Kerberos clients will automatically get what they need from the `/etc/kerberosIV` directory.

```

kerberos &
Kerberos server starting
Sleep forever on error
Log file is /var/log/kerberos.log
Current Kerberos master key version is 1.

Master key entered. BEWARE!

Current Kerberos master key version is 1
Local realm: EXAMPLE.COM
kadmind -n &
KADM Server KADM0.0A initializing
Please do not use 'kill -9' to kill this job, use a
regular kill instead

Current Kerberos master key version is 1.

Master key entered. BEWARE!

```

Now we can try using the `kinit` command to get a ticket for the ID `jane` that we created above:

```

% kinit jane
MIT Project Athena (grunt.example.com)

```



```
Kerberos Initialization for "jane"
Password:
```

Try listing the tokens using `klist` to see if we really have them:

```
% klist
Ticket file: /tmp/tkt245
Principal: jane@EXAMPLE.COM

 Issued Expires Principal
Apr 30 11:23:22 Apr 30 19:23:22 krbtgt.EXAMPLE.COM@EXAMPLE.COM
```

Now try changing the password using [passwd\(1\)](#) to check if the `kpasswd` daemon can get authorization to the Kerberos database:

```
% passwd
realm EXAMPLE.COM
Old password for jane:
New Password for jane:
Verifying password
New Password for jane:
Password changed.
```

### 10.6.7. Adding `su` Privileges

Kerberos allows us to give *each* user who needs `root` privileges their own *separate* [su\(1\)](#) password. We could now add an ID which is authorized to [su\(1\)](#) to `root`. This is controlled by having an instance of `root` associated with a principal. Using `kdb_edit` we can create the entry `jane.root` in the Kerberos database:

```
kdb_edit
Opening database...

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered. BEWARE!
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: jane
Instance: root

<Not found>, Create [y] ? y

Principal: jane, Instance: root, kdc_key_ver: 1
New Password: <---- enter a SECURE password here
Verifying password

New Password: <---- re-enter the password here

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [2000-01-01] ?
Max ticket lifetime (*5 minutes) [255] ? 12 <--- Keep this short!
Attributes [0] ?
Edit O.K.
```

```
Principal name: <---- null entry here will cause an
exit
```

Now try getting tokens for it to make sure it works:

```
kinit jane.root
MIT Project Athena (grunt.example.com)
Kerberos Initialization for "jane.root"
Password:
```

Now we need to add the user to root's .klogin file:

```
cat /root/.klogin
jane.root@EXAMPLE.COM
```

Now try doing the [su\(1\)](#):

```
% su
Password:
```

and take a look at what tokens we have:

```
klist
Ticket file: /tmp/tkt_root_245
Principal: jane.root@EXAMPLE.COM

 Issued Expires Principal
May 2 20:43:12 May 3 04:43:12 krbtgt.EXAMPLE.COM@EXAMPLE.COM
```

## 10.6.8. Using Other Commands

In an earlier example, we created a principal called *jane* with an instance *root*. This was based on a user with the same name as the principal, and this is a Kerberos default; that a *<principal>.<instance>* of the form *<username>.root* will allow that *<username>* to [su\(1\)](#) to *root* if the necessary entries are in the *.klogin* file in *root*'s home directory:

```
cat /root/.klogin
jane.root@EXAMPLE.COM
```

Likewise, if a user has in their own home directory lines of the form:

```
% cat ~/.klogin
jane@EXAMPLE.COM
jack@EXAMPLE.COM
```

This allows anyone in the *EXAMPLE.COM* realm who has authenticated themselves as *jane* or *jack* (via *kinit*, see above) to access to *jane*'s account or files on this system (*grunt*) via [rlogin\(1\)](#), [rsh\(1\)](#) or [rcp\(1\)](#).

For example, *jane* now logs into another system using Kerberos:

```
% kinit
MIT Project Athena (grunt.example.com)
```

```
Password:
% rlogin grunt
Last login: Mon May 1 21:14:47 from grumble
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
 The Regents of the University of California. All rights
reserved.
```

```
FreeBSD BUILT-19950429 (GR386) #0: Sat Apr 29 17:50:09 SAT 1995
```

Or jack logs into jane's account on the same machine (jane having set up the .klogin file as above, and the person in charge of Kerberos having set up principal *jack* with a null instance):

```
% kinit
% rlogin grunt -l jane
MIT Project Athena (grunt.example.com)
Password:
Last login: Mon May 1 21:16:55 from grumble
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
 The Regents of the University of California. All rights
reserved.
FreeBSD BUILT-19950429 (GR386) #0: Sat Apr 29 17:50:09 SAT 1995
```

---

## 10.7. Kerberos5

*Contribuição de Tillman Hodgson. Based on a contribution by Mark Murray.*

Every FreeBSD release beyond FreeBSD-5.1 includes support only for **Kerberos5**. Hence **Kerberos5** is the only version included, and its configuration is similar in many aspects to that of **KerberosIV**. The following information only applies to **Kerberos5** in post FreeBSD-5.0 releases. Users who wish to use the **KerberosIV** package may install the [security/krb4](#) port.

**Kerberos** is a network add-on system/protocol that allows users to authenticate themselves through the services of a secure server. Services such as remote login, remote copy, secure inter-system file copying and other high-risk tasks are made considerably safer and more controllable.

**Kerberos** can be described as an identity-verifying proxy system. It can also be described as a trusted third-party authentication system. **Kerberos** provides only one function -- the secure authentication of users on the network. It does not provide authorization functions (what users are allowed to do) or auditing functions (what those users did). After a client and server have used **Kerberos** to prove their identity, they can also encrypt all of their communications to assure privacy and data integrity as they go about their business.

Therefore it is highly recommended that **Kerberos** be used with other security methods which provide authorization and audit services.

The following instructions can be used as a guide on how to set up **Kerberos** as distributed for FreeBSD. However, you should refer to the relevant manual pages for a complete description.

For purposes of demonstrating a **Kerberos** installation, the various namespaces will be handled as follows:

- The DNS domain ("zone") will be example.org.
- The **Kerberos** realm will be EXAMPLE.ORG.

**Nota:** Please use real domain names when setting up **Kerberos** even if you intend to run it internally. This avoids DNS problems and assures inter-operation with other **Kerberos** realms.

### 10.7.1. History

**Kerberos** was created by MIT as a solution to network security problems. The **Kerberos** protocol uses strong cryptography so that a client can prove its identity to a server (and vice versa) across an insecure network connection.

**Kerberos** is both the name of a network authentication protocol and an adjective to describe programs that implement the program (**Kerberos** telnet, for example). The current version of the protocol is version 5, described in RFC 1510.

Several free implementations of this protocol are available, covering a wide range of operating systems. The Massachusetts Institute of Technology (MIT), where **Kerberos** was originally developed, continues to develop their **Kerberos** package. It is commonly used in the US as a cryptography product, as such it has historically been affected by US export regulations. The MIT **Kerberos** is available as a port ([security/krb5](#)). Heimdal **Kerberos** is another version 5 implementation, and was explicitly developed outside of the US to avoid export regulations (and is thus often included in non-commercial UNIX® variants). The Heimdal **Kerberos** distribution is available as a port ([security/heimdal](#)), and a minimal installation of it is included in the base FreeBSD install.

In order to reach the widest audience, these instructions assume the use of the Heimdal distribution included in FreeBSD.

### 10.7.2. Setting up a Heimdal KDC

The Key Distribution Center (KDC) is the centralized authentication service that **Kerberos** provides -- it is the computer that issues **Kerberos** tickets. The KDC is considered "trusted" by all other computers in the **Kerberos** realm, and thus has heightened security concerns.

Note that while running the **Kerberos** server requires very few computing resources, a dedicated machine acting only as a KDC is recommended for security reasons.

To begin setting up a KDC, ensure that your `/etc/rc.conf` file contains the correct settings to act as a KDC (you may need to adjust paths to reflect your own system):

```
kerberos5_server_enable="YES"
kadmind5_server_enable="YES"
kerberos_stash="YES"
```

**Nota:** The `kerberos_stash` is only available in FreeBSD 4.X.

Next we will set up your **Kerberos** config file, `/etc/krb5.conf`:

```
[libdefaults]
 default_realm = EXAMPLE.ORG
[realms]
 EXAMPLE.ORG = {
 kdc = kerberos.example.org
 }
[domain_realm]
 .example.org = EXAMPLE.ORG
```

Note that this `/etc/krb5.conf` file implies that your KDC will have the fully-qualified hostname of `kerberos.example.org`. You will need to add a CNAME (alias) entry to your zone file to accomplish this if your KDC has a different hostname.

**Nota:** For large networks with a properly configured BIND DNS server, the above example could be trimmed to:

```
[libdefaults]
 default_realm = EXAMPLE.ORG
```

With the following lines being appended to the `example.org` zonefile:

```
_kerberos._udp IN SRV 01 00 88 kerberos.example.org.
_kerberos._tcp IN SRV 01 00 88 kerberos.example.org.
_kpasswd._udp IN SRV 01 00 464 kerberos.example.org.
_kerberos-adm._tcp IN SRV 01 00 749 kerberos.example.org.
_kerberos IN TXT EXAMPLE.ORG.
```

Next we will create the **Kerberos** database. This database contains the keys of all principals encrypted with a master password. You are not required to remember this password, it will be stored in a file (`/var/heimdal/m-key`). To create the master key, run `kstash` and enter a password.

Once the master key has been created, you can initialize the database using the `kadmin` program with the `-l` option (standing for `local`). This option instructs `kadmin` to modify the database files directly rather than going through the `kadmind` network service. This handles the chicken-and-egg problem of trying to connect to the database before it is created. Once you have the `kadmin` prompt, use the `init` command to create your realms initial database.

Lastly, while still in `kadmin`, create your first principal using the `add` command. Stick to the defaults options for the principal for now, you can always change them later with

the `modify` command. Note that you can use the `?` command at any prompt to see the available options.

A sample database creation session is shown below:

```
kstash
 Master key: xxxxxxxxx
 Verifying password - Master key: xxxxxxxxx

kadmin -l
kadmin> init EXAMPLE.ORG
Realm max ticket life [unlimited]:
kadmin> add tillman
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Attributes []:
Password: xxxxxxxxx
Verifying password - Password: xxxxxxxxx
```

Now it is time to start up the KDC services. Run `/etc/rc.d/kerberos start` and `/etc/rc.d/kadmind start` to bring up the services. Note that you won't have any kerberized daemons running at this point but you should be able to confirm that the KDC is functioning by obtaining and listing a ticket for the principal (user) that you just created from the command-line of the KDC itself:

```
%k5init tillman
tillman@EXAMPLE.ORG's Password:

%k5list
Credentials cache: FILE:/tmp/krb5cc_500
Principal: tillman@EXAMPLE.ORG

 Issued Expires Principal
krbtgt/EXAMPLE.ORG@EXAMPLE.ORG
 Aug 27 15:37:58 Aug 28 01:37:58
krbtgt/EXAMPLE.ORG@EXAMPLE.ORG
 Aug 27 15:37:58 Aug 28 01:37:58
krbtgt/EXAMPLE.ORG@EXAMPLE.ORG

v4-ticket file: /tmp/tkt500
k5list: No ticket file (tf_util)
```

### 10.7.3. Kerberos enabling a server with Heimdal services

First, we need a copy of the **Kerberos** configuration file, `/etc/krb5.conf`. To do so, simply copy it over to the client computer from the KDC in a secure fashion (using network utilities, such as [scp\(1\)](#), or physically via a floppy disk).

Next you need a `/etc/krb5.keytab` file. This is the major difference between a server providing **Kerberos** enabled daemons and a workstation -- the server must have a `keytab` file. This file contains the servers host key, which allows it and the KDC to verify each others identity. It must be transmitted to the server in a secure fashion, as the security of the server can be broken if the key is made public. This explicitly means that transferring it via a clear text channel, such as FTP, is a very bad idea.

Typically, you transfer to the `keytab` to the server using the `kadmin` program. This is handy because you also need to create the host principal (the KDC end of the `krb5.keytab`) using `kadmin`.

Note that you must have already obtained a ticket and that this ticket must be allowed to use the `kadmin` interface in the `kadmin.acl`. See the section titled "Remote administration" in the Heimdal info pages (`info heimdal`) for details on designing access control lists. If you do not want to enable remote `kadmin` access, you can simply securely connect to the KDC (via local console, [ssh\(1\)](#) or **Kerberos** [telnet\(1\)](#)) and perform administration locally using `kadmin -l`.

After installing the `/etc/krb5.conf` file, you can use `kadmin` from the **Kerberos** server. The `add --random-key` command will let you add the servers host principal, and the `ext` command will allow you to extract the servers host principal to its own keytab. For example:

```
kadmin
kadmin> add --random-key host/myserver.EXAMPLE.ORG
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Attributes []:
kadmin> ext host/myserver.EXAMPLE.ORG
kadmin> exit
```

Note that the `ext` command (short for "extract") stores the extracted key in `/etc/krb5.keytab` by default.

If you do not have `kadmin` running on the KDC (possibly for security reasons) and thus do not have access to `kadmin` remotely, you can add the host principal (`host/myserver.EXAMPLE.ORG`) directly on the KDC and then extract it to a temporary file (to avoid over-writing the `/etc/krb5.keytab` on the KDC) using something like this:

```
kadmin
kadmin> ext --keytab=/tmp/example.keytab host/myserver.example.org
kadmin> exit
```

You can then securely copy the keytab to the server computer (using `scp` or a floppy, for example). Be sure to specify a non-default keytab name to avoid over-writing the keytab on the KDC.

At this point your server can communicate with the KDC (due to its `krb5.conf` file) and it can prove its own identity (due to the `krb5.keytab` file). It is now ready for you to enable some **Kerberos** services. For this example we will enable the `telnet` service by putting a line like this into your `/etc/inetd.conf` and then restarting the [inetd\(8\)](#) service with `/etc/rc.d/inetd restart`:

```
telnet stream tcp nowait root /usr/libexec/telnetd
telnetd -a user
```

The critical bit is that the `-a` (for authentication) type is set to `user`. Consult the [telnetd\(8\)](#) manual page for more details.

#### 10.7.4. Kerberos enabling a client with Heimdal

Setting up a client computer is almost trivially easy. As far as **Kerberos** configuration goes, you only need the **Kerberos** configuration file, located at `/etc/krb5.conf`. Simply securely copy it over to the client computer from the KDC.

Test your client computer by attempting to use `kinit`, `klist`, and `kdestroy` from the client to obtain, show, and then delete a ticket for the principal you created above. You should also be able to use **Kerberos** applications to connect to **Kerberos** enabled servers, though if that does not work and obtaining a ticket does the problem is likely with the server and not with the client or the KDC.

When testing an application like `telnet`, try using a packet sniffer (such as [tcpdump\(1\)](#)) to confirm that your password is not sent in the clear. Try using `telnet` with the `-x` option, which encrypts the entire data stream (similar to `ssh`).

The core **Kerberos** client applications (traditionally named `kinit`, `klist`, `kdestroy`, and `kpasswd`) are installed in the base FreeBSD install. Note that FreeBSD versions prior to 5.0 renamed them to `k5init`, `k5list`, `k5destroy`, `k5passwd`, and `k5stash` (though it is typically only used once).

Various non-core **Kerberos** client applications are also installed by default. This is where the "minimal" nature of the base Heimdal installation is felt: `telnet` is the only **Kerberos** enabled service.

The Heimdal port adds some of the missing client applications: **Kerberos** enabled versions of `ftp`, `rsh`, `rcp`, `rlogin`, and a few other less common programs. The MIT port also contains a full suite of **Kerberos** client applications.

#### 10.7.5. User configuration files: `.k5login` and `.k5users`

Users within a realm typically have their **Kerberos** principal (such as `tillman@EXAMPLE.ORG`) mapped to a local user account (such as a local account named `tillman`). Client applications such as `telnet` usually do not require a user name or a principal.

Occasionally, however, you want to grant access to a local user account to someone who does not have a matching **Kerberos** principal. For example, `tillman@EXAMPLE.ORG` may need access to the local user account `webdevelopers`. Other principals may also need access to that local account.

The `.k5login` and `.k5users` files, placed in a user's home directory, can be used similar to a powerful combination of `.hosts` and `.rhosts`, solving this problem. For example, if a `.k5login` with the following contents:

```
tillman@example.org
 jdoe@example.org
```



Were to be placed into the home directory of the local user `webdevelopers` then both principals listed would have access to that account without requiring a shared password.

Reading the man pages for these commands is recommended. Note that the `ksu` man page covers `.k5users`.

### 10.7.6. Kerberos Tips, Tricks, and Troubleshooting

- When using either the Heimdal or MIT **Kerberos** ports ensure that your `PATH` environment variable lists the **Kerberos** versions of the client applications before the system versions.
- Is your time in sync? Are you sure? If the time is not in sync (typically within five minutes) authentication will fail.
- MIT and Heimdal inter-operate nicely. Except for `kadmin`, the protocol for which is not standardized.
- If you change your hostname, you also need to change your `host/` principal and update your keytab. This also applies to special keytab entries like the `www/` principal used for Apache's [www/mod\\_auth\\_kerb](#).
- All hosts in your realm must be resolvable (both forwards and reverse) in DNS (or `/etc/hosts` as a minimum). CNAMEs will work, but the A and PTR records must be correct and in place. The error message isn't very intuitive:  
``Kerberos5 refuses authentication because Read req failed: Key table entry not found"`.
- Some operating systems that may be acting as clients to your KDC do not set the permissions for `ksu` to be `setuid root`. This means that `ksu` does not work, which is a good security idea but annoying. This is not a KDC error.
- With MIT **Kerberos**, if you want to allow a principal to have a ticket life longer than the default ten hours, you must use `modify_principal` in `kadmin` to change the `maxlife` of both the principal in question and the `krbtgt` principal. Then the principal can use the `-l` option with `kinit` to request a ticket with a longer lifetime.
- **Nota:** If you run a packet sniffer on your KDC to add in troubleshooting and then run `kinit` from a workstation, you will notice that your TGT is sent immediately upon running `kinit` -- even before you type your password! The explanation is that the **Kerberos** server freely transmits a TGT (Ticket Granting Ticket) to any unauthorized request; however, every TGT is encrypted in a key derived from the user's password. Therefore, when a user types their password it is not being sent to the KDC, it is being used to decrypt the TGT that `kinit` already obtained. If the decryption process results in a valid ticket with a valid time stamp, the user has valid **Kerberos** credentials. These credentials include a session key for establishing secure communications with the **Kerberos** server in the future, as well as the actual ticket-granting ticket, which is actually encrypted with the **Kerberos** server's own key. This second layer of encryption is unknown to the user, but it is what allows the **Kerberos** server to verify the authenticity of each TGT.
- You have to keep the time in sync between all the computers in your realm. NTP is perfect for this. For more information on NTP, see [Seção 19.12](#).
- If you want to use long ticket lifetimes (a week, for example) and you are using **OpenSSH** to connect to the machine where your ticket is stored, make sure that

**Kerberos** *TicketCleanup* is set to *no* in your `sshd_config` or else your tickets will be deleted when you log out.

- Remember that host principals can have a longer ticket lifetime as well. If your user principal has a lifetime of a week but the host you are connecting to has a lifetime of nine hours, you will have an expired host principal in your cache and the ticket cache will not work as expected.
- When setting up a `krb5.dict` file to prevent specific bad passwords from being used (the manual page for `kadmin` covers this briefly), remember that it only applies to principals that have a password policy assigned to them. The `krb5.dict` files format is simple: one string per line. Creating a symbolic link to `/usr/share/dict/words` might be useful.

### 10.7.7. Differences with the MIT port

The major difference between the MIT and Heimdal installs relates to the `kadmin` program which has a different (but equivalent) set of commands and uses a different protocol. This has a large implications if your KDC is MIT as you will not be able to use the Heimdal `kadmin` program to administer your KDC remotely (or vice versa, for that matter).

The client applications may also take slightly different command line options to accomplish the same tasks. Following the instructions on the MIT **Kerberos** web site (<http://web.mit.edu/Kerberos/www/>) is recommended. Be careful of path issues: the MIT port installs into `/usr/local/` by default, and the "normal" system applications may be run instead of MIT if your `PATH` environment variable lists the system directories first.

**Nota:** With the MIT [security/krb5](http://web.mit.edu/Kerberos/www/) port that is provided by FreeBSD, be sure to read the `/usr/local/share/doc/krb5/README.FreeBSD` file installed by the port if you want to understand why logins via `telnetd` and `klogind` behave somewhat oddly. Most importantly, correcting the "incorrect permissions on cache file" behavior requires that the `login.krb5` binary be used for authentication so that it can properly change ownership for the forwarded credentials.

### 10.7.8. Mitigating limitations found in Kerberos

#### 10.7.8.1. Kerberos is an all-or-nothing approach

Every service enabled on the network must be modified to work with **Kerberos** (or be otherwise secured against network attacks) or else the users credentials could be stolen and re-used. An example of this would be **Kerberos** enabling all remote shells (via `rsh` and `telnet`, for example) but not converting the POP3 mail server which sends passwords in plaintext.

#### 10.7.8.2. Kerberos is intended for single-user workstations

In a multi-user environment, **Kerberos** is less secure. This is because it stores the tickets in the `/tmp` directory, which is readable by all users. If a user is sharing a

computer with several other people simultaneously (i.e. multi-user), it is possible that the user's tickets can be stolen (copied) by another user.

This can be overcome with the `-c filename` command-line option or (preferably) the `KRB5CCNAME` environment variable, but this is rarely done. In principal, storing the ticket in the users home directory and using simple file permissions can mitigate this problem.

### 10.7.8.3. The KDC is a single point of failure

By design, the KDC must be as secure as the master password database is contained on it. The KDC should have absolutely no other services running on it and should be physically secured. The danger is high because **Kerberos** stores all passwords encrypted with the same key (the ``master" key), which in turn is stored as a file on the KDC.

As a side note, a compromised master key is not quite as bad as one might normally fear. The master key is only used to encrypt the **Kerberos** database and as a seed for the random number generator. As long as access to your KDC is secure, an attacker cannot do much with the master key.

Additionally, if the KDC is unavailable (perhaps due to a denial of service attack or network problems) the network services are unusable as authentication can not be performed, a recipe for a denial-of-service attack. This can be alleviated with multiple KDCs (a single master and one or more slaves) and with careful implementation of secondary or fall-back authentication (PAM is excellent for this).

### 10.7.8.4. Kerberos Shortcomings

**Kerberos** allows users, hosts and services to authenticate between themselves. It does not have a mechanism to authenticate the KDC to the users, hosts or services. This means that a trojanned `kinit` (for example) could record all user names and passwords. Something like [security/tripwire](#) or other file system integrity checking tools can alleviate this.

## 10.7.9. Resources and further information

- [The Kerberos FAQ](#)
- [Designing an Authentication System: a Dialogue in Four Scenes](#)
- [RFC 1510, The \*\*Kerberos\*\* Network Authentication Service \(V5\)](#)
- [MIT \*\*Kerberos\*\* home page](#)
- [Heimdal \*\*Kerberos\*\* home page](#)

## 10.8. *Firewalls*

*Contribuição de Gary Palmer e Alex Nash.*

Firewalls are an area of increasing interest for people who are connected to the Internet, and are even finding applications on private networks to provide enhanced security. This section will hopefully explain what firewalls are, how to use them, and how to use the facilities provided in the FreeBSD kernel to implement them.

**Nota:** People often think that having a firewall between your internal network and the ``Big Bad Internet" will solve all your security problems. It may help, but a poorly set up firewall system is more of a security risk than not having one at all. A firewall can add another layer of security to your systems, but it cannot stop a really determined cracker from penetrating your internal network. If you let internal security lapse because you believe your firewall to be impenetrable, you have just made the crackers job that much easier.

### 10.8.1. What Is a Firewall?

There are currently two distinct types of firewalls in common use on the Internet today. The first type is more properly called a *packet filtering router*. This type of firewall utilizes a multi-homed machine and a set of rules to determine whether to forward or block individual packets. A multi-homed machine is simply a device with multiple network interfaces. The second type, known as a *proxy server*, relies on daemons to provide authentication and to forward packets, possibly on a multi-homed machine which has kernel packet forwarding disabled.

Sometimes sites combine the two types of firewalls, so that only a certain machine (known as a *bastion host*) is allowed to send packets through a packet filtering router onto an internal network. Proxy services are run on the bastion host, which are generally more secure than normal authentication mechanisms.

FreeBSD comes with a kernel packet filter (known as IPFW), which is what the rest of this section will concentrate on. Proxy servers can be built on FreeBSD from third party software, but there is such a variety of proxy servers available that it would be impossible to cover them in this section.

#### 10.8.1.1. Packet Filtering Routers

A router is a machine which forwards packets between two or more networks. A packet filtering router is programmed to compare each packet to a list of rules before deciding if it should be forwarded or not. Most modern IP routing software includes packet filtering functionality that defaults to forwarding all packets. To enable the filters, you need to define a set of rules.

To decide whether a packet should be passed on, the firewall looks through its set of rules for a rule which matches the contents of the packet's headers. Once a match is found, the rule action is obeyed. The rule action could be to drop the packet, to forward the packet, or even to send an ICMP message back to the originator. Only the first

match counts, as the rules are searched in order. Hence, the list of rules can be referred to as a "rule chain".

The packet-matching criteria varies depending on the software used, but typically you can specify rules which depend on the source IP address of the packet, the destination IP address, the source port number, the destination port number (for protocols which support ports), or even the packet type (UDP, TCP, ICMP, etc).

### 10.8.1.2. Proxy Servers

Proxy servers are machines which have had the normal system daemons (**telnetd**, **ftpd**, etc) replaced with special servers. These servers are called *proxy servers*, as they normally only allow onward connections to be made. This enables you to run (for example) a proxy **telnet** server on your firewall host, and people can **telnet** in to your firewall from the outside, go through some authentication mechanism, and then gain access to the internal network (alternatively, proxy servers can be used for signals coming from the internal network and heading out).

Proxy servers are normally more secure than normal servers, and often have a wider variety of authentication mechanisms available, including "one-shot" password systems so that even if someone manages to discover what password you used, they will not be able to use it to gain access to your systems as the password expires immediately after the first use. As they do not actually give users access to the host machine, it becomes a lot more difficult for someone to install backdoors around your security system.

Proxy servers often have ways of restricting access further, so that only certain hosts can gain access to the servers. Most will also allow the administrator to specify which users can talk to which destination machines. Again, what facilities are available depends largely on what proxy software you choose.

### 10.8.2. What Does IPFW Allow Me to Do?

IPFW, the software supplied with FreeBSD, is a packet filtering and accounting system which resides in the kernel, and has a user-land control utility, [ipfw\(8\)](#). Together, they allow you to define and query the rules used by the kernel in its routing decisions.

There are two related parts to IPFW. The firewall section performs packet filtering. There is also an IP accounting section which tracks usage of the router, based on rules similar to those used in the firewall section. This allows the administrator to monitor how much traffic the router is getting from a certain machine, or how much WWW traffic it is forwarding, for example.

As a result of the way that IPFW is designed, you can use IPFW on non-router machines to perform packet filtering on incoming and outgoing connections. This is a special case of the more general use of IPFW, and the same commands and techniques should be used in this situation.

### 10.8.3. Enabling IPFW on FreeBSD

As the main part of the IPFW system lives in the kernel, you will need to add one or more options to your kernel configuration file, depending on what facilities you want, and recompile your kernel. See "Reconfiguring your Kernel" ([Capítulo 9](#)) for more details on how to recompile your kernel.

**Atenção** IPFW defaults to a policy of *deny ip from any to any*. If you do not add other rules during startup to allow access, *you will lock yourself out* of the server upon rebooting into a firewall-enabled kernel. We suggest that you set *firewall\_type=open* in your `/etc/rc.conf` file when first enabling this feature, then refining the firewall rules in `/etc/rc.firewall` after you have tested that the new kernel feature works properly. To be on the safe side, you may wish to consider performing the initial firewall configuration from the local console rather than via **ssh**. Another option is to build a kernel using both the `IPFIREWALL` and `IPFIREWALL_DEFAULT_TO_ACCEPT` options. This will change the default rule of IPFW to *allow ip from any to any* and avoid the possibility of a lockout.

There are currently four kernel configuration options relevant to IPFW:

*options IPFIREWALL*

Compiles into the kernel the code for packet filtering.

*options IPFIREWALL\_VERBOSE*

Enables code to allow logging of packets through [syslogd\(8\)](#). Without this option, even if you specify that packets should be logged in the filter rules, nothing will happen.

*options IPFIREWALL\_VERBOSE\_LIMIT=10*

Limits the number of packets logged through [syslogd\(8\)](#) on a per entry basis. You may wish to use this option in hostile environments in which you want to log firewall activity, but do not want to be open to a denial of service attack via syslog flooding.

When a chain entry reaches the packet limit specified, logging is turned off for that particular entry. To resume logging, you will need to reset the associated counter using the [ipfw\(8\)](#) utility:

```
ipfw zero 4500
```

Where 4500 is the chain entry you wish to continue logging.

*options IPFIREWALL\_DEFAULT\_TO\_ACCEPT*

This changes the default rule action from `deny` to `allow`. This avoids the possibility of locking yourself out if you happen to boot a kernel with

*IPFIREWALL* support but have not configured your firewall yet. It is also very useful if you often use [ipfw\(8\)](#) as a filter for specific problems as they arise. Use with care though, as this opens up the firewall and changes the way it works.

**Nota:** Previous versions of FreeBSD contained an *IPFIREWALL\_ACCT* option. This is now obsolete as the firewall code automatically includes accounting facilities.

#### 10.8.4. Configuring IPFW

The configuration of the IPFW software is done through the [ipfw\(8\)](#) utility. The syntax for this command looks quite complicated, but it is relatively simple once you understand its structure.

There are currently four different command categories used by the utility: addition/deletion, listing, flushing, and clearing. Addition/deletion is used to build the rules that control how packets are accepted, rejected, and logged. Listing is used to examine the contents of your rule set (otherwise known as the chain) and packet counters (accounting). Flushing is used to remove all entries from the chain. Clearing is used to zero out one or more accounting entries.

##### 10.8.4.1. Altering the IPFW Rules

The syntax for this form of the command is:

```
ipfw [-N] command [index] action [log] protocol addresses [options]
```

There is one valid flag when using this form of the command:

-N

Resolve addresses and service names in output.

The *command* given can be shortened to the shortest unique form. The valid *commands* are:

add

Add an entry to the firewall/accounting rule list

delete

Delete an entry from the firewall/accounting rule list

Previous versions of IPFW used separate firewall and accounting entries. The present version provides packet accounting with each firewall entry.

If an *index* value is supplied, it is used to place the entry at a specific point in the chain. Otherwise, the entry is placed at the end of the chain at an index 100 greater than the last chain entry (this does not include the default policy, rule 65535, deny).

The *log* option causes matching rules to be output to the system console if the kernel was compiled with *IPFIREWALL\_VERBOSE*.

Valid *actions* are:

reject

Drop the packet, and send an ICMP host or port unreachable (as appropriate) packet to the source.

allow

Pass the packet on as normal. (aliases: *pass*, *permit*, and *accept*)

deny

Drop the packet. The source is not notified via an ICMP message (thus it appears that the packet never arrived at the destination).

count

Update packet counters but do not allow/deny the packet based on this rule. The search continues with the next chain entry.

Each *action* will be recognized by the shortest unambiguous prefix.

The *protocols* which can be specified are:

all

Matches any IP packet

icmp

Matches ICMP packets

tcp

Matches TCP packets

udp

Matches UDP packets

The *address* specification is:



from *address/mask [port]* to *address/mask [port] [via interface]*

You can only specify *port* in conjunction with *protocols* which support ports (UDP and TCP).

The *via* is optional and may specify the IP address or domain name of a local IP interface, or an interface name (e.g. *ed0*) to match only packets coming through this interface. Interface unit numbers can be specified with an optional wildcard. For example, *ppp\** would match all kernel PPP interfaces.

The syntax used to specify an *address/mask* is:

*address*

or

*address/mask-bits*

or

*address:mask-pattern*

A valid hostname may be specified in place of the IP address. *mask-bits* is a decimal number representing how many bits in the address mask should be set. e.g. specifying *192.216.222.1/24* will create a mask which will allow any address in a class C subnet (in this case, *192.216.222*) to be matched. *mask-pattern* is an IP address which will be logically AND'ed with the address given. The keyword *any* may be used to specify ``any IP address''.

The port numbers to be blocked are specified as:

*port [,port [,port [...]]]*

to specify either a single port or a list of ports, or

*port-port*

to specify a range of ports. You may also combine a single range with a list, but the range must always be specified first.

The *options* available are:

*frag*

Matches if the packet is not the first fragment of the datagram.

*in*

Matches if the packet is on the way in.

*out*

Matches if the packet is on the way out.

`ipoptions spec`

Matches if the IP header contains the comma separated list of options specified in *spec*. The supported IP options are: *ssrr* (strict source route), *lsrr* (loose source route), *rr* (record packet route), and *ts* (time stamp). The absence of a particular option may be specified with a leading *!*.

`established`

Matches if the packet is part of an already established TCP connection (i.e. it has the RST or ACK bits set). You can optimize the performance of the firewall by placing *established* rules early in the chain.

`setup`

Matches if the packet is an attempt to establish a TCP connection (the SYN bit is set but the ACK bit is not).

`tcpflags flags`

Matches if the TCP header contains the comma separated list of *flags*. The supported flags are *fin*, *syn*, *rst*, *psh*, *ack*, and *urg*. The absence of a particular flag may be indicated by a leading *!*.

`icmptypes types`

Matches if the ICMP type is present in the list *types*. The list may be specified as any combination of ranges and/or individual types separated by commas. Commonly used ICMP types are: *0* echo reply (ping reply), *3* destination unreachable, *5* redirect, *8* echo request (ping request), and *11* time exceeded (used to indicate TTL expiration as with [traceroute\(8\)](#)).

#### 10.8.4.2. Listing the IPFW Rules

The syntax for this form of the command is:

```
ipfw [-a] [-c] [-d] [-e] [-t] [-N] [-S] list
```

There are seven valid flags when using this form of the command:

`-a`

While listing, show counter values. This option is the only way to see accounting counters.

-c

List rules in compact form.

-d

Show dynamic rules in addition to static rules.

-e

If *-d* was specified, also show expired dynamic rules.

-t

Display the last match times for each chain entry. The time listing is incompatible with the input syntax used by the [ipfw\(8\)](#) utility.

-N

Attempt to resolve given addresses and service names.

-S

Show the set each rule belongs to. If this flag is not specified, disabled rules will not be listed.

### 10.8.4.3. Flushing the IPFW Rules

The syntax for flushing the chain is:

```
ipfw flush
```

This causes all entries in the firewall chain to be removed except the fixed default policy enforced by the kernel (index 65535). Use caution when flushing rules; the default deny policy will leave your system cut off from the network until allow entries are added to the chain.

### 10.8.4.4. Clearing the IPFW Packet Counters

The syntax for clearing one or more packet counters is:

```
ipfw zero [index]
```

When used without an *index* argument, all packet counters are cleared. If an *index* is supplied, the clearing operation only affects a specific chain entry.

### 10.8.5. Example Commands for ipfw

This command will deny all packets from the host `evil.crackers.org` to the telnet port of the host `nice.people.org`:

```
ipfw add deny tcp from evil.crackers.org to nice.people.org 23
```

The next example denies and logs any TCP traffic from the entire `crackers.org` network (a class C) to the `nice.people.org` machine (any port).

```
ipfw add deny log tcp from evil.crackers.org/24 to nice.people.org
```

If you do not want people sending X sessions to your internal network (a subnet of a class C), the following command will do the necessary filtering:

```
ipfw add deny tcp from any to my.org/28 6000 setup
```

To see the accounting records:

```
ipfw -a list
or in the short form
```

```
ipfw -a l
```

You can also see the last time a chain entry was matched with:

```
ipfw -at l
```

### 10.8.6. Building a Packet Filtering Firewall

**Nota:** The following suggestions are just that: suggestions. The requirements of each firewall are different and we cannot tell you how to build a firewall to meet your particular requirements.

When initially setting up your firewall, unless you have a test bench setup where you can configure your firewall host in a controlled environment, it is strongly recommend you use the logging version of the commands and enable logging in the kernel. This will allow you to quickly identify problem areas and cure them without too much disruption. Even after the initial setup phase is complete, I recommend using the logging for `'deny'` as it allows tracing of possible attacks and also modification of the firewall rules if your requirements alter.

**Nota:** If you use the logging versions of the `accept` command, be aware that it can generate *large* amounts of log data. One log entry will be generated for every packet that passes through the firewall, so large FTP/http transfers, etc, will really slow the system down. It also increases the latencies on those packets as it requires more work to be done by the kernel before the packet can be passed on. **syslogd** will also start using up a lot more processor time as it logs all the extra data to disk, and it could quite easily fill the partition `/var/log` is located on.

You should enable your firewall from `/etc/rc.conf.local` or `/etc/rc.conf`. The associated manual page explains which knobs to fiddle and lists some preset firewall configurations. If you do not use a preset configuration, `ipfw list` will output the current ruleset into a file that you can pass to `rc.conf`. If you do not use `/etc/rc.conf.local` or `/etc/rc.conf` to enable your firewall, it is important to make sure your firewall is enabled before any IP interfaces are configured.

The next problem is what your firewall should actually *do*! This is largely dependent on what access to your network you want to allow from the outside, and how much access to the outside world you want to allow from the inside. Some general rules are:

- Block all incoming access to ports below 1024 for TCP. This is where most of the security sensitive services are, like finger, SMTP (mail) and telnet.
- Block *all* incoming UDP traffic. There are very few useful services that travel over UDP, and what useful traffic there is, is normally a security threat (e.g. Sun's RPC and NFS protocols). This has its disadvantages also, since UDP is a connectionless protocol, denying incoming UDP traffic also blocks the replies to outgoing UDP traffic. This can cause a problem for people (on the inside) using external archie (prospero) servers. If you want to allow access to archie, you will have to allow packets coming from ports 191 and 1525 to any internal UDP port through the firewall. **ntp** is another service you may consider allowing through, which comes from port 123.
- Block traffic to port 6000 from the outside. Port 6000 is the port used for access to X11 servers, and can be a security threat (especially if people are in the habit of doing `xhost +` on their workstations). X11 can actually use a range of ports starting at 6000, the upper limit being how many X displays you can run on the machine. The upper limit as defined by RFC 1700 (Assigned Numbers) is 6063.
- Check what ports any internal servers use (e.g. SQL servers, etc). It is probably a good idea to block those as well, as they normally fall outside the 1-1024 range specified above.

Another checklist for firewall configuration is available from CERT at [http://www.cert.org/tech\\_tips/packet\\_filtering.html](http://www.cert.org/tech_tips/packet_filtering.html)

As stated above, these are only *guidelines*. You will have to decide what filter rules you want to use on your firewall yourself. We cannot accept ANY responsibility if someone breaks into your network, even if you follow the advice given above.

### 10.8.7. IPFW Overhead and Optimization

Many people want to know how much overhead IPFW adds to a system. The answer to this depends mostly on your rule set and processor speed. For most applications dealing with Ethernet and small rule sets, the answer is "negligible". For those of you that need actual measurements to satisfy your curiosity, read on.

The following measurements were made using 2.2.5-STABLE on a 486-66. (While IPFW has changed slightly in later releases of FreeBSD, it still performs with similar speed.) IPFW was modified to measure the time spent within the `ip_fw_chk` routine, displaying the results to the console every 1000 packets.

Two rule sets, each with 1000 rules, were tested. The first set was designed to demonstrate a worst case scenario by repeating the rule:

```
ipfw add deny tcp from any to any 55555
```

This demonstrates a worst case scenario by causing most of IPFW's packet check routine to be executed before finally deciding that the packet does not match the rule (by virtue of the port number). Following the 999th iteration of this rule was an *allow ip from any to any*.

The second set of rules were designed to abort the rule check quickly:

```
ipfw add deny ip from 1.2.3.4 to 1.2.3.4
```

The non-matching source IP address for the above rule causes these rules to be skipped very quickly. As before, the 1000th rule was an *allow ip from any to any*.

The per-packet processing overhead in the former case was approximately 2.703 ms/packet, or roughly 2.7 microseconds per rule. Thus the theoretical packet processing limit with these rules is around 370 packets per second. Assuming 10 Mbps Ethernet and a ~1500 byte packet size, we would only be able to achieve 55.5% bandwidth utilization.

For the latter case each packet was processed in approximately 1.172 ms, or roughly 1.2 microseconds per rule. The theoretical packet processing limit here would be about 853 packets per second, which could consume 10 Mbps Ethernet bandwidth.

The excessive number of rules tested and the nature of those rules do not provide a real-world scenario -- they were used only to generate the timing information presented here. Here are a few things to keep in mind when building an efficient rule set:

- Place an *established* rule early on to handle the majority of TCP traffic. Do not put any *allow tcp* statements before this rule.
- Place heavily triggered rules earlier in the rule set than those rarely used (*without changing the permissiveness of the firewall*, of course). You can see which rules are used most often by examining the packet counting statistics with `ipfw -a 1`.

---

## 10.9. OpenSSL

As of FreeBSD 4.0, the OpenSSL toolkit is a part of the base system. [OpenSSL](#) provides a general-purpose cryptography library, as well as the Secure Sockets Layer v2/v3 (SSLv2/SSLv3) and Transport Layer Security v1 (TLSv1) network security protocols.

However, one of the algorithms (specifically IDEA) included in OpenSSL is protected by patents in the USA and elsewhere, and is not available for unrestricted use. IDEA is

included in the OpenSSL sources in FreeBSD, but it is not built by default. If you wish to use it, and you comply with the license terms, enable the `MAKE_IDEA` switch in `/etc/make.conf` and rebuild your sources using `make world`.

Today, the RSA algorithm is free for use in USA and other countries. In the past it was protected by a patent.

### 10.9.1. Source Code Installations

OpenSSL is part of the *src-crypto* and *src-secure* CVSup collections. See the [Obtaining FreeBSD](#) section for more information about obtaining and updating FreeBSD source code.

---

## 10.10. VPN sobre IPsec

*Escrito por Nik Clayton.*

Criando uma VPN entre duas redes, separadas pela internet, usando *gateways* FreeBSD.

### 10.10.1. Understanding IPsec

*Written by Hiten M. Pandya.*

This section will guide you through the process of setting up IPsec, and to use it in an environment which consists of FreeBSD and **Microsoft® Windows® 2000/XP** machines, to make them communicate securely. In order to set up IPsec, it is necessary that you are familiar with the concepts of building a custom kernel (see [Capítulo 9](#)).

*IPsec* is a protocol which sits on top of the Internet Protocol (IP) layer. It allows two or more hosts to communicate in a secure manner (hence the name). The FreeBSD IPsec "network stack" is based on the [KAME](#) implementation, which has support for both protocol families, IPv4 and IPv6.

**Nota:** FreeBSD 5.X contains a "hardware accelerated" IPsec stack, known as "Fast IPsec", that was obtained from OpenBSD. It employs cryptographic hardware (whenever possible) via the [crypto\(4\)](#) subsystem to optimize the performance of IPsec. This subsystem is new, and does not support all the features that are available in the KAME version of IPsec. However, in order to enable hardware-accelerated IPsec, the following kernel option has to be added to your kernel configuration file:

```
options FAST_IPSEC # new IPsec (cannot define w/ IPSEC)
```

Note, that it is not currently possible to use the "Fast IPsec" subsystem in lue with the KAME implementation of IPsec. Consult the [fast\\_ipsec\(4\)](#) manual page for more information.

IPsec consists of two sub-protocols:

- *Encapsulated Security Payload (ESP)*, protects the IP packet data from third party interference, by encrypting the contents using symmetric cryptography algorithms (like Blowfish, 3DES).
- *Authentication Header (AH)*, protects the IP packet header from third party interference and spoofing, by computing a cryptographic checksum and hashing the IP packet header fields with a secure hashing function. This is then followed by an additional header that contains the hash, to allow the information in the packet to be authenticated.

ESP and AH can either be used together or separately, depending on the environment.

IPsec can either be used to directly encrypt the traffic between two hosts (known as *Transport Mode*); or to build "virtual tunnels" between two subnets, which could be used for secure communication between two corporate networks (known as *Tunnel Mode*). The latter is more commonly known as a *Virtual Private Network (VPN)*. The [ipsec\(4\)](#) manual page should be consulted for detailed information on the IPsec subsystem in FreeBSD.

To add IPsec support to your kernel, add the following options to your kernel configuration file:

```
options IPSEC #IP security
options IPSEC_ESP #IP security (crypto; define w/ IPSEC)
```

If IPsec debugging support is desired, the following kernel option should also be added:

```
options IPSEC_DEBUG #debug for IP security
```

### 10.10.2. The Problem

There's no standard for what constitutes a VPN. VPNs can be implemented using a number of different technologies, each of which have their own strengths and weaknesses. This article presents a number of scenarios, and strategies for implementing a VPN for each scenario.

### 10.10.3. Scenario #1: Two networks, connected to the Internet, to behave as one

This is the scenario that caused me to first investigating VPNs. The premise is as follows:

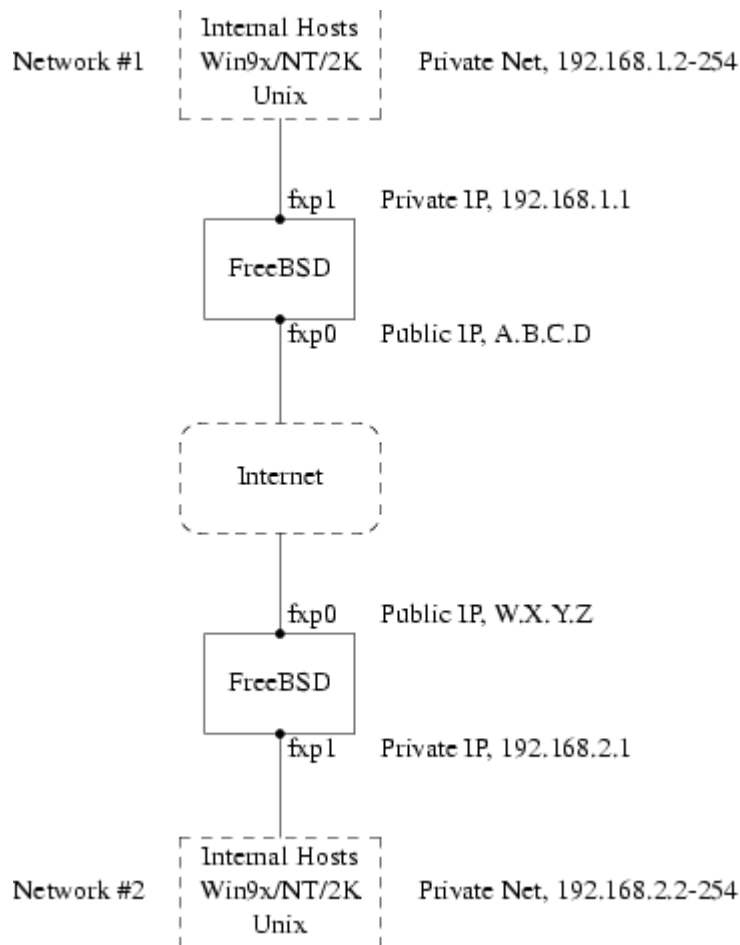
- You have at least two sites
- Both sites are using IP internally
- Both sites are connected to the Internet, through a gateway that is running FreeBSD.
- The gateway on each network has at least one public IP address.



- The internal addresses of the two networks can be public or private IP addresses, it doesn't matter. You can be running NAT on the gateway machine if necessary.
- The internal IP addresses of the two networks *do not collide*. While I expect it is theoretically possible to use a combination of VPN technology and NAT to get this to work, I expect it to be a configuration nightmare.

If you find that you are trying to connect two networks, both of which, internally, use the same private IP address range (e.g., both of them use 192.168.1.x), then one of the networks will have to be renumbered.

The network topology might look something like this:



Notice the two public IP addresses. I'll use the letters to refer to them in the rest of this article. Anywhere you see those letters in this article, replace them with your own public IP addresses. Note also that that internally, the two gateway machines have .1 IP addresses, and that the two networks have different private IP address (192.168.1.x and 192.168.2.x respectively). All the machines on the private networks have been configured to use the .1 machine as their default gateway.

The intention is that, from a network point of view, each network should view the machines on the other network as though they were directly attached the same router -- albeit a slightly slow router with an occasional tendency to drop packets.

This means that (for example), machine 192.168.1.20 should be able to run

```
ping 192.168.2.34
```

and have it work, transparently. Windows machines should be able to see the machines on the other network, browse file shares, and so on, in exactly the same way that they can browse machines on the local network.

And the whole thing has to be secure. This means that traffic between the two networks has to be encrypted.

Creating a VPN between these two networks is a multi-step process. The stages are as follows:

1. Create a ``virtual" network link between the two networks, across the Internet. Test it, using tools like [ping\(8\)](#), to make sure it works.
2. Apply security policies to ensure that traffic between the two networks is transparently encrypted and decrypted as necessary. Test this, using tools like [tcpdump\(1\)](#), to ensure that traffic is encrypted.
3. Configure additional software on the FreeBSD gateways, to allow Windows machines to see one another across the VPN.

### 10.10.3.1. Step 1: Creating and testing a ``virtual" network link

Suppose that you were logged in to the gateway machine on network #1 (with public IP address `A.B.C.D`, private IP address `192.168.1.1`), and you ran `ping 192.168.2.1`, which is the private address of the machine with IP address `w.x.y.z`. What needs to happen in order for this to work?

1. The gateway machine needs to know how to reach `192.168.2.1`. In other words, it needs to have a route to `192.168.2.1`.
2. Private IP addresses, such as those in the `192.168.x` range are not supposed to appear on the Internet at large. Instead, each packet you send to `192.168.2.1` will need to be wrapped up inside another packet. This packet will need to appear to be from `A.B.C.D`, and it will have to be sent to `w.x.y.z`. This process is called *encapsulation*.
3. Once this packet arrives at `w.x.y.z` it will need to ``unencapsulated", and delivered to `192.168.2.1`.

You can think of this as requiring a ``tunnel" between the two networks. The two ``tunnel mouths" are the IP addresses `A.B.C.D` and `w.x.y.z`, and the tunnel must be told the addresses of the private IP addresses that will be allowed to pass through it. The tunnel is used to transfer traffic with private IP addresses across the public Internet.

This tunnel is created by using the generic interface, or `gif` devices on FreeBSD. As you can imagine, the `gif` interface on each gateway host must be configured with four IP addresses; two for the public IP addresses, and two for the private IP addresses.

Support for the `gif` device must be compiled in to the FreeBSD kernel on both machines. You can do this by adding the line:

```
pseudo-device gif
```

to the kernel configuration files on both machines, and then compile, install, and reboot as normal.

Configuring the tunnel is a two step process. First the tunnel must be told what the outside (or public) IP addresses are, using [gifconfig\(8\)](#). Then the private IP addresses must be configured using [ifconfig\(8\)](#).

On the gateway machine on network #1 you would run the following two commands to configure the tunnel.

```
gifconfig gif0 A.B.C.D W.X.Y.Z
ifconfig gif0 inet 192.168.1.1 192.168.2.1 netmask 0xffffffff
```

On the other gateway machine you run the same commands, but with the order of the IP addresses reversed.

```
gifconfig gif0 W.X.Y.Z A.B.C.D
ifconfig gif0 inet 192.168.2.1 192.168.1.1 netmask 0xffffffff
```

You can then run:

```
gifconfig gif0
```

to see the configuration. For example, on the network #1 gateway, you would see this:

```
gifconfig gif0
gif0: flags=8011<UP,POINTTOPOINT,MULTICAST> mtu 1280
inet 192.168.1.1 --> 192.168.2.1 netmask 0xffffffff
physical address inet A.B.C.D --> W.X.Y.Z
```

As you can see, a tunnel has been created between the physical addresses A.B.C.D and W.X.Y.Z, and the traffic allowed through the tunnel is that between 192.168.1.1 and 192.168.2.1.

This will also have added an entry to the routing table on both machines, which you can examine with the command `netstat -rn`. This output is from the gateway host on network #1.

```
netstat -rn
Routing tables

Internet:
Destination Gateway Flags Refs Use Netif Expire
...
192.168.2.1 192.168.1.1 UH 0 0 gif0
...
```

As the ``Flags" value indicates, this is a host route, which means that each gateway knows how to reach the other gateway, but they do not know how to reach the rest of their respective networks. That problem will be fixed shortly.

It is likely that you are running a firewall on both machines. This will need to be circumvented for your VPN traffic. You might want to allow all traffic between both networks, or you might want to include firewall rules that protect both ends of the VPN from one another.

It greatly simplifies testing if you configure the firewall to allow all traffic through the VPN. You can always tighten things up later. If you are using [ipfw\(8\)](#) on the gateway machines then a command like

```
ipfw add 1 allow ip from any to any via gif0
```

will allow all traffic between the two end points of the VPN, without affecting your other firewall rules. Obviously you will need to run this command on both gateway hosts.

This is sufficient to allow each gateway machine to ping the other. On 192.168.1.1, you should be able to run

```
ping 192.168.2.1
```

and get a response, and you should be able to do the same thing on the other gateway machine.

However, you will not be able to reach internal machines on either network yet. This is because of the routing -- although the gateway machines know how to reach one another, they do not know how to reach the network behind each one.

To solve this problem you must add a static route on each gateway machine. The command to do this on the first gateway would be:

```
route add 192.168.2.0 192.168.2.1 netmask 0xfffff00
```

This says "In order to reach the hosts on the network 192.168.2.0, send the packets to the host 192.168.2.1". You will need to run a similar command on the other gateway, but with the 192.168.1.x addresses instead.

IP traffic from hosts on one network will now be able to reach hosts on the other network.

That has now created two thirds of a VPN between the two networks, in as much as it is "virtual" and it is a "network". It is not private yet. You can test this using [ping\(8\)](#) and [tcpdump\(1\)](#). Log in to the gateway host and run

```
tcpdump dst host 192.168.2.1
```

In another log in session on the same host run

```
ping 192.168.2.1
```

You will see output that looks something like this:

```
16:10:24.018080 192.168.1.1 > 192.168.2.1: icmp: echo request
16:10:24.018109 192.168.1.1 > 192.168.2.1: icmp: echo reply
16:10:25.018814 192.168.1.1 > 192.168.2.1: icmp: echo request
16:10:25.018847 192.168.1.1 > 192.168.2.1: icmp: echo reply
16:10:26.028896 192.168.1.1 > 192.168.2.1: icmp: echo request
16:10:26.029112 192.168.1.1 > 192.168.2.1: icmp: echo reply
```

As you can see, the ICMP messages are going back and forth unencrypted. If you had used the `-s` parameter to [tcpdump\(1\)](#) to grab more bytes of data from the packets you would see more information.

Obviously this is unacceptable. The next section will discuss securing the link between the two networks so that it all traffic is automatically encrypted.

### Summary:

- Configure both kernels with ``pseudo-device gif''.
- Edit `/etc/rc.conf` on gateway host #1 and add the following lines (replacing IP addresses as necessary).
  - `gifconfig_gif0="A.B.C.D W.X.Y.Z"`
  - `ifconfig_gif0="inet 192.168.1.1 192.168.2.1 netmask 0xffffffff"`
  - `static_routes="vpn"`
  - `route_vpn="192.168.2.0 192.168.2.1 netmask 0xffffffff00"`
  -
- Edit your firewall script (`/etc/rc.firewall`, or similar) on both hosts, and add
  - `ipfw add 1 allow ip from any to any via gif0`
- Make similar changes to `/etc/rc.conf` on gateway host #2, reversing the order of IP addresses.

### 10.10.3.2. Step 2: Securing the link

To secure the link we will be using IPsec. IPsec provides a mechanism for two hosts to agree on an encryption key, and to then use this key in order to encrypt data between the two hosts.

There are two areas of configuration to be considered here.

1. There must be a mechanism for two hosts to agree on the encryption mechanism to use. Once two hosts have agreed on this mechanism there is said to be a ``security association" between them.
2. There must be a mechanism for specifying which traffic should be encrypted. Obviously, you don't want to encrypt all your outgoing traffic -- you only want to encrypt the traffic that is part of the VPN. The rules that you put in place to determine what traffic will be encrypted are called ``security policies".

Security associations and security policies are both maintained by the kernel, and can be modified by userland programs. However, before you can do this you must configure

the kernel to support IPsec and the Encapsulated Security Payload (ESP) protocol. This is done by configuring a kernel with:

```
options IPSEC
options IPSEC_ESP
```

and recompiling, reinstalling, and rebooting. As before you will need to do this to the kernels on both of the gateway hosts.

You have two choices when it comes to setting up security associations. You can configure them by hand between two hosts, which entails choosing the encryption algorithm, encryption keys, and so forth, or you can use daemons that implement the Internet Key Exchange protocol (IKE) to do this for you.

I recommend the latter. Apart from anything else, it is easier to set up.

Editing and displaying security policies is carried out using [setkey\(8\)](#). By analogy, `setkey` is to the kernel's security policy tables as [route\(8\)](#) is to the kernel's routing tables. `setkey` can also display the current security associations, and to continue the analogy further, is akin to `netstat -r` in that respect.

There are a number of choices for daemons to manage security associations with FreeBSD. This article will describe how to use one of these, `racoon`. `racoon` is in the FreeBSD ports collection, in the `security/` category, and is installed in the usual way.

`racoon` must be run on both gateway hosts. On each host it is configured with the IP address of the other end of the VPN, and a secret key (which you choose, and must be the same on both gateways).

The two daemons then contact one another, confirm that they are who they say they are (by using the secret key that you configured). The daemons then generate a new secret key, and use this to encrypt the traffic over the VPN. They periodically change this secret, so that even if an attacker were to crack one of the keys (which is as theoretically close to unfeasible as it gets) it won't do them much good -- by the time they've cracked the key the two daemons have chosen another one.

`racoon`'s configuration is stored in `/${PREFIX}/etc/racoon`. You should find a configuration file there, which should not need to be changed too much. The other component of `racoon`'s configuration, which you will need to change, is the "pre-shared key".

The default `racoon` configuration expects to find this in the file `/${PREFIX}/etc/racoon/psk.txt`. It is important to note that the pre-shared key is *not* the key that will be used to encrypt your traffic across the VPN link, it is simply a token that allows the key management daemons to trust one another.

`psk.txt` contains a line for each remote site you are dealing with. In this example, where there are two sites, each `psk.txt` file will contain one line (because each end of the VPN is only dealing with one other end).

On gateway host #1 this line should look like this:

```
W.X.Y.Z secret
```

That is, the *public* IP address of the remote end, whitespace, and a text string that provides the secret. Obviously, you shouldn't use ``secret" as your key -- the normal rules for choosing a password apply.

On gateway host #2 the line would look like this

```
A.B.C.D secret
```

That is, the public IP address of the remote end, and the same secret key. `psk.txt` must be mode `0600` (i.e., only read/write to `root`) before `racoon` will run.

You must run `racoon` on both gateway machines. You will also need to add some firewall rules to allow the IKE traffic, which is carried over UDP to the ISAKMP (Internet Security Association Key Management Protocol) port. Again, this should be fairly early in your firewall ruleset.

```
ipfw add 1 allow udp from A.B.C.D to W.X.Y.Z isakmp
ipfw add 1 allow udp from W.X.Y.Z to A.B.C.D isakmp
```

Once `racoon` is running you can try pinging one gateway host from the other. The connection is still not encrypted, but `racoon` will then set up the security associations between the two hosts -- this might take a moment, and you may see this as a short delay before the ping commands start responding.

Once the security association has been set up you can view it using [setkey\(8\)](#). Run

```
setkey -D
```

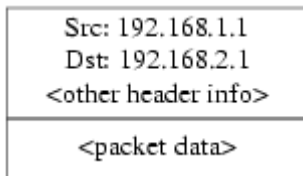
on either host to view the security association information.

That's one half of the problem. The other half is setting your security policies.

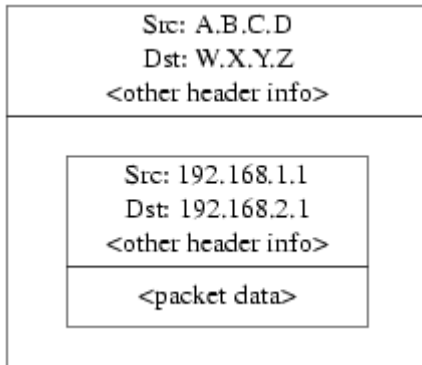
To create a sensible security policy, let's review what's been set up so far. This discussion holds for both ends of the link.

Each IP packet that you send out has a header that contains data about the packet. The header includes the IP addresses of both the source and destination. As we already know, private IP addresses, such as the `192.168.x.y` range are not supposed to appear on the public Internet. Instead, they must first be encapsulated inside another packet. This packet must have the public source and destination IP addresses substituted for the private addresses.

So if your outgoing packet started looking like this:



Then it will be encapsulated inside another packet, looking something like this:



This encapsulation is carried out by the `gif` device. As you can see, the packet now has real IP addresses on the outside, and our original packet has been wrapped up as data inside the packet that will be put out on the Internet.

Obviously, we want all traffic between the VPNs to be encrypted. You might try putting this in to words, as:

``If a packet leaves from `A.B.C.D`, and it is destined for `w.x.y.z`, then encrypt it, using the necessary security associations."

``If a packet arrives from `w.x.y.z`, and it is destined for `A.B.C.D`, then decrypt it, using the necessary security associations."

That's close, but not quite right. If you did this, all traffic to and from `w.x.y.z`, even traffic that was not part of the VPN, would be encrypted. That's not quite what you want. The correct policy is as follows

``If a packet leaves from `A.B.C.D`, and that packet is encapsulating another packet, and it is destined for `w.x.y.z`, then encrypt it, using the necessary security associations."

``If a packet arrives from `w.x.y.z`, and that packet is encapsulating another packet, and it is destined for `A.B.C.D`, then decrypt it, using the necessary security associations."

A subtle change, but a necessary one.

Security policies are also set using [setkey\(8\)](#). [setkey\(8\)](#) features a configuration language for defining the policy. You can either enter configuration instructions via `stdin`, or you can use the `-f` option to specify a filename that contains configuration instructions.



The configuration on gateway host #1 (which has the public IP address `A.B.C.D`) to force all outbound traffic to `W.X.Y.Z` to be encrypted is:

```
spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P out ipsec esp/tunnel/A.B.C.D-
W.X.Y.Z/require;
```

Put these commands in a file (e.g., `/etc/ipsec.conf`) and then run

```
setkey -f /etc/ipsec.conf
```

`spdadd` tells [setkey\(8\)](#) that we want to add a rule to the secure policy database. The rest of this line specifies which packets will match this policy. `A.B.C.D/32` and `W.X.Y.Z/32` are the IP addresses and netmasks that identify the network or hosts that this policy will apply to. In this case, we want it to apply to traffic between these two hosts. `ipencap` tells the kernel that this policy should only apply to packets that encapsulate other packets. `-P out` says that this policy applies to outgoing packets, and `ipsec` says that the packet will be secured.

The second line specifies how this packet will be encrypted. `esp` is the protocol that will be used, while `tunnel` indicates that the packet will be further encapsulated in an IPsec packet. The repeated use of `A.B.C.D` and `W.X.Y.Z` is used to select the security association to use, and the final `require` mandates that packets must be encrypted if they match this rule.

This rule only matches outgoing packets. You will need a similar rule to match incoming packets.

```
spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P in ipsec esp/tunnel/W.X.Y.Z-
A.B.C.D/require;
```

Note the *in* instead of *out* in this case, and the necessary reversal of the IP addresses.

The other gateway host (which has the public IP address `W.X.Y.Z`) will need similar rules.

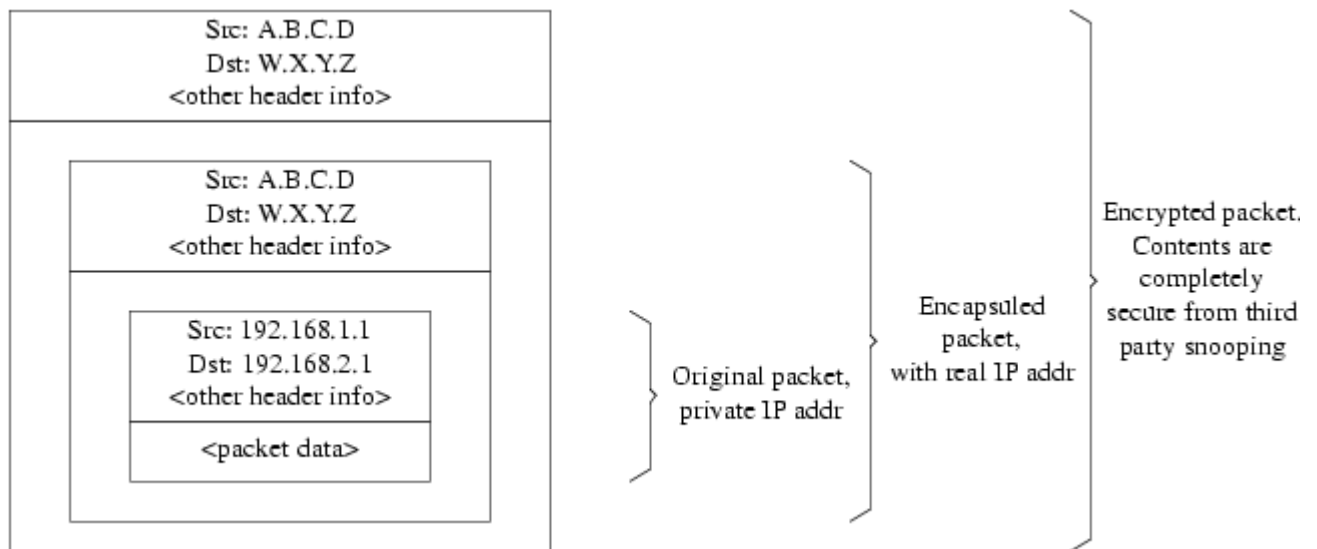
```
spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P out ipsec esp/tunnel/W.X.Y.Z-
A.B.C.D/require;
 spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P in ipsec
esp/tunnel/A.B.C.D-W.X.Y.Z/require;
```

Finally, you need to add firewall rules to allow ESP and IPENCAP packets back and forth. These rules will need to be added to both hosts.

```
ipfw add 1 allow esp from A.B.C.D to W.X.Y.Z
ipfw add 1 allow esp from W.X.Y.Z to A.B.C.D
ipfw add 1 allow ipencap from A.B.C.D to W.X.Y.Z
ipfw add 1 allow ipencap from W.X.Y.Z to A.B.C.D
```

Because the rules are symmetric you can use the same rules on each gateway host.

Outgoing packets will now look something like this:



When they are received by the far end of the VPN they will first be decrypted (using the security associations that have been negotiated by racoon). Then they will enter the `gif` interface, which will unwrap the second layer, until you are left with the innermost packet, which can then travel in to the inner network.

You can check the security using the same [ping\(8\)](#) test from earlier. First, log in to the A.B.C.D gateway machine, and run:

```
tcpdump dst host 192.168.2.1
```

In another log in session on the same host run

```
ping 192.168.2.1
```

This time you should see output like the following:

```
XXX tcpdump output
```

Now, as you can see, [tcpdump\(1\)](#) shows the ESP packets. If you try and examine them with the `-s` option you will see (apparently) gibberish, because of the encryption.

Congratulations. You have just set up a VPN between two remote sites.

## Summary

- Configure both kernels with:
  - `options IPSEC`
  - `options IPSEC_ESP`
  -
- Install [security/racoon](#). Edit `/${PREFIX}/etc/racoon/psk.txt` on both gateway hosts, adding an entry for the remote host's IP address and a secret key that they both know. Make sure this file is mode 0600.
- Add the following lines to `/etc/rc.conf` on each host:
  - `ipsec_enable="YES"`
  - `ipsec_file="/etc/ipsec.conf"`

- 
- Create an `/etc/ipsec.conf` on each host that contains the necessary `spdadd` lines. On gateway host #1 this would be:
  - `spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P out ipsec`
  - `esp/tunnel/A.B.C.D-W.X.Y.Z/require;`
  - `spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P in ipsec`
  - `esp/tunnel/W.X.Y.Z-A.B.C.D/require;`

On gateway host #2 this would be:

```
spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P out ipsec
 esp/tunnel/W.X.Y.Z-A.B.C.D/require;
spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P in ipsec
 esp/tunnel/A.B.C.D-W.X.Y.Z/require;
```

- Add firewall rules to allow IKE, ESP, and IPENCAP traffic to both hosts:
  - `ipfw add 1 allow udp from A.B.C.D to W.X.Y.Z isakmp`
  - `ipfw add 1 allow udp from W.X.Y.Z to A.B.C.D isakmp`
  - `ipfw add 1 allow esp from A.B.C.D to W.X.Y.Z`
  - `ipfw add 1 allow esp from W.X.Y.Z to A.B.C.D`
  - `ipfw add 1 allow ipencap from A.B.C.D to W.X.Y.Z`
  - `ipfw add 1 allow ipencap from W.X.Y.Z to A.B.C.D`
- 

The previous two steps should suffice to get the VPN up and running. Machines on each network will be able to refer to one another using IP addresses, and all traffic across the link will be automatically and securely encrypted.

---

## 10.11. OpenSSH

*Contribuição de Chern Lee.*

**OpenSSH** is a set of network connectivity tools used to access remote machines securely. It can be used as a direct replacement for `rlogin`, `rsh`, `rcp`, and `telnet`. Additionally, any other TCP/IP connections can be tunneled/forwarded securely through SSH. **OpenSSH** encrypts all traffic to effectively eliminate eavesdropping, connection hijacking, and other network-level attacks.

**OpenSSH** is maintained by the OpenBSD project, and is based upon SSH v1.2.12 with all the recent bug fixes and updates. It is compatible with both SSH protocols 1 and 2. **OpenSSH** has been in the base system since FreeBSD 4.0.

### 10.11.1. Advantages of Using OpenSSH

Normally, when using [telnet\(1\)](#) or [rlogin\(1\)](#), data is sent over the network in a clear, un-encrypted form. Network sniffers anywhere in between the client and server can steal your user/password information or data transferred in your session. **OpenSSH**

offers a variety of authentication and encryption methods to prevent this from happening.

### 10.11.2. Enabling sshd

Be sure to make the following addition to your `rc.conf` file:

```
sshd_enable="YES"
```

This will load [sshd\(8\)](#), the daemon program for **OpenSSH**, the next time your system initializes. Alternatively, you can simply run directly the `sshd` daemon by typing `sshd` on the command line.

### 10.11.3. SSH Client

The [ssh\(1\)](#) utility works similarly to [rlogin\(1\)](#).

```
ssh user@example.com
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)? yes
Host 'example.com' added to the list of known hosts.
user@example.com's password: *****
```

The login will continue just as it would have if a session was created using `rlogin` or `telnet`. SSH utilizes a key fingerprint system for verifying the authenticity of the server when the client connects. The user is prompted to enter *yes* only when connecting for the first time. Future attempts to login are all verified against the saved fingerprint key. The SSH client will alert you if the saved fingerprint differs from the received fingerprint on future login attempts. The fingerprints are saved in `~/.ssh/known_hosts`, or `~/.ssh/known_hosts2` for SSH v2 fingerprints.

By default, **OpenSSH** servers are configured to accept both SSH v1 and SSH v2 connections. The client, however, can choose between the two. Version 2 is known to be more robust and secure than its predecessor.

The [ssh\(1\)](#) command can be forced to use either protocol by passing it the `-1` or `-2` argument for v1 and v2, respectively.

### 10.11.4. Secure Copy

The [scp\(1\)](#) command works similarly to [rcp\(1\)](#); it copies a file to or from a remote machine, except in a secure fashion.

```
scp user@example.com:/COPYRIGHT COPYRIGHT
user@example.com's password: *****
COPYRIGHT 100% |*****| 4735
00:00
#
```

Since the fingerprint was already saved for this host in the previous example, it is verified when using [scp\(1\)](#) here.

The arguments passed to [scp\(1\)](#) are similar to [cp\(1\)](#), with the file or files in the first argument, and the destination in the second. Since the file is fetched over the network, through SSH, one or more of the file arguments takes on the form `user@host:<path_to_remote_file>`.

### 10.11.5. Configuration

The system-wide configuration files for both the **OpenSSH** daemon and client reside within the `/etc/ssh` directory.

`ssh_config` configures the client settings, while `sshd_config` configures the daemon.

Additionally, the `sshd_program` (`/usr/sbin/sshd` by default), and `sshd_flags rc.conf` options can provide more levels of configuration.

### 10.11.6. ssh-keygen

Instead of using passwords, [ssh-keygen\(1\)](#) can be used to generate RSA keys to authenticate a user:

```
% ssh-keygen -t rsa1
Initializing random number generator...
Generating p: .++ (distance 66)
Generating q: ++ (distance 498)
Computing the keys...
Key generation complete.
Enter file in which to save the key (/home/user/.ssh/identity):
Enter passphrase:
Enter the same passphrase again:
Your identification has been saved in /home/user/.ssh/identity.
...
```

[ssh-keygen\(1\)](#) will create a public and private key pair for use in authentication. The private key is stored in `~/.ssh/identity`, whereas the public key is stored in `~/.ssh/identity.pub`. The public key must be placed in `~/.ssh/authorized_keys` of the remote machine in order for the setup to work.

This will allow connection to the remote machine based upon RSA authentication instead of passwords.

**Nota:** The `-t rsa1` option will create RSA keys for use by SSH protocol version 1. If you want to use RSA keys with the SSH protocol version 2, you have to use the command `ssh-keygen -t rsa`.

If a passphrase is used in [ssh-keygen\(1\)](#), the user will be prompted for a password each time in order to use the private key.

A SSH protocol version 2 DSA key can be created for the same purpose by using the `ssh-keygen -t dsa` command. This will create a public/private DSA key for use in SSH protocol version 2 sessions only. The public key is stored in `~/.ssh/id_dsa.pub`, while the private key is in `~/.ssh/id_dsa`.

DSA public keys are also placed in `~/.ssh/authorized_keys` on the remote machine.

[ssh-agent\(1\)](#) and [ssh-add\(1\)](#) are utilities used in managing multiple passworded private keys.

**Atenção** The various options and files can be different according to the **OpenSSH** version you have on your system, to avoid problems you should consult the [ssh-keygen\(1\)](#) manual page.

### 10.11.7. SSH Tunneling

**OpenSSH** has the ability to create a tunnel to encapsulate another protocol in an encrypted session.

The following command tells [ssh\(1\)](#) to create a tunnel for **telnet**:

```
% ssh -2 -N -f -L 5023:localhost:23 user@foo.example.com
%
```

The `ssh` command is used with the following options:

`-2`

Forces `ssh` to use version 2 of the protocol. (Do not use if you are working with older SSH servers)

`-N`

Indicates no command, or tunnel only. If omitted, `ssh` would initiate a normal session.

`-f`

Forces `ssh` to run in the background.

`-L`

Indicates a local tunnel in *localport:remotehost:remoteport* fashion.

*user@foo.example.com*

The remote SSH server.

An SSH tunnel works by creating a listen socket on `localhost` on the specified port. It then forwards any connection received on the local host/port via the SSH connection to the specified remote host and port.

In the example, port 5023 on localhost is being forwarded to port 23 on localhost of the remote machine. Since 23 is **telnet**, this would create a secure **telnet** session through an SSH tunnel.

This can be used to wrap any number of insecure TCP protocols such as SMTP, POP3, FTP, etc.

### **Exemplo 10-1. Using SSH to Create a Secure Tunnel for SMTP**

```
% ssh -2 -N -f -L 5025:localhost:25 user@mailserver.example.com
user@mailserver.example.com's password: *****
% telnet localhost 5025
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mailserver.example.com ESMTTP
```

This can be used in conjunction with an [ssh-keygen\(1\)](#) and additional user accounts to create a more seamless/hassle-free SSH tunneling environment. Keys can be used in place of typing a password, and the tunnels can be run as a separate user.

## **10.11.7.1. Practical SSH Tunneling Examples**

### ***10.11.7.1.1. Secure Access of a POP3 Server***

At work, there is an SSH server that accepts connections from the outside. On the same office network resides a mail server running a POP3 server. The network, or network path between your home and office may or may not be completely trustable. Because of this, you need to check your e-mail in a secure manner. The solution is to create an SSH connection to your office's SSH server, and tunnel through to the mail server.

```
% ssh -2 -N -f -L 2110:mail.example.com:110 user@ssh-
server.example.com
user@ssh-server.example.com's password: *****
```

When the tunnel is up and running, you can point your mail client to send POP3 requests to localhost port 2110. A connection here will be forwarded securely across the tunnel to mail.example.com.

### ***10.11.7.1.2. Bypassing a Draconian Firewall***

Some network administrators impose extremely draconian firewall rules, filtering not only incoming connections, but outgoing connections. You may be only given access to contact remote machines on ports 22 and 80 for SSH and web surfing.

You may wish to access another (perhaps non-work related) service, such as an Ogg Vorbis server to stream music. If this Ogg Vorbis server is streaming on some other port than 22 or 80, you will not be able to access it.

The solution is to create an SSH connection to a machine outside of your network's firewall, and use it to tunnel to the Ogg Vorbis server.

```
% ssh -2 -N -f -L 8888:music.example.com:8000 user@unfirewalled-
system.example.org
user@unfirewalled-system.example.org's password: *****
```

Your streaming client can now be pointed to `localhost` port 8888, which will be forwarded over to `music.example.com` port 8000, successfully evading the firewall.

### 10.11.8. Further Reading

[OpenSSH](#)

[ssh\(1\)](#) [scp\(1\)](#) [ssh-keygen\(1\)](#) [ssh-agent\(1\)](#) [ssh-add\(1\)](#)

[sshd\(8\)](#) [sftp-server\(8\)](#)

---

## 10.12. Controle de Acesso Obrigatório (MAC-Mandatory Access Control)

*Patrocinado por DARPA e Network Associates Laboratories. Contribuição de Robert Watson.*

FreeBSD 5.0 includes a new kernel security framework, the TrustedBSD MAC Framework. The MAC Framework permits compile-time, boot-time, and run-time extension of the kernel access control policy, and can be used to load support for Mandatory Access Control (MAC), and custom security modules such as hardening modules. The MAC Framework is currently considered to be an experimental feature, and should not yet be used in production environments without careful consideration. It is anticipated that the MAC Framework will be appropriate for more widespread production use by FreeBSD 5.2.

When configured into a kernel, the MAC Framework permits security modules to augment the existing kernel access control model, restricting access to system services and objects. For example, the [mac\\_bsdextended\(4\)](#) module augments file system access control, permitting administrators to provide a firewall-like ruleset constraining access to file system objects based on user ids and group membership. Some modules require little or no configuration, such as [mac\\_seeotheruids\(4\)](#), whereas others perform ubiquitous object labeling, such as [mac\\_biba\(4\)](#) and [mac\\_mls\(4\)](#), and require extensive configuration.

To enable the MAC Framework in your system kernel, you must add the following entry to your kernel configuration:

```
options MAC
```

Security policy modules shipped with the base system may be loaded using [kldload\(8\)](#) or in the boot [loader\(8\)](#). They may also be compiled directly into the kernel using the following options, if the use of modules is not desired.



Different MAC policies may be configured in different ways; frequently, MAC policy modules export configuration parameters using the [sysctl\(8\)](#) MIB using the *security.mac* namespace. Policies relying on file system or other labels may require a configuration step that involves assigning initial labels to system objects or creating a policy configuration file. For information on how to configure and use each policy module, see its man page.

A variety of tools are available to configure the MAC Framework and labels maintained by various policies. Extensions have been made to the login and credential management mechanisms ([setusercontext\(3\)](#)) to support initial user labeling using [login.conf\(5\)](#). In addition, modifications have been made to [su\(1\)](#), [ps\(1\)](#), [ls\(1\)](#), and [ifconfig\(8\)](#) to inspect and set labels on processes, files, and interfaces. In addition, several new tools have been added to manage labels on objects, including [getfmac\(8\)](#), [setfmac\(8\)](#), and [setfsmac\(8\)](#) to manage labels on files, and [getpmac\(8\)](#) and [setpmac\(8\)](#).

What follows is a list of policy modules shipped with FreeBSD 5.0.

### 10.12.1. Biba Integrity Policy ([mac\\_biba](#))

Vendor: TrustedBSD Project

Module name: `mac_biba.ko`

Kernel option: `MAC_BIBA`

The Biba Integrity Policy ([mac\\_biba\(4\)](#)) provides for hierarchical and non-hierarchical labeling of all system objects with integrity data, and the strict enforcement of an information flow policy to prevent corruption of high integrity subjects and data by low-integrity subjects. Integrity is enforced by preventing high integrity subjects (generally processes) from reading low integrity objects (often files), and preventing low integrity subjects from writing to high integrity objects. This security policy is frequently used in commercial trusted systems to provide strong protection for the Trusted Code Base (TCB). Because it provides ubiquitous labeling, the Biba integrity policy must be compiled into the kernel or loaded at boot.

### 10.12.2. File System Firewall Policy ([mac\\_bsdextended](#))

Vendor: TrustedBSD Project

Module name: `mac_bsdextended.ko`

Kernel option: `MAC_BSDEXTENDED`

The File System Firewall Policy ([mac\\_bsdextended\(4\)](#)) provides an extension to the BSD file system permission model, permitting the administrator to define a set of firewall-like rules for limiting access to file system objects owned by other users and groups. Managed using [ugidfw\(8\)](#), rules may limit access to files and directories based on the uid and gids of the process attempting the access, and the owner and group of the target of the access attempt. All rules are restrictive, so they may be placed in any order. This policy requires no prior configuration or labeling, and may be appropriate in multi-

user environments where mandatory limits on inter-user data exchange are required. Caution should be exercised in limiting access to files owned by the super-user or other system user ids, as many useful programs and directories are owned by these users. As with a network firewall, improper application of file system firewall rules may render the system unusable. New tools to manage the rule set may be easily written using the [libugidfw\(3\)](#) library.

### 10.12.3. Interface Silencing Policy ([mac\\_woff](#))

Vendor: TrustedBSD Project

Module name: `mac_woff.ko`

Kernel option: `MAC_WOFF`

The interface silencing policy ([mac\\_woff\(4\)](#)) prohibits the use of network interfaces during the boot until explicitly enabled, preventing spurious stack output stack response to incoming packets. This is appropriate for use in environments where the monitoring of packets is required, but no traffic may be generated.

### 10.12.4. Low-Watermark Mandatory Access Control (LOMAC) ([mac\\_lomac](#))

Vendor: Network Associates Laboratories

Module name: `mac_lomac.ko`

Kernel option: `MAC_LOMAC`

Similar to the Biba Integrity Policy, the LOMAC policy ([mac\\_lomac\(4\)](#)) relies on the ubiquitous labeling of all system objects with integrity labels. Unlike Biba, LOMAC permits high integrity subjects to read from low integrity objects, but then downgrades the label on the subject to prevent future writes to high integrity objects. This policy may provide for greater compatibility, as well as require less initial configuration than Biba. However, as with Biba, it ubiquitously labels objects and must therefore be compiled into the kernel or loaded at boot.

### 10.12.5. Multi-Level Security Policy (MLS) ([mac\\_mls](#))

Vendor: TrustedBSD Project

Module name: `mac_mls.ko`

Kernel option: `MAC_MLS`

Multi-Level Security (MLS) ([mac\\_mls\(4\)](#)) provides for hierarchical and non-hierarchical labeling of all system objects with sensitivity data, and the strict enforcement of an information flow policy to prevent the leakage of confidential data to untrusted parties. The logical conjugate of the Biba Integrity Policy, MLS is frequently

shipped in commercial trusted operating systems to protect data secrecy in multi-user environments. Hierarchical labels provide support for the notion of clearances and classifications in traditional parlance; non-hierarchical labels provide support for "need-to-know." As with Biba, ubiquitous labeling of objects occurs, and it must therefore be compiled into the kernel or loaded at boot. As with Biba, extensive initial configuration may be required.

#### 10.12.6. MAC Stub Policy ([mac\\_none](#))

Vendor: TrustedBSD Project

Module name: `mac_none.ko`

Kernel option: `MAC_NONE`

The None policy ([mac\\_none\(4\)](#)) provides a stub sample policy for developers, implementing all entry points, but not changing the system access control policy. Running this on a production system would not be highly beneficial.

#### 10.12.7. Process Partition Policy ([mac\\_partition](#))

Vendor: TrustedBSD Project

Module name: `mac_partition.ko`

Kernel option: `MAC_PARTITION`

The Partition policy ([mac\\_partition\(4\)](#)) provides for a simple process visibility limitation, assigning labels to processes identifying what numeric system partition they are present in. If none, all other processes are visible using standard monitoring tools; if a partition identifier is present, then only other processes in the same partition are visible. This policy may be compiled into the kernel, loaded at boot, or loaded at run-time.

#### 10.12.8. See Other Uids Policy ([mac\\_seeotheruids](#))

Vendor: TrustedBSD Project

Module name: `mac_seeotheruids.ko`

Kernel option: `MAC_SEEOTHERUIDS`

The See Other Uids policy ([mac\\_seeotheruids\(4\)](#)) implements a similar process visibility model to `mac_partition`, except that it relies on process credentials to control visibility of processes, rather than partition labels. This policy may be configured to exempt certain users and groups, including permitting system operators to view all processes without special privilege. This policy may be compiled into the kernel, loaded at boot, or loaded at run-time.

### 10.12.9. MAC Framework Test Policy ([mac\\_test](#))

Vendor: TrustedBSD Project

Module name: `mac_test.ko`

Kernel option: `MAC_TEST`

The Test policy ([mac\\_test\(4\)](#)) provides a regression test environment for the MAC Framework, and will cause a fail-stop in the event that internal MAC Framework assertions about proper data labeling fail. This module can be used to detect failures to properly label system objects in the kernel implementation. This policy may be compiled into the kernel, loaded at boot, or loaded at run-time.

---

## 10.13. File System Access Control Lists

*Contribuição de Tom Rhodes.*

In conjunction with file system enhancements like snapshots, FreeBSD 5.0 and later offers the security of File System Access Control Lists (ACLs).

Access Control Lists extend the standard UNIX® permission model in a highly compatible (POSIX®.1e) way. This feature permits an administrator to make use of and take advantage of a more sophisticated security model.

To enable ACL support for UFS file systems, the following:

```
options UFS_ACL
```

must be compiled into the kernel. If this option has not been compiled in, a warning message will be displayed when attempting to mount a file system supporting ACLs. This option is included in the `GENERIC` kernel. ACLs rely on extended attributes being enabled on the file system. Extended attributes are natively supported in the next generation UNIX file system, UFS2.

**Nota:** A higher level of administrative overhead is required to configure extended attributes on UFS1 than on UFS2. The performance of extended attributes on UFS2 is also substantially higher. As a result, UFS2 is generally recommended in preference to UFS1 for use with access control lists.

ACLs are enabled by the mount-time administrative flag, `acls`, which may be added to `/etc/fstab`. The mount-time flag can also be automatically set in a persistent manner using [tunefs\(8\)](#) to modify a superblock ACLs flag in the file system header. In general, it is preferred to use the superblock flag for several reasons:

- The mount-time ACLs flag cannot be changed by a remount ([mount\(8\)](#) `-u`), only by means of a complete [umount\(8\)](#) and fresh [mount\(8\)](#). This means that ACLs

cannot be enabled on the root file system after boot. It also means that you cannot change the disposition of a file system once it is in use.

- Setting the superblock flag will cause the file system to always be mounted with ACLs enabled even if there is not an `fstab` entry or if the devices re-order. This prevents accidental mounting of the file system without ACLs enabled, which can result in ACLs being improperly enforced, and hence security problems.

**Nota:** We may change the ACLs behavior to allow the flag to be enabled without a complete fresh [mount\(8\)](#), but we consider it desirable to discourage accidental mounting without ACLs enabled, because you can shoot your feet quite nastily if you enable ACLs, then disable them, then re-enable them without flushing the extended attributes. In general, once you have enabled ACLs on a file system, they should not be disabled, as the resulting file protections may not be compatible with those intended by the users of the system, and re-enabling ACLs may re-attach the previous ACLs to files that have since had their permissions changed, resulting in other unpredictable behavior.

File systems with ACLs enabled will show a + (plus) sign in their permission settings when viewed. For example:

```
drwx----- 2 robert robert 512 Dec 27 11:54 private
drwxrwx---+ 2 robert robert 512 Dec 23 10:57 directory1
drwxrwx---+ 2 robert robert 512 Dec 22 10:20 directory2
drwxrwx---+ 2 robert robert 512 Dec 27 11:57 directory3
drwxr-xr-x 2 robert robert 512 Nov 10 11:54 public_html
```

Here we see that the `directory1`, `directory2`, and `directory3` directories are all taking advantage of ACLs. The `public_html` directory is not.

### 10.13.1. Making Use of ACLs

The file system ACLs can be viewed by the [getfacl\(1\)](#) utility. For instance, to view the ACL settings on the `test` file, one would use the command:

```
%getfacl test
#file:test
#owner:1001
#group:1001
user::rw-
group::r--
other::r--
```

To change the ACL settings on this file, invoke the [setfacl\(1\)](#) utility. Observe:

```
%setfacl -k test
```

The `-k` flag will remove all of the currently defined ACLs from a file or file system. The more preferable method would be to use `-b` as it leaves the basic fields required for ACLs to work.

```
%-m u:trhodes:rw,group:web:r--,o:--- test
```

In the aforementioned command, the *-m* option was used to modify the default ACL entries. Since there were no pre-defined entries, as they were removed by the previous command, this will restore the default options and assign the options listed. Take care to notice that if you add a user or group which does not exist on the system, an "Invalid argument" error will be printed to `stdout`.

---

## 10.14. Recomendações de Segurança do FreeBSD

*Contribuição de Tom Rhodes.*

Como muitos sistemas operacionais com qualidade de produção, FreeBSD publica "Avisos de Segurança". Estes avisos são usualmente enviados por email para listas de segurança e listados na Errata somente depois que as *releases* estão devidamente corrigidas. Esta seção irá explicar o que é um aviso, como entendê-lo, e quais os meios para aplicar um *patch* em um sistema.

### 10.14.1. Com o que um aviso se parece?

Os avisos de segurança do FreeBSD são similares a este abaixo, retirado da lista de email sobre segurança.

```
=====
=====
FreeBSD-SA-XX:XX.UTIL Security
Advisory The FreeBSD
Project

Topic: denial of service due to some problem1

Category: core2
Module: sys3
Announced: 2003-09-234
Credits: Person@EMAIL-ADDRESS5
Affects: All releases of FreeBSD6
 FreeBSD 4-STABLE prior to the correction date
Corrected: 2003-09-23 16:42:59 UTC (RELENG_4, 4.9-PRERELEASE)
 2003-09-23 20:08:42 UTC (RELENG_5_1, 5.1-RELEASE-p6)
 2003-09-23 20:07:06 UTC (RELENG_5_0, 5.0-RELEASE-p15)
 2003-09-23 16:44:58 UTC (RELENG_4_8, 4.8-RELEASE-p8)
 2003-09-23 16:47:34 UTC (RELENG_4_7, 4.7-RELEASE-p18)
 2003-09-23 16:49:46 UTC (RELENG_4_6, 4.6-RELEASE-p21)
 2003-09-23 16:51:24 UTC (RELENG_4_5, 4.5-RELEASE-p33)
 2003-09-23 16:52:45 UTC (RELENG_4_4, 4.4-RELEASE-p43)
 2003-09-23 16:54:39 UTC (RELENG_4_3, 4.3-RELEASE-p39)7

FreeBSD only: NO8
```

For general information regarding FreeBSD Security Advisories, including descriptions of the fields above, security branches, and the

following sections, please visit  
<http://www.freebsd.org/security/>.

- I. Background **9**
- II. Problem Description **10**
- III. Impact **(11)**
- IV. Workaround **(12)**
- V. Solution **(13)**
- VI. Correction details **(14)**
- VII. References **(15)**

**1**

O campo *Topic* indica exatamente o que é o problema. Ele é basicamente uma introdução ao aviso de segurança atual e cita o utilitário com a vulnerabilidade.

**2**

O *Category* referencia a parte afetada do sistema que pode ser *core*, *contrib*, ou *ports*. A categoria *core* significa que a vulnerabilidade afeta um componente do núcleo do sistema operacional FreeBSD. A categoria *contrib* significa que a vulnerabilidade afeta aplicativos contribuídos ao Projeto FreeBSD, como **Sendmail**. Finalmente a categoria *ports* indica que a vulnerabilidade afeta aplicativos adicionais disponíveis como parte da coleção de ports.

**3**

O campo *Module* referencia a localização do componente, por exemplo ```sys"`. Neste exemplo, nós vemos que o módulo, ```sys"`, está afetado; conseqüentemente, esta vulnerabilidade afeta um componente usado dentro do *kernel*.

**4**

O campo *Announced* reflete a data em que o aviso de segurança foi publicado, ou anunciado ao mundo. Isto significa que a equipe de segurança verificou que o problema existe e que a correção foi enviada ao repositório de código fonte do FreeBSD.

**5**

O campo *Credits* dá os créditos ao individuo ou organização que notou a vulnerabilidade e a reportou.

6

O campo *Affects* explica quais as *releases* do FreeBSD são afetadas por esta vulnerabilidade. Para o *kernel*, uma rápida olhada na saída do `ident` nos arquivos afetados ajudará em determinar uma revisão. Para *ports*, o número da versão está listado depois do nome do port em `/var/db/pkg`. Se o sistema não faz sincronização com o repositório CVS do FreeBSD e é recompilado diariamente, há possibilidades de estar afetado.

7

O campo *Corrected* indica a data, hora, time offset, e a *release* que foi corrigida.

8

O campo *FreeBSD only* indica se esta vulnerabilidade afeta apenas o FreeBSD, ou se afeta outros sistemas operacionais também.

9

O campo *background* dá informações sobre exatamente o que é o utilitário afetado. Na maioria das vezes porque o utilitário existe no FreeBSD, para que é usado, e um pouco de informações sobre como o utilitário vem a ser.

10

O campo *Problem Description* explica a brecha na segurança em profundidade. Esta pode conter informações do código danificado, ou mesmo como o utilitário pode ser usado maliciosamente para abrir uma brecha na segurança.

[\(11\)](#)

O campo *Impact* descreve que tipo de impacto o problema pode causar no sistema. Por exemplo, este pode ser qualquer coisa como ataque de negação de serviço, atribuir privilégios extras disponíveis a usuários, ou dar acesso de superusuário ao atacante.

[\(12\)](#)

O campo *Workaround* oferece um feasible workaround para os administradores que podem estar impossibilitados de atualizar o sistema. Isto pode ser devido constraints de horario, disponibilidade de rede, ou uma série de outras razões. De qualquer forma, a segurança não deve ser taken lightly, e um sistema afetado deve ser corrigido ou o workaround para a brecha na segurança deve ser implementado.

[\(13\)](#)

O campo *Solution* oferece instruções para corrigir o sistema afetado. Este é um método passo-a-passo testado e verificado para obter um sistema corrigido e trabalhando com segurança.



## [\(14\)](#)

O campo *Correction Details* mostra a branch CVS ou nomes de *release* com os pontos trocados para caracteres *underscore*. Ele também mostra o número da revisão dos arquivos afetados em cada branch.

## [\(15\)](#)

O campo *References* geralmente oferece fontes de outras informações. Este pode incluir URLs da web, livros, listas de discussão, e grupos de notícia.

---

# Capítulo 11. Impressão

## Índice

- 11.1. [Sinopse](#)
- 11.2. [Introdução](#)
- 11.3. [Configuração Básica](#)
- 11.4. [Configuração Avançada de Impressora](#)
- 11.5. [Usando as Impessoras](#)
- 11.6. [Alternativas para o Spooler Padrão](#)
- 11.7. [Correção de Erros](#)

*Contributed by Sean Kelly. Reestruturado e atualizado por Jim Mock.*

## 11.1. Sinopse

O FreeBSD pode ser usado para imprimir em uma grande variedade de impressoras, desde as mais antigas impressoras matriciais até as modernas laser, permitindo produzir trabalhos de alta qualidade em seus aplicativos preferidos.

O FreeBSD pode também ser utilizado como servidor de impressão em rede; com isso, o FreeBSD pode receber trabalhos de impressão de uma grande variedade de outros computadores, inclusive de outros FreeBSD, Windows® e Mac OS®. O FreeBSD assegura que apenas um pedido será impresso por vez, e mantém a estatísticas de cada usuário e de cada máquina que fez uso do recurso de impressão, produzindo páginas de ``rosto'', mostrando de quem é a impressão atual.

Após a leitura deste capítulo, você estará sabendo:

- Como configurar o FreeBSD como um servidor de impressão.
- Como instalar filtros de impressão, para gerenciar trabalhos de impressão especiais diferentemente, incluindo conversores de formatos para o formato padrão da sua impressora.
- Como habilitar cabeçalhos ou páginas de rosto à sua impressão.
- Como imprimir em impressoras conectadas em outros computadores.
- Como imprimir em impressoras conectadas diretamente na rede.

- Como controlar restrições para impressora, incluindo limitação de tamanho dos trabalhos de impressão e impedindo certos usuários de utilizarem a(s) impressora(s).
- Como manter estatísticas de impressão e de usuários.
- Como resolver problemas de impressão.

Antes de se aventurar na leitura deste capítulo, você deve:

- Saber como configurar e instalar um novo *kernel* ([Capítulo 9](#)).
- 

## 11.2. Introdução

Para usar impressoras no FreeBSD, você irá precisar configura-lo para trabalhar com o sistema de impressão da Berkeley, também conhecido como sistema de *spool LPD*. Ele também é o sistema padrão de controle do FreeBSD. Esse capítulo introduz ao sistema de *spool LPD*, frequentemente chamado de **LPD**, e irá guia-lo através de sua configuração.

Se você já está familiarizado com o **LPD** ou com outro sistema de impressão, você poderá pular esta seção. e ir para [Configurando o sistema de Impressão](#).

O **LPD** controla tudo sobre o servidor de impressão. É responsável por um grande número de coisas:

- Ele controla o acesso as impressoras locais e impressoras remotas em outras máquinas e impressoras conectadas diretamente à rede.
- Ele habilita os usuários a enviar arquivos para serem impressos; estes envios são chamados de *jobs* (trabalhos de impressão).
- Ele permite que vários usuários acessem uma única impressora ao mesmo tempo, mantendo uma *queue* (fila de impressão) para cada impressora.
- Ele pode imprimir *folhas de rosto* (também conhecidas como *banner* ou *burst*, então os usuários podem com facilidade encontrar seus trabalhos de impressão na pilha de impressos.
- Ele se encarrega dos parâmetros das conexões de impressoras seriais.
- Ele pode enviar trabalhos de impressão através da rede para outro sistema **LPD** em uma outra máquina (servidor).
- Ele pode executar filtros especiais para formatar trabalhos de impressão para serem impressos em várias linguagens de impressora ou para compatibilizar as impressoras.
- Ele pode gerenciar o uso da impressora.

Através do arquivo de configuração (`/etc/printcap`), e por providenciar programas especiais de filtro, você pode habilitar o sistema **LPD** para fazer tudo ou alguma parte do conjunto das funções de uma grande variedade de impressoras.

### 11.2.1. Como Você Deve Utilizar o Spooler

Se você é um usuário único do sistema você deve se surpreender. Por que você deveria se preocupar com o *spooler* quando você não necessita de controle de acesso, páginas de cabeçalho ou controle de impressão. Mesmo que fosse possível habilitar o acesso direto a impressora, você deverá usar o *spooler* de qualquer maneira. Veja algumas vantagens:

- O **LPD** imprime o trabalho de impressão em segundo plano, você não precisa aguardar os dados serem enviados para a impressora.
- O **LPD** pode convenientemente executar um trabalho de impressão para ser impresso através de filtros para adicionar cabeçalhos com data/hora ou converter formatos especiais de arquivo (como o arquivos **TeX** DVI) em formatos que as impressoras poderão entender. Você não precisará fazer manualmente.
- Muitos programas gratuitos e comerciais preveem a impressão através da funcionalidade de conversar com o *spooler* do seu sistema. Pela configuração do sistema de impressão, você irá mais facilmente suportar outros programas que você poderá adicionar ou já ter.

---

## 11.3. Configuração Básica

Para usar as impressoras com o sistema **LPD** de impressão, você irá precisar de configurar tanto a impressora quanto o programa **LPD**. Esse documento descreve dois níveis de configuração:

- Veja a seção [Configuração Simples da Impressora](#) para aprender como conectar a impressora, informar ao **LPD** como se comunicar com a(s) impressora(s), e imprimir arquivo em texto puro nelas.
- Veja a seção [Configuração Avançada de Impressora](#) para encontrar como imprimir uma variedade de formatos especiais de arquivo, para imprimir cabeçalhos nas páginas, para imprimir através da rede, para controlar o acesso às impressoras, e fazer o balanço da utilização da impressora.

### 11.3.1. Configuração Básica

Essa seção mostra como configurar a impressora e o programa **LPD** para usá-la. Ela irá ensinar o básico:

- Seção [Configuração de Hardware](#)

Dá algumas dicas na conexão da impressora com a porta do seu computador.

- Seção [Configuração de Programas](#) mostra como configurar o arquivo de configuração (`/etc/printcap`) do **LPD**.

Se você estiver configurando a impressora para usar o protocolo de rede para aceitar dados para imprimir ao invés de uma porta paralela ou serial, veja [Impressoras com Interfaces de Dados Através da Rede](#).

Apesar desta seção ser chamada de "Configuração Simples da Impressora" é um pouco complexa. Fazer com que a impressora funcione com seu computador e o LPD é uma parte difícil da configuração. As opções avançadas como cabeçalhos de páginas e gerenciamento são facilmente configuráveis para funcionar na impressora.

### 11.3.1.1. Configuração de Hardware

Esta seção mostra vários meios para você conectar a impressora no seu PC. Fala sobre portas de comunicação e cabos, e também sobre configuração do kernel que será necessária habilitar para que o FreeBSD possa comunicar-se com a impressora.

Se você já tem uma impressora conectada e obteve sucesso em imprimir em outro sistema operacional, você provavelmente poderá pular para a seção [Configuração de Programas](#).

#### 11.3.1.1.1. Portas e Cabos

A grande maioria de impressoras que você pode ter para usar em um PC, possui suporte a pelo menos uma das interfaces abaixo:

- Interface *serial*: usa a porta serial do seu computador para enviar dados para a impressora. A interface serial é comumente encontrada em computadores industriais, os cabos estão disponíveis e também são de fácil construção. Interfaces seriais, algumas vezes necessitam de cabos especiais e normalmente requerem que você configure algumas opções complexas da comunicação.
- Interface *paralela*: usa a porta paralela do seu computador para enviar dados para a impressora. Interface paralela é mais comum em PC. Os cabos são disponibilizados mais facilmente, mas são mais difíceis de serem montados manualmente. Normalmente não existem opções para serem configuradas nas interfaces paralelas, tornando a configuração extremamente mais simples.

Interfaces paralelas são normalmente conhecidas como interfaces "Centronics", nome este dado por causa do tipo de conector utilizado na impressora.

Em geral, as interfaces seriais são mais lentas que as paralelas. Interfaces paralelas normalmente são disponibilizadas em uma única direção para comunicação (do computador para a impressora) enquanto que a serial proporciona uma comunicação bi-direcional. As novas portas paralelas (EPP e ECP) e as impressoras podem se comunicar em ambas as direções sob o FreeBSD quando um cabo padrão IEEE1284 estiver sendo usado.

Usualmente, você somente precisará da comunicação bi-direcional se sua impressora reconhecer PostScript®. Impressora PostScript pode ser muito "tagarela". De fato trabalhos de impressão em PostScript são as saídas atuais que os programas enviam para as impressoras; Eles não precisam produzir papel como um todo e pode retornar resultados diretamente para o computador. PostScript também usa comunicação bi-

direcional para informar ao computador sobre problemas ocorridos, tais como erros em programas PostScript ou papel embaralhado (paper jam). Seus usuários podem ficar agradecidos com essas informações. Entretanto, o melhor meio para fazer o controle efetivo com uma impressora PostScript requer que a comunicação seja bi-direcional: você pergunta a impressora sobre a contagem de páginas (quantas páginas ela já imprimiu durante seu uso, vida útil), então envia o trabalho de impressão do usuário, faz novamente a pergunta à impressora sobre o total de páginas impressas e então faz a subtração dos valores para saber o valor exato de páginas impressa pelo usuário.

#### **11.3.1.1.2. Porta Paralela**

Para conectar a impressora usando a interface paralela, utilize o cabo Centronics entre a impressora e o computador. As instruções que vem no manual da impressora e/ou do computador, devem dar a você um passo-a-passo completo.

Lembre-se qual porta paralela você está usando no computador. A primeira porta paralela é `/dev/ppc0` para o FreeBSD; a segunda é `/dev/ppc1`, e assim por diante. Os dispositivos de impressora usam o seguinte esquema de nomes: `/dev/lpt0` Para a impressora conectada na primeira porta paralela etc.

#### **11.3.1.1.3. Porta Serial**

Para conectar a impressora em uma porta serial, você deverá possuir o cabo serial apropriado para interligar o computador com a impressora. As instruções que vem com a impressora e/ou com o computador, deverá dar à você um completo passo-a-passo.

Se você não está seguro sobre qual o tipo de "cabo serial" que está usando, você pode tentar uma dessas alternativas:

- O cabo tipo *modem* conecta cada pino do conector de um lado ao correspondente do outro conector. Esse tipo de cabo também é conhecido como cabo "DTE-to-DCE".
- O cabo tipo *null-modem* conecta alguns dos pinos diretamente e troca outros (como por exemplo os pinos responsáveis por enviar e receber dados), curta-circuita outros (liga-os entre si) internamente em cada conector. Esse tipo de cabo é também conhecido como cabo "DTE-to-DTE"
- O cabo de *impressora serial*, é necessário para algumas impressoras não comuns, é semelhante ao cabo null-modem, mas envia alguns sinais que estão curto-circuitados internamente.

Você deverá configurar alguns parametros da comunicação da impressora, que usualmente são feitos no painel frontal ou nas mini chaves (DIP switches). Altere o valor de *bps* (bits por segundos, chamado frequentemente de *baud rate*) que é a taxa de transferência entre o computador e a impressora, ambos devem ter a mesma taxa de transferência. Selecione entre 7 e 8 bits; paridade par, ímpar ou nenhuma; ou 1 ou 2 *stop bits*. Também o protocolo de controle de fluxo; nenhum ou XON/XOFF (também conhecido como controle de fluxo por "hardware" ("*in-band*") ou por "software". Lembre-se dessas configurações para poder aplica-las na configuração dos programas, como vamos verificar a seguir.

### 11.3.1.2. Configuração de Programa

Essa seção descreve as configurações necessárias para imprimir com o sistema **LPD** no FreeBSD.

Um resumo do que será visto neste passo:

1. Configure o seu kernel, se necessário, para utilizar a porta que você deseja conectar a sua impressora. A seção [Configuração do Kernel](#) informa à você o que é necessário fazer.
2. Configure o modo de comunicação da porta paralela, se você está usando a porta paralela; veja a seção [Configure o Modo de Comunicação para a Porta Paralela](#) para obter mais detalhes.
3. Teste se o sistema operacional pode enviar dados para a impressora. A seção [Verificando a Comunicação com a Impressora](#) dá algumas sugestões de como fazer isto.
4. Configure o **LPD** para a utilização da impressora, modificando o arquivo `/etc/printcap`. Você verá o que fazer mais tarde neste capítulo.

#### 11.3.1.2.1. Configuração do Kernel

O kernel do sistema operacional é compilado para trabalhar com um determinado conjunto de dispositivos. As interfaces serial e paralela para a utilização da impressora fazem parte deste conjunto. Entretanto, pode ser necessário adicionar o suporte para uma porta serial ou paralela caso o seu kernel não esteja configurado.

Para saber se seu kernel já possui o suporte para interface serial, digite:

```
grep sioN /var/run/dmesg.boot
```

Onde *N* é o número da porta serial, iniciando do zero. Se você ver algo semelhante à:

```
sio2 at port 0x3e8-0x3ef irq 5 on isa
sio2: type 16550A
```

Então o kernel suporta a porta.

Para saber se seu kernel já possui o suporte para interface paralela, digite:

```
grep ppcN /var/run/dmesg.boot
```

Onde *N* é o número da porta paralela, iniciando do zero. Se você ver algo semelhante à:

```
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
ppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/8 bytes threshold
```

Então o kernel suporta a porta.

Você pode ter que reconfigurar o seu kernel para que o sistema operacional possa reconhecer e usar as portas paralela e/ou serial, para que você possa utilizar a impressora.

Para adicionar suporte a porta serial, consulte a seção de configuração do kernel. Para adicionar suporte a porta paralela consulte esta seção e a seção a seguir.

### 11.3.1.3. Adicionando Entradas para as Portas no `/dev`

**Nota:** FreeBSD 5.0 inclui o sistema de arquivos *devfs*, o qual automaticamente cria os nós dos dispositivos necessários. Se você está executando a versão do FreeBSD com *devfs* habilitado, você pode pular com segurança esta seção.

Uma vez que o kernel já suporta a comunicação ou pela porta paralela ou porta serial, você ainda precisa da interface de programa através da qual você irá enviar e receber dados. É para isso que server a entrada no diretório `/dev` server.

*Para adicionar uma entrada para a porta no diretório `/dev` siga os passos:*

1. Torne-se o usuário `root` com o comando [su\(1\)](#). Entre com a senha do usuário `root` quando for solicitado.
2. Mude para o diretório `/dev`
3. `# cd /dev`
4. Digite:
5. `# ./MAKEDEV porta`
- 6.

Onde a *porta* é a entrada para o dispositivo para a porta que você quer criar. Use *lpt0* para a impressora conectada na primeira porta paralela, *lpt1* para a segunda porta paralela, e assim consecutivamente; use *ttyd0* para a primeira porta serial, *ttyd1* para a segunda e assim consecutivamente.

7. Digite:
8. `# ls -l porta`

Para ter certeza de que a entrada do dispositivo foi criada.

#### 11.3.1.3.1. Configurando o Modo de Comunicação para a Porta Paralela

Quando você for usar a interface paralela, você deve escolher entre o modo utilizando interrupção ou por *polling*. Driver para o dispositivo genérico de impressora ([lpt\(4\)](#)) para FreeBSD 4.X e 5.X usa o sistema [ppbus\(4\)](#), o qual controla a porta do chipset com o driver [ppc\(4\)](#) driver.

- O método de *interrupção* é o padrão com o kernel GENERIC. Com esse método o sistema operacional usa uma linha de interrupção IRQ para determinar se a impressora está pronta para receber os dados.
- O método de *polled* direciona o sistema operacional para perguntar repetidamente se a impressora está pronta para receber dados. Quando a resposta é afirmativa, o sistema envia os dados.

O método de interrupção é normalmente mais rápido mas usa uma linha de IRQ preciosa. Algumas das novas impressoras HP não funcionam corretamente neste modo, aparentemente é um problema de temporização (ainda não completamente compreendido). Estas impressoras necessitam do modo de *polling*. Você deve utilizar um dos dois métodos para trabalhar. Algumas impressoras irão trabalhar em ambos os modos, mas ficaram sensivelmente mais lentas no modo de interrupção

Você pode configurar o modo de comunicação de duas maneiras: pela configuração do kernel ou usando o programa [lptcontrol\(8\)](#)

*Para configurar o modo de comunicação pelo kernel:*

1. Você deve editar o arquivo de configuração do kernel. Procure pela entrada *ppc0*. Se você for configurar a segunda porta paralela, use *ppc1*. Use *ppc2* para a terceira porta, e assim consecutivamente.
  - o Se você preferir o modo de interrupção, para FreeBSD 4.X inclua a especificação da *irq*:
  - o `device ppc0 at isa? irq N`

Onde *N* é a IRQ para a porta paralela do seu computador.

Para o FreeBSD 5.X, edite a seguinte linha:

```
hint.ppc.0.irq="N"
```

No arquivo `/boot/device.hints` e troque *N* pelo número correto da IRQ. O arquivo de configuração do kernel também deve conter o driver [ppc\(4\)](#)

```
device ppc
```

- o Se você for utilizar o modo de *polling*, não adicione a especificação da *irq*:

Para o FreeBSD 4.X, use a seguinte linha no arquivo de configuração do kernel:

```
device ppc0 at isa?
```

Para o FreeBSD 5.X, simplesmente remova em seu arquivo `/boot/device.hints` a seguinte linha:

```
hint.ppc.0.irq="N"
```

Em alguns casos, isso não será suficiente para colocar a porta em modo de *polling* no FreeBSD 5.X. Na maior parte das vezes, isso é devido ao driver [acpi\(4\)](#), mas tarde habilita para detecção e *attach* os dispositivos, e então controla o modo de acesso à porta da impressora. Você deve checar sua configuração [acpi\(4\)](#) para corrigir esse problema.



2. Salve o arquivo. Então configure, compile e instale o novo kernel, e a seguir reinicie o seu sistema. Veja [Configuração do Kernel](#) para maiores detalhes.

Para configurar o modo de comunicação com [lptcontrol\(8\)](#):

1. Digite:
2. # `lptcontrol -i -d /dev/lptN`

para configurar o modo de interrupção para a *lptN*.

3. Digite:
4. # `lptcontrol -p -d /dev/lptN`

Para configurar o modo de *polling* para *lptN*.

Você pode colocar esses comandos no arquivo `/etc/rc.local` para configurar automaticamente na hora da inicialização do seu sistema. Veja [lptcontrol\(8\)](#) para maiores detalhes.

#### 11.3.1.3.2. Testando a Comunicação com a Impressora

Antes de prosseguir com a configuração do sistema *spooling*, você deve ter certeza que o seu sistema operacional teve sucesso em enviar dados para a impressora. É muito mais fácil depurar a conexão com a impressora e o sistema de *spooling* separadamente.

Para fazer o teste da impressora, precisamos enviar algum texto para a impressora. Para impressoras que podem imprimir caracteres diretamente, o programa [lptest\(1\)](#) é perfeito: ele gera todos os 96 caracteres imprimíveis da tabela ASCII em 96 linhas.

Para a impressora PostScript (ou outra linguagem) nos iremos precisar de algo mais sofisticado para testar. Um pequeno programa PostScript como o que segue, é o suficiente

```
%!PS
100 100 moveto 300 300 lineto stroke
310 310 moveto /Helvetica findfont 12 scalefont setfont
(Is this thing working?) show
showpage
```

O código PostScript acima pode ser colocado em um arquivo e usado como mostra os exemplos nas seções a seguir.

**Nota:** Quando esse documento refere-se a linguagem da impressora, é assumido que a linguagem é semelhante ao PostScript, e não o PCL da Hewlett Packard. Ainda que a linguagem PCL tenha grande funcionalidade, você pode mesclar texto puro com sequências de controle com caracteres ESC. PostScript não pode ser impresso diretamente em impressoras de texto puro, e para impressoras que necessitem de outra linguagem para imprimir, será necessário acomodações.

#### 11.3.1.3.2.1. Verificando a Impressora Paralela

Essa seção informa à você como verificar se o FreeBSD pode comunicar-se com a impressora conectada na porta paralela.

*Para testar a impressora na porta paralela:*

1. Torne-se root com [su\(1\)](#).
2. Envie Dados para a impressora.
  - o Se a impressora poder imprimir texto puro, use [lptest\(1\)](#). Digite:  
# `lptest > /dev/lptN`

Onde *N* é o número da porta paralela, iniciando em 0

- o Se a impressora entender PostScript ou outra linguagem de impressora, então envie um pequeno programa para a impressora. Digite:  
# `cat > /dev/lptN`

Então a linha por linha, digite o programa *cuidadosamente* como você não poderá editar a linha caso tenha pressionado a tecla *RETURN* ou *ENTER*. Quando você terminar de digitar o programa, pressione as teclas *CONTROL+D*, ou tecla que for fim-de-arquivo.

Alternativamente, você pode colocar o programa em um arquivo e digitar:

```
cat arquivo > /dev/lptN
```

Onde *arquivo* é o nome do arquivo a ser digitado, que contenha o programa que você envia para a impressora.

Você deverá ver algo semelhante impresso. Não se preocupe se o texto não sair como esperado; nos iremos consertar algumas coisas mais tarde.

#### 11.3.1.3.2.2. Verificando a Impressora Serial

Essa seção informa à você como verificar se o FreeBSD pode comunicar com a impressora pela porta serial.

*Para testar a impressora na porta serial:*

1. Torne-se root com [su\(1\)](#).
2. Edite o arquivo `/etc/remote`. Adicione a seguinte entrada:
3. `printer:dv=/dev/porta:br#bps-rate:pa=paridade`

Onde *porta* é o dispositivo para a porta serial (*ttyd0*, *ttyd1*, etc.), *bps-rate* é a taxa em bits-por-segundo pela qual a impressora irá se comunicar, e *paridade* é a paridade requerida pela impressora (deve ser uma dessas, a seguir: *par*, *ímpar*, *nenhum*, ou *zero*).

Aqui tem um exemplo de uma impressora conectada via terceira porta serial a uma taxa de 19200 bps sem paridade:

```
printer:dv=/dev/ttyd2:br#19200:pa=none
```

4. Conectando a impressora com [tip\(1\)](#). Digite:
5. # tip printer

Se esse passo não funcionar, edite o arquivo `/etc/remote` e tente usando `/dev/cuaaN` ao invés de `/dev/ttydN`.

6. Envie dados para a impressora.
  - o Se a Impressora pode imprimir texto puro, use [lptest\(1\)](#). Digite:  
% \$lptest
  - o Se a impressora entende PostScript ou outra linguagem, envie um pequeno programa para a impressora. Digite o programa linha-a-linha  *muito cuidadosamente* porque as teclas de edição podem ter significado para a impressora. Você também pode precisar digitar o caracter especial de fim-de-arquivo para a impressora reconhecer que recebeu todo o programa. Para a impressora PostScript, pressione *CONTROL+D*.

Alternativamente, você pode colocar o programa em um arquivo e digitar:

```
% >arquivo
```

Onde *arquivo* é o nome do arquivo que contém o programa. Após [tip\(1\)](#) envia o arquivo, pressione a sequencia fim-de-arquivo.

Você deve ver algo impresso. Não se preocupe se o texto não se parecer correto; nós iremos consertar isso mais tarde.

#### **11.3.1.4. Habilitando o Spooler: O Arquivo `/etc/printcap`**

Nesse ponto, sua impressora deve estar conectada, seu kernel configurado e comunicando com a impressora, e você teve sucesso ao enviar dados para a impressora. Agora, nos estamos prontos para configurar **LPD** para controlar o acesso a sua impressora.

Você configura o **LPD** pela edição do arquivo `/etc/printcap`. O sistema de *spooling* **LPD** lê este arquivo a cada vez que o *spooler* é necessário então a atualização do arquivo se reflete imediatamente na aplicação.

O formato do arquivo [printcap\(5\)](#) é direto. Use o seu editor de texto favorito para realizar as mudanças no arquivo `/etc/printcap`. O formato é idêntico aos outros arquivos de compatibilidades como os `/usr/share/misc/termcap` e `/etc/remote`. Para completa informação sobre o formato, veja [cgetent\(3\)](#).

A simples configuração do *spooler* consiste em seguir os seguintes passos:

1. Escolha um nome (e algum apelido conveniente) para a sua impressora, e coloque-o no arquivo `/etc/printcap`. Veja a seção [Nomeando a Impressora](#) para maiores informações sobre dar nomes.
2. Para desligar o cabeçalho de páginas (ele é habilitado por padrão) inserindo a chave `sh`; veja a seção [Suprimindo o Cabeçalho de Páginas](#) para maiores informações.
3. Criando o diretório de *spooling*, e especificando a sua posição com a chave `sd`; veja a seção [Criando o Diretório de Spooling](#) para maiores informações.
4. Veja a entrada `/dev` para usar a sua impressora, e coloque ela dentro do arquivo `/etc/printcap` na chave `lp`; veja a seção [Identificação do Dispositivo de Impressão](#) para maiores informações. Se a impressora estiver na porta serial, configure os parâmetros de comunicação use a chave `the ms#` o qual é discutido na seção [Configurando os Parâmetros Comunicação do Spooler](#).
5. Instalar um filtro de entrada para texto puro; veja a seção [Instalando o Filtro de Texto](#) para mais detalhes:
6. Testando a configuração, imprimindo alguma coisa com o comando `lpr(1)`. Mais detalhes estão disponíveis em [Tente isso](#) e [Localização de Defeitos](#)

**Nota:** Impressora baseada em linguagem, como impressoras PostScript, não podem imprimir diretamente texto puro. A simples configuração descrita acima e descrita nas seções seguintes assume que se você está instalando uma impressora que você só irá imprimir arquivos que ela possa entender.

Usuários geralmente têm expectativa que eles possam imprimir texto puro para qualquer impressora instaladas em seu sistema. Programas que fazem interface entre o **LPD** e a impressora. Se você está instalando a impressora e gostaria de habilitar os *jobs* na linguagem da impressora e texto puro, você será fortemente encorajado a adicionar mais um passo ao esquema acima; instalar um programa de conversão de textopuro para PostScript (ou outra linguagem de impressora). A seção intitulada como [Acomodando Jobs de Texto Puro em impressoras PostScript](#) mostra como fazê-lo.

#### **11.3.1.4.1. Nomeando a Impressora**

O primeiro e mais simples passo é escolher o nome para a sua impressora. Isso é; você deve escolher um nome funcional e cuidadosamente escolhido, deve também escolher apelidos para a sua impressora.

Pelo menos uma das impressoras especificadas no arquivo `/etc/printcap` deve ser um apelido para `lp`. Essa é o nome padrão para a impressora. Se o usuário não possuir a variável do ambiente `PRINTER` definida ou não especificar o nome da impressora na linha de comando de qualquer aplicativo que use **LPD**, então a impressora `lp` será usada como padrão.

Também é uma prática comum fazer um apelido para um nome que define completamente a impressora, incluindo a marca e o modelo.

Uma vez que você já tenha escolhido um nome e alguns apelidos, coloque eles no arquivo `/etc/printcap`. O nome da impressora deve estar na coluna mais a esquerda. Separando cada apelido com uma barra vertical (pipe) e terminar com o caracter `:` (dois pontos)

O exemplo a seguir é um esqueleto para o arquivo `/etc/printcap` que define duas impressoras (uma Diablo 630 matricial e uma Panasonic KX-P4455 PostScript que é uma impressora à laser):

```
#
/etc/printcap for host rose
#
rattan|line|diablo|lp|Diablo 630 Line Printer:
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:
```

Nesse exemplo, a primeira impressora é chamada de *rattan* e possui como apelido *line*, *diablo*, *lp*, e *Diablo 630 Line Printer*. Como ela tem como apelido *lp*, ela é a impressora padrão para o sistema. E a segunda é chamada de *bamboo*, e tem como apelidos *ps*, *PS*, *S*, *panasonic*, e *Panasonic KX-P4455 PostScript v51.4*.

#### 11.3.1.4.2. Suprimindo a Página de Cabeçalho

O sistema de spooling **LPD** por padrão imprime uma *página de cabeçalho* para cada *job*. A página de cabeçalho contém o nome do usuário que pediu o serviço, a máquina de onde o pedido originou, e o nome do *job*, em letras grandes. Para testes iremos desativar a página de cabeçalho.

Para desabilitar a página de cabeçalho, adicione a entrada *sh* no arquivo `/etc/printcap`. Segue um exemplo do arquivo `/etc/printcap` com essa opção *sh* desabilitada.

```
#
/etc/printcap for host rose - no header pages anywhere
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
 :sh:
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
 :sh:
```

Note como usamos o correto formato: A primeira linha inicia na coluna mais à esquerda, e as linhas subsequentes são alinhadas com uma caracter TAB simples. Cada linha, exceto a última, termina com uma barra invertida (`\`).

#### 11.3.1.4.3. Criando o Diretório de Spooling

O próximo passo na configuração simples do *spooler* é criar o *diretório de spooling*, é o diretório onde o *jobs* mantém o que será impresso, e outros arquivos de suporte do *spooler*.

Por causa da variação natural do diretório de *spooling*, ele é colocado sob o diretório `/var/spool`. Não é necessário realizar *backups* do conteúdo deste diretório. Você pode recriá-lo com o comando [mkdir\(1\)](#).

Ele também cria o diretório com o nome idêntico ao da impressora, como é mostrado abaixo:

```
mkdir /var/spool/printer-name
```

Entretando, se você possui uma grande quantidade de impressoras na sua rede, você pode querer colocar o diretório de *spooling* em um único diretório que você reservou para a impressão com **LPD**. Nós iremos fazer isso com exemplo de duas impressoras *rattan* e *bamboo*:

```
mkdir /var/spool/lpd
mkdir /var/spool/lpd/rattan
mkdir /var/spool/lpd/bamboo
```

**Nota:** Se você está preocupado com a privacidade dos *jobs* que seus usuarios estão imprimindo, você talvez queira proteger o diretório de *spooling* que o mesmo não tenha acesso público. O diretório de *spooling* deve ter como dono o usuário e grupo *daemon* e deverá ser acessível para leitura, escrita e execução e para mais ninguém. Veja o próximo exemplo onde será feito isso:

```
chown daemon:daemon /var/spool/lpd/rattan
chown daemon:daemon /var/spool/lpd/bamboo
chmod 770 /var/spool/lpd/rattan
chmod 770 /var/spool/lpd/bamboo
```

Finalmente, você deverá informar ao **LPD** sobre este diretório, usando o arquivo */etc/printcap*. Você especifica o nome do diretório de *spooling* com a opção *sd*:

```
#
/etc/printcap for host rose - added spooling directories
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
 :sh:sd=/var/spool/lpd/rattan:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
 :sh:sd=/var/spool/lpd/bamboo:
```

Note que o nome da impressora inicia na primeira coluna mas as outras entradas deverá ser alinhada com um caracter TAB e cada linha deverá terminar com uma barra invertida (\).

Se você não especificar o diretório de *spooling* com *sd*, o sistema de *spooling* irá usar o diretório padrão */var/spool/lpd*

#### **11.3.1.4.4. Identificando o Dispositivo de Impressão**

Na seção [Adicionando Entradas no /dev Para Portas](#), identificamos qual entrada no diretório */dev* o FreeBSD irá usar para se comunicar com a impressora. Agora, nós iremos informar ao **LPD** qual porta usar. Quando o sistema de *spooling* tem um *job* para imprimir, ele irá abrir o dispositivo especificado no suporte do programa de filtro (o qual é responsável por passar os dados para a impressora).

Listar as entradas do diretório */dev* no arquivo */etc/printcap* usando a opção *lp*.

Em nosso exemplo executado, nós assumimos que a impressora *rattan* está na primeira porta paralela e a impressora *bamboo* está na sexta porta serial; segue aqui as adições ao arquivo `/etc/printcap`:

```
#
/etc/printcap for host rose - identified what devices to use
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
 :sh:sd=/var/spool/lpd/rattan:\
 :lp=/dev/lpt0:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
 :sh:sd=/var/spool/lpd/bamboo:\
 :lp=/dev/ttyd5:
```

Se você não especificar a opção *lp* para a impressora em seu arquivo `/etc/printcap`, o **LPD** irá utilizar `/dev/lp` como padrão. A entrada `/dev/lp` não existe mais no FreeBSD.

Se a impressora que você está instalando é conectada na porta paralela, pule para a seção intitulada como [Instalando o Filtro de Texto](#). Caso contrário, siga as instruções na próxima seção

#### **11.3.1.4.5. Configurando Parametros de Comunicação do spooling**

Para impressoras na porta serial, no **LPD** podemos ajustar a taxa de bps, a paridade, e outros parâmetros da porta serial com o apoio do programa de filtro que envia dados para a impressora. Isso pode ser uma grande vantagem uma vez que:

- Permite tentar diferentes configurações de parametros para a comunicação, simplesmente editando o arquivo `/etc/printcap`; você não precisa recompilar o programa de filtro.
- Ele habilita o sistema de *spooling* para usar o mesmo programa de filtro para multiplas impressoras, as quais têm diferentes configurações de comunicação serial.

As seguintes entradas do arquivo `/etc/printcap` possuem habilidade para controlar os parâmetro da comunicação serial do dispositivos listados nas habilidades do *lp*:

*br#bps-rate*

Configura a velocidade de comunicação com o dispositivo para *bps-rate*, onde *bps-rate* pode ser 50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, ou 115200 bits-per-second.

*ms#stty-mode*

Configura a opção do dispositivo terminal depois de aberto o dispositivo. [stty\(1\)](#) explica a opções disponíveis

Quando o **LPD** abre o dispositivo especificado pela opção *lp*, ele configura as características do dispositivos para aquelas especificadas na opção *ms#*. De particular interesse podemos destacar os modos *parenb*, *parodd*, *cs5*, *cs6*, *cs7*, *cs8*, *cstopb*, *crtscs*, e *ixon* os quais serão explicados na página do manual [stty\(1\)](#)

Vamos mostrar um exemplo de configuração de uma impressora serial conectada na sexta porta serial. Nós iremos configurar com a taxa de bps para 38400. Para o modo nós iremos configurar sem paridade com a opção *-parenb*, caracteres de 8-bit *cs8*, sem controle de modem *cllocal* e *hardware flow control* *crtscs*:

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
:sh:sd=/var/spool/lpd/bamboo:\
:lp=/dev/ttyd5:ms#-parenb cs8 cllocal crtscs:
```

#### 11.3.1.4.6. Instalando um Filtro de Texto

Nós já estamos prontos para informar ao **LPD** qual o filtro de texto que queremos utilizar para enviar *jobs* para a impressora. Um *filtro de texto*, também conhecido como *filtro de entrada*, é um programam que é executado pelo **LPD** quando um tem *job* para ser impresso. Quando o **LPD** executa o filtro de texto para a impressora, ele configura a entrada padrão do filtro para o *job* a ser impresso, e a sua saída padrão é enviada para a impressora especificada em *lp*. O filtro está pronto para receber o *job* pela entrada padrão, executando a tradução necessária para a impressora, e escrevendo o resultado na saída padrão, a qual será enviada à impressora. Para maiores informações, consulte a seção de [Como os Filtros funcionam](#)

Para essa configuração simples, o filtro pode ser um pequeno *shell script* que execute o comando `/bin/cat` para enviar o *job* para a impressora. O FreeBSD vem com um outro filtro chamado `lpf` e pega os caracteres de *backspacing* e sublinhados que algumas impressoras não conseguem manipular, você pode então utilizar outro programa de filtro que desejar. O filtro `lpf` é descrito em detalhes na seção intitulada [lpf: Um Filtro de Texto](#).

Primeiro, iremos criar um *shell script* chamado de `/usr/local/libexec/if-simple` para ser um filtro de texto simples. Coloque o seguinte texto dentro dele, utilizando o seu editor de texto favorito:

```
#!/bin/sh
#
if-simple - Simple text input filter for lpd
Installed in /usr/local/libexec/if-simple
#
Simply copies stdin to stdout. Ignores all filter arguments.

/bin/cat && exit 0
exit 2
```

Deixe o arquivo com permissão de execução

```
chmod 555 /usr/local/libexec/if-simple
```



E então diga ao **LPD** que utilize esse filtro, especificando a opção *if* dentro do arquivo `/etc/printcap`. Nós iremos adicionar duas impressoras como no exemplo do `/etc/printcap`:

```
#
/etc/printcap for host rose - added text filter
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
:sh:sd=/var/spool/lpd/rattan:\ :lp=/dev/lpt0:\
:if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
:sh:sd=/var/spool/lpd/bamboo:\
:lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:\
:if=/usr/local/libexec/if-simple:
```

#### 11.3.1.4.7. Ligando o LPD

[lpd\(8\)](#) é executado a partir do arquivo `/etc/rc`, controlado pela variável `lpd_enable`. O valor padrão para esta variável é *NO*. Se você ainda não o fez, adicione a linha

```
lpd_enable="YES"
```

ao arquivo `/etc/rc.conf`, e então reinicie a sua máquina, ou então execute simplesmente [lpd\(8\)](#).

```
lpd
```

#### 11.3.1.4.8. Tente Isso

Você chegou ao fim da configuração simples do **LPD**. Infelizmente, ainda não é hora de *cantar vitória*, uma vez que ainda não testamos a configuração e correção de algum problema. Para testar a configuração, tente imprimir algo. Para imprimir com o sistema **LPD**, você usa o comando [lpr\(1\)](#), que envia o *job* para a impressora.

Você pode combinar o [lpr\(1\)](#) com o programa [lptest\(1\)](#), introduzido na seção [Verificando a comunicação com a Impressora](#) para gerar texto para texto.

*Para testar a configuração simples do LPD*

Digite:

```
lptest 20 5 | lpr -PNome-da-impressora
```

Onde *Nome-da-impressora* é o nome dado à impressora (ou um de seus apelidos) no arquivo `/etc/printcap`. Para testar a impressora padrão do sistema, digite [lpr\(1\)](#) sem opções *-P*. Novamente, se você está testando a impressora que espera PostScript, utilize um programa que gere uma saída PostScript ao invés de utilizar o [lptest\(1\)](#). Você pode colocar um código PostScript dentro de um arquivo e digitar `lpr nome do arquivo`.

Para uma impressora PostScript, você deve saber quais foram os dados enviados do programa para a impressora, para saber se este foi impresso corretamente. Se você estiver usando o [lptest\(1\)](#), você deve ter como resultado algo semelhante à:

```
! "#$%&'()*+,-./01234
"#$%&'()*+,-./012345
#$%&'()*+,-./0123456
$%&'()*+,-./01234567
%&'()*+,-./012345678
```

Para facilitar o teste de impressão, tente fazer a transferência de um arquivo grande (para a linguagem da impressora) ou executar [lptest\(1\)](#) com diferentes argumentos. Por exemplo, `lptest 80 60` irá produzir 60 linhas com 80 caracteres cada.

Se a impressora não funcionar, veja a seção intitulada [Localização de Defeitos](#)

---

## 11.4. Configuração Avançada de Impressora

Essa seção descreve filtros para formatos especiais de impressão de arquivos, páginas de cabeçalho, impressão através da rede, e restringir acesso ao uso da impressora.

### 11.4.1. Filtros

Através do **LPD** podemos acessar o protocolos de rede, filas, acesso de controle, e outros aspectos da impressão, a maioria dos trabalhos *reais* ocorre nos *filters*. Os filtros são programas que se comunicam com a impressora e manipulam as dependências do dispositivo e necessidades especiais. Na seção configuração simples da impressora, nós instalamos um filtro--an texto puro extremamente simples que deve funcionar na maioria das impressoras (seção [Instalando o Filtro de Texto](#) ).

Entretando, se quisermos tirar vantagens de conversão de formatos, relatório de uso da impressora, especificar uma configuração especial da impressora, entre outras coisas, devemos compreender como funcionam os filtros. Ultimamente a responsabilidade do filtro é tratar desses aspectos. E a má notícia é que a maior parte do tempo *you* terá que providenciar os seus próprios filtros. A boa notícia é que muitos já são disponíveis; e se não encontrar, você facilmente poderá escrevê-los.

Também, o FreeBSD vem com um `/usr/libexec/lpr/lpf`, que funciona com a maioria das impressoras que podem imprimir texto puro. (Ele manipula caracteres de retorno, *backspaces* e tabulações no arquivo, e faz a contabilidade, mas isso é tudo que ele faz). Existem vários filtros e componentes de filtro na Coleção de Ports do FreeBSD.

Aqui está o que você irá encontrar nessa seção:

- A Seção [Como os Filtros funcionam](#), tenta dar um panorama geral da função de um filtro no processo de impressão. Você deverá ler esta seção para compreender o que acontece "por baixo dos panos" quando **LPD** utiliza os filtros. Esse conhecimento poderá ajuda-lo antecipadamente e livra-lo dos

defeitos que poderam encontrar na instalação de vários filtros para suas impressoras.

- O **LPD** Espera que cada impressora esteja habilitada para imprimir texto puro por padrão. É um problema presente para impressoras PostScript® (ou outra impressora baseada em ligagem) a qual não pode imprimir diretamente texto puro. A seção [Acomodando Impressão em Texto Puro em Impressoras PostScript](#) mostra à você como proceder para resolver este problema. Você deve ler essa seção caso você tenha uma impressora PostScript
- PostScript é o formato mais comum para a maioria dos programas. Mesmo que algumas pessoas (incluindo eu mesmo) escreve código PostScript diretamente, mas impressoras PostScript são caras. Na seção [Simulando PostScript em impressora não PostScript](#) informa como você pode modificar os filtros para aceitar e imprimir dados PostScript em uma impressora *não PostScript*. você deverá ler essa seção se você não possuir uma impressora PostScript.
- A seção [Filtro de Conversão](#) informa sobre o meio que você pode automatizar a conversão especificando o formato do arquivo, tais como gráficos ou dados de *typesetting*, para formatos que a impressora possa entender. Depois de lida esta seção, você está apto a configurar sua impressora para que os usuários possam digitar `lpr -t` para imprimir dados de troff, ou `lpr -d` para imprimir dados TeX DVI data, ou `lpr -v` para imprimir dados de imagem rasterizadas, e por aí vai, recomendamos a leitura desta seção.
- A seção [Filtros de saída](#) Informa sobre o uso não comum dos filtros de saída do **LPD**. A menos que você esteja imprimindo páginas de cabeçalho (veja [Páginas de cabeçalho](#)), você provavelmente poderá pular toda esta seção.
- A Seção [lpf: um filtro de texto](#) chamado `lpf`, completo se considerado um filtro de texto simples para impressoras matriciais (e impressoras laser que funcionam como matriciais) que vem com o FreeBSD. Se você necessita de um meio rápido para fazer a impressora funcionar com texto puro, ou se voce tem uma impressora a qual emite fumaça quando ela imprime caracteres *backspace*, você deve definitivamente considerar `lpf`.

#### 11.4.1.1. Como os Filtros Funcionam

Como mencionado antes, o filtro é um programa executável inicializado pelo **LPD** para manipular a parte dependente de dispositivo para comunicação com a impressora.

Quando o **LPD** precisa imprimir um arquivo no *job*, ele inicializa o filtro. Ele configura a entrada padrão do filtro para o arquivo a ser impresso, a saída padrão para a impressora e a saída padrão de erro para o arquivo de log de erros (especificado na opção *lf* no arquivo `/etc/printcap`, ou `/dev/console` por padrão).

A escolha do filtro inicializado pelo **LPD** depende do argumento utilizado no arquivo `/etc/printcap` e qual foi o argumento que o usuário especificou no comando de linha [lpr\(1\)](#). Por exemplo, se o usuário digita o seguinte comando: `lpr -t`, o **LPD** irá inicializar o filtro troff, que é o argumento da opção *tf* para a impressora padrão. Se o usuário quiser imprimir um texto puro, ele deverá inicializar o filtro *if* (isso é muito comum, veja em [Output Filters](#) para maiores detalhes).

Existem tres tipos de filtros que você pode especificar no arquivo `/etc/printcap`:

- O *filtro de texto*, erradamente chamado de *filtro de entrada* na documentação do **LPD**, ele manipula a impressão de texto puro, pense nele como um filtro padrão para o **LPD**, ele espera que todas as impressoras, por padrão, estejam habilitadas a imprimir texto puro, e ela *tabs*, em um ambiente onde tem estatísticas do uso da impressora, o filtro de texto também deve gerar a estatística sobre o que foi impresso, geralmente isto é feito contando o número de linhas impressas e comparando-as com o número de linhas que a impressora suporta. O filtro de texto é inicializado com a seguinte lista de argumentos:

```
nome-do-filtro [-c] -wlargura -lcomprimento -iindent -n login -h host
arquivo-estatistica
```

onde

-c

É usado quando o *job* é enviado com o comando `lpr -l`

*largura*

É o valor da opção *pw* (largura da página) especificado no arquivo `/etc/printcap`, O padrão é 132

*comprimento*

É o valor da opção *pl* (comprimento da página), o padrão é 66

*indent*

É a quantidade de *identação* fornecido pelo comando `lpr -i`, o padrão é 0

*login*

É o nome da conta de usuário para pode imprimir o arquivo

*host*

É o nome do computador para onde a impressão será enviada.

*arquivo\_estatistica*

É o nome do arquivo onde será armazenada a estatística da impressão, este valor é atribuído à opção *af*.

- Um *filtro de conversão* converte um formato específico do arquivo em outro que a impressora possa compreender e transformar em páginas impressas. Por exemplo, o formato ditroff typesetting não pode ser impresso diretamente, mas você pode instalar um filtro conversor para o formato que a impressora possa imprimir corretamente. A seção [Filtros Conversores](#) mostra como fazê-los.

Filtros de conversão que também necessite de estatísticas, você precisa habilitar a estatística na impressora. Filtros de conversão são inicializados com os seguintes argumentos:

```
nome-do-filtro -x largura do pixel -y altura do pixel -n login -h host arquivo-estatística
```

Onde *largura do pixel* é o valor da opção *px* (valor padrão é 0) e *altura do pixel* é o valor da opção *py* (o valor padrão é 0).

- O *filtro de saída* é usado somente se não existir um filtro de texto, ou se o cabeçalho das páginas estiver habilitado. Em minha experiência, filtros de saída são raramente utilizados. A seção [Filtros de Saída](#) mostra mais detalhes. Existem somente dois argumentos para os filtros de saída, são eles:

```
nome do filtro -wlargura -lcomprimento
```

os quais são idênticos aos argumentos dos filtros de texto *-w* e *-l*.

Os filtros devem também *sair* com os seguintes status:

exit 0

Se o filtro obteve sucesso ao imprimir o arquivo.

exit 1

Se o filtro falhou ao imprimir o arquivo, mas aguarda o **LPD** para tentar imprimir outra vez. **LPD** irá reiniciar o filtro se ele sair com esse status.

exit 2

Se o filtro falhar na impressão do arquivo e não irá aguardar o **LPD** para tentar de novo. O **LPD** irá jogar fora o arquivo.

O filtro de texto que vem com a versão do FreeBSD, `/usr/libexec/lpr/lpf`, tira vantagens dos argumentos de largura e comprimento para determinar quando enviar um caractere de alimentação de folha (*form feed*) e como gerar estatísticas do uso da impressora. Ele usa o *login*, *host*, e argumentos do arquivo para gerar as estatísticas.

Se você deseja comprar filtros, veja se são compatíveis com o LPD. Se forem, eles devem suportar a lista de argumentos descrita acima. Se você planeja escrever seu próprio filtro para uso geral, eles também devem suportar a lista de argumentos e deve utilizar os códigos descritos acima.

#### 11.4.1.2. Acomodando Texto Puro em Impressoras PostScript®

Se só você é o usuário do seu computador sua impressora suporta PostScript (ou outra linguagem), e você prometeu nunca enviar texto puro para a sua impressora e nunca

usará as vantagens que vários programas que gostariam de enviar texto puro para a sua impressora, então você não precisa se preocupar com esta seção.

Mas, se você gostaria de enviar tanto PostScript e texto puro para a sua impressora, então você está convidado a incrementar a configuração da impressora. Para fazer isso, nós temos que ter um filtro para detectar se o texto é texto puro ou PostScript. Todas as impressões em PostScript devem iniciar com `%!` (para outras linguagens de impressora, veja a documentação da sua impressora). Se estes dois primeiros caracteres aparecerem no arquivo de impressão nós temos PostScript, e podemos passar o resto da impressão diretamente para a impressora. Se não temos os dois primeiros caracteres no arquivo de impressão, então o filtro deve convertê-lo em PostScript e imprimir o resultado.

Como nós faremos isso:

Se você possui uma impressora serial, o melhor caminho é instalar o `lprps`. O `lprps` é um filtro de impressora PostScript, o qual habilita a comunicação bi-direcional com a impressora. Ele atualiza o arquivo de status da impressora com informação recebida da impressora, então os usuários e administradores podem ver exatamente o estado da impressora (tais como ```toner baixo"` ou ```paper jam"`). mas o mais importante, é incluir um programa chamado de `psif`, o qual detecta quando um *job* é em texto puro e então executa o comando `textps` (outro programa que vem com o `lprps`) para convertê-lo em PostScript. Ele pode usar `lprps` para enviar um *job* para a impressora.

O `lprps` é parte da coleção de programas do FreeBSD (veja [A Coleção de Programas](#)). Você pode procurar, construir e instalar, por você mesmo. Depois da instalação do `lprps`, especifique o diretório para o programa `psif` que faz parte do `lprps`. Se você instalou o `lprps` através dos *ports*, use o seguinte na entrada referente a impressora serial PostScript configurada no arquivo `/etc/printcap`:

```
:if=/usr/local/libexec/psif:
```

Você também deve especificar a opção `rw`; que informa ao **LPD** para abrir a comunicação com a impressora no modo leitura-escrita.

Se você tem uma impressora PostScript (e não utiliza o modo bi-direcional, como necessita o `lprps`), você pode utilizar o seguinte script como um filtro texto:

```

#!/bin/sh
#
psif - Print PostScript or plain text on a PostScript printer
Script version; NOT the version that comes with lprps
Installed in /usr/local/libexec/psif
#

IFS="" read -r first_line
first_two_chars=`expr "$first_line" : '\(..\)'`

if ["$first_two_chars" = "%!"]; then
 #
 # PostScript job, print it.
 #
 echo "$first_line" && cat && printf "\004" && exit 0
 exit 2
else
 #
 # Plain text, convert it, then print it.
 #
 (echo "$first_line"; cat) | /usr/local/bin/textps && printf
"\004" && exit 0
 exit 2
fi

```

No script acima, `textps` é um programa que instalamos separadamente para converter texto puro em PostScript. Você pode usar qualquer conversor de texto para PostScript que você desejar. A coleção de programas do FreeBSD (*ports*, veja em [Coleção de Programas](#)) inclui um programa conversor de texto para PostScript com facilidades chamado *a2ps* que você deveria considerar.

### 11.4.1.3. Simulando PostScript em Impressoras Não PostScript

PostScript é *de facto* padrão para impressão de alta qualidade. PostScript é entretanto, um padrão *caro*. Ficamos gratos à Aladdin Enterprises que possui um programa chamado **Ghostscript** que pode trabalhar com a maioria dos arquivos PostScript e pode mostrar suas páginas em uma variedade de dispositivos, incluindo um grande número de impressoras não PostScript. Com a instalação do Ghostscript e usando filtros especiais de texto para a sua impressora, você pode fazer uma impressora não PostScript funcionar como uma impressora PostScript real.

Ghostscript está nos *ports* do FreeBSD, se você quiser instalar a partir de lá. Você pode buscar, compilar e instalar muito facilmente.

Para simular PostScript, nós temos um detector de filtro de texto para verificar se estamos imprimindo em um arquivo PostScript ou texto puro. Se for o caso de texto puro, então o filtro deverá repassar as informações diretamente para a impressora, caso contrário, ele deverá utilizar o Ghostscript antes para converter PostScript para o formato reconhecido pela a impressora.

Aqui tem um exemplo: o *script* a seguir é um filtro texto para uma impressora da Hewlett Packard, a Deskjet 500. Para outras impressoras, você deve substituir o argumento da opção `-sDEVICE` do comando `gs` (Ghostscript). Digitando `gs -h` você terá a lista dos dispositivos suportados pela versão instalada do Ghostscript.

```

#!/bin/sh
#
ifhp - Print Ghostscript-simulated PostScript on a DeskJet 500
Installed in /usr/local/libexec/ifhp

#
Treat LF as CR+LF:
#
printf "\033&k2G" || exit 2

#
Read first two characters of the file
#
IFS="" read -r first_line
first_two_chars=`expr "$first_line" : '\(..\)'`

if ["$first_two_chars" = "%!"]; then
 #
 # It is PostScript; use Ghostscript to scan-convert and print it.
 #
 # Note that PostScript files are actually interpreted programs,
 # and those programs are allowed to write to stdout, which will
 # mess up the printed output. So, we redirect stdout to stderr
 # and then make descriptor 3 go to stdout, and have Ghostscript
 # write its output there. Exercise for the clever reader:
 # capture the stderr output from Ghostscript and mail it back to
 # the user originating the print job.
 #
 exec 3>&1 1>&2
 /usr/local/bin/gs -dSAFER -dNOPAUSE -q -sDEVICE=djet500 \
 -sOutputFile=/dev/fd/3 - && exit 0
else
 #
 # Plain text or HP/PCL, so just print it directly; print a form
 # feed
 # at the end to eject the last page.
 #
 echo "$first_line" && cat && printf "\033&l0H" &&
 exit 0
fi

exit 2

```

Finalmente, você deve notificar o **LPD** do filtro através da opção *if*

```
:if=/usr/local/libexec/ifhp:
```

Feito isso, você poderá digitar `lpr texto.puro` e `lpr qualquer-coisa.ps` e ambos deverão ser impressos corretamente na impressora.

#### 11.4.1.4. Filtros Conversores

Após ter completado a configuração descrita na seção [Configuração Simples da Impressora](#), a primeira coisa que você provavelmente pensa é instalar filtros conversores para seus formatos favoritos de arquivo.



#### 11.4.1.4.1. Por que Instalar Filtros Conversores?

Filtros conversores podem imprimir uma variedade de tipos de arquivo facilmente. Por exemplo, suponha que tenhamos que fazer um grande trabalho com o sistema **TeX**, e nós temos uma impressora PostScript. Cada vez que geramos um arquivo DVI através do **TeX**, nós não podemos imprimi-lo diretamente, pois precisamos convertê-lo em PostScript. A sequência de comandos será algo semelhante a:

```
% dvips seaweed-analysis.dvi
% lpr seaweed-analysis.ps
```

Instalando um filtro para converter os arquivos DVI em PostScript, poderíamos pular um passo a cada vez que tivéssemos que imprimir. O **LPD** poderá realizar esse passo para nós.

```
% lpr -d seaweed-analysis.dvi
```

Nós informamos o **LPD** para converter o arquivo DVI através da opção *-d*. A seção [Opções de Formatação e Conversão](#) mostra uma lista de opções de conversão

Para cada opção de conversão que você deseja que sua impressora suporte, deve instalar um *Filtro Conversor* e especificar o diretório no arquivo `/etc/printcap`. Um filtro conversor é um filtro texto como foi definido na seção Configuração Simples de Impressora (veja a seção [Instalando um Filtro de Texto](#)) Exceto pelo fato imprimir arquivos de texto puro, os filtros convertem os arquivos em um formato que a impressora pode entender.

#### 11.4.1.4.2. Qual o Filtro Conversor Devo Instalar?

Você deve instalar o filtro conversor que você espera utilizar. Se você imprime uma grande quantidade de dados no formato DVI, então você deve ter um filtro para convertê-lo. Se você deseja imprimir saídas através de troff, então provavelmente você terá um filtro troff.

A seguinte tabela sumariza os filtros que o **LPD** pode trabalhar, e suas opções estão no arquivo `/etc/printcap`, e como devem ser chamados com o comando `lpr`:

Tipo de Arquivo	Opção no <code>/etc/printcap</code>	Opções do <code>lpr</code>
cifplot	<i>cf</i>	<i>-c</i>
DVI	<i>df</i>	<i>-d</i>
plot	<i>gf</i>	<i>-g</i>
ditroff	<i>nf</i>	<i>-n</i>
texto em FORTRAN	<i>rf</i>	<i>-f</i>

Tipo de Arquivo	Opção no <code>/etc/printcap</code>	Opções do <code>lpr</code>
troff	<i>tf</i>	<i>-f</i>
raster	<i>vf</i>	<i>-v</i>
Texto Puro	<i>if</i>	none, <i>-p</i> , or <i>-l</i>

No nosso exemplo, usamos o comando `lpr -d` que significa dizer que precisamos da opção *df* dentro do arquivo `/etc/printcap`.

Apesar de algumas opiniões contrárias, formatos como o FORTRAN e plot são provavelmente obsoletos. No seu sítio, você pode dar novas opções de formatação para estes ou qualquer outra opção apenas instalando novos filtros ajustados para o seu uso. Por exemplo, suponhamos que seja necessário imprimir diretamente arquivos Printerleaf (arquivos vindos do programa de publicação Interleaf), mas você nunca irá imprimir arquivos de plot. Você poderá instalar um filtro para converter esse formato utilizando a opção *gf* e educar os seus usuários a utilizarem o comando `lpr -g` quando forem imprimir arquivos do tipo "Printerleaf".

#### 11.4.1.4.3. Instalando Filtros Conversores

Uma vez que os filtros de conversão são programas instalados por fora da instalação padrão do FreeBSD, eles devem, provavelmente, ficar no diretório `/usr/local`. O diretório `/usr/local/libexec` é a localização mais indicada, pois são programas especializados que somente o **LPD** irá usar; usuários regulares do sistema, não deverão ter necessidade de executá-los.

Para habilitar um filtro conversor, especifique o diretório no qual ele se encontra, na opção específica do arquivo `/etc/printcap`.

No nosso exemplo iremos adicionar um filtro conversor DVI na impressora chamada *bamboo*. Aqui um exemplo de um arquivo `/etc/printcap`, com a nova opção *df* para a impressora *bamboo*.

```
#
/etc/printcap for host rose - added df filter for bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
 :sh:sd=/var/spool/lpd/rattan:\
 :lp=/dev/lpt0:\
 :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
 :sh:sd=/var/spool/lpd/bamboo:\
 :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
 :if=/usr/local/libexec/psif:\
 :df=/usr/local/libexec/psdf:
```

O filtro DVI é um script para o shell chamado de `/usr/local/libexec/psdf`. aqui está o script

```
#!/bin/sh
#
psdf - DVI to PostScript printer filter
Installed in /usr/local/libexec/psdf
#
Invoked by lpd when user runs lpr -d
#
exec /usr/local/bin/dvips -f | /usr/local/libexec/lprps "$@"
```

Esse script executa `dvips` no modo de filtro (o argumento `-f`) na entrada padrão, na qual o job será impresso. Ele inicia o filtro PostScript da impressora `lprps` (veja a seção [Acomodando Trabalhos de Impressão em Texto Puro em Impressoras PostScript](#)) com os argumentos do **LPD** passado pelo script. O comando `lprps` será usado como argumento para a contabilidade de páginas impressas.

#### 11.4.1.4.4. Mais Exemplos de Filtros Conversores

Uma vez que não existe um conjunto de passos para a instalação de filtros conversores, providenciaremos mais exempemplos. Use-os como guia para fazer os seus próprios filtros. Use-os diretamente se forem apropriados.

Esse exemplo de script é um filtro conversor *raster* (com arquivos GIF atualmente) para as impressoras Hewlett Packard LaserJet III-Si:

```
#!/bin/sh
#
hpvf - Convert GIF files into HP/PCL, then print
Installed in /usr/local/libexec/hpvf

PATH=/usr/X11R6/bin:$PATH; export PATH
giftopnm | ppmtopgm | pgmtopbm | pbmtolj -resolution 300 \
 && exit 0 \
 || exit 2
```

Ele converte um arquivo GIF em um formato *anymap* portátil, convertendo em um mapa de cinza portátil, convertendo em um formato bitmap portátil, e então convertendo em dados compatível com PCL.

Um exemplo de arquivo `/etc/printcap` com uma entrada para uma impressora usando o filtro acima.

```
#
/etc/printcap for host orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
 :lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak:mx#0:\
 :if=/usr/local/libexec/hpif:\
 :vf=/usr/local/libexec/hpvf:
```

O próximo script converte dados troff vindos de um sistema groff para uma impressora PostScript chamada *bamboo*:

```
#!/bin/sh
#
pstf - Convert groff's troff data into PS, then print.
```

```
Installed in /usr/local/libexec/pstf
#
exec grops | /usr/local/libexec/lprps "$@"
```

O script acima, faz uso novamente do comando `lprps` para manipular a comunicação com a impressora. Se a impressora estiver na porta paralela, nós devemos usar este script:

```
#!/bin/sh
#
pstf - Convert groff's troff data into PS, then print.
Installed in /usr/local/libexec/pstf
#
exec grops
```

Aqui está a entrada que precisamos adicionar no arquivo `/etc/printcap` para habilitar o filtro:

```
:tf=/usr/local/libexec/pstf:
```

Aqui está um filtro para texto em FORTRAN para qualquer impressora que imprima diretamente em texto puro. Nós iremos instalar isso para a impressora chamada *teak*:

```
#!/bin/sh
#
hprf - FORTRAN text filter for LaserJet 3si:
Installed in /usr/local/libexec/hprf
#

printf "\033&k2G" && fpr && printf "\033&l0H" &&
exit 0
exit 2
```

E nós adicionamos esta linha no arquivo `/etc/printcap` para a impressora *teak* para habilitar esse filtro:

```
:rf=/usr/local/libexec/hprf:
```

Aqui é o fim desse exemplo complexo. Nós iremos adicionar o filtro DVI para a impressora LaserJet *teak* introduzido recentemente. Primeiro, a parte mais fácil; atualize o arquivo `/etc/printcap` com a localização do filtro DVI:

```
:df=/usr/local/libexec/hpdf:
```

Agora, a parte mais difícil: fazendo o filtro. Para tanto, nós iremos necessitar de um programa conversor de DVI para LaserJet/PCL. A coleção de programas do FreeBSD (veja em [A Coleção de Programas](#)) possui um: `dvi2xx`, é o nome do pacote. Instalando esse pacote, teremos o programa necessário, `dvilj2p`, o qual converte DVI em arquivos LaserJet IIp, LaserJet III, e LaserJet 2000 compatíveis.

O comando `dvilj2p` deixa o filtro `hpdf` pouco complexo uma vez que o comando `dvilj2p` não pode ler da entrada padrão. Ele necessita trabalhar com o nome do arquivo. Isso é ruim, o nome do arquivo deve ter no final `.dvi` então usando o `/dev/fd/0` para a entrada padrão seria problemático. Nós podemos contornar o

problema linkando (simbolicamente) um nome temporário de arquivo (um que terminaria em `.dvi`) para o `/dev/fd/0`, forçando assim o `dvi1j2p` ler da entrada padrão.

Outro ponto importante é que não podemos utilizar o diretório `/tmp` para um link temporário. Links simbólicos são propriedade do usuário e grupo `bin`. O filtro executa como o usuário `daemon`. E o filtro não conseguirá criar o link, mas também ele não estará apto quando de removê-lo quando terminar uma vez que o link pertence a um usuário diferente.

Ao invés disso o filtro criará um link simbólico no diretório de trabalho corrente, que é o diretório de spooling (especificado na opção `sd` no arquivo `/etc/printcap`). Este é um lugar perfeito para os filtros realizarem o seu trabalho, especialmente porque na maioria das vezes possui mais espaço do que o diretório `/tmp`.

Aqui, finalmente, o filtro:

```
#!/bin/sh
#
hpdf - Print DVI data on HP/PCL printer
Installed in /usr/local/libexec/hpdf

PATH=/usr/local/bin:$PATH; export PATH

#
Define a function to clean up our temporary files. These exist
in the current directory, which will be the spooling directory
for the printer.
#
cleanup() {
 rm -f hpdf$.dvi
}

#
Define a function to handle fatal errors: print the given message
and exit 2. Exiting with 2 tells LPD to do not try to reprint the
job.
#
fatal() {
 echo "$@" 1>&2
 cleanup
 exit 2
}

#
If user removes the job, LPD will send SIGINT, so trap SIGINT
(and a few other signals) to clean up after ourselves.
#
trap cleanup 1 2 15

#
Make sure we are not colliding with any existing files.
#
cleanup

#
Link the DVI input file to standard input (the file to print).
```

```

#
ln -s /dev/fd/0 hpdf$$.dvi || fatal "Cannot symlink /dev/fd/0"

#
Make LF = CR+LF
#
printf "\033&k2G" || fatal "Cannot initialize printer"

#
Convert and print. Return value from dvi2p does not seem to be
reliable, so we ignore it.
#
dvi2p -Ml -q -e- dfhp$$.dvi

#
Clean up and exit
#
cleanup
exit 0

```

#### 11.4.1.4.5. Conversão Automática: Uma Alternativa aos Filtros de Conversão

Todos esses filtros de conversão completam o seu ambiente de impressão, mas a um custo de forçar seu usuário a especificar (no comando de linha [lpr\(1\)](#)) qual ele irá usar. Se seus usuários não estão familiarizados, ou não tiverem costume com literatura computacional, eles acharão isso muito chato. O que é ruim, entretanto, é a escolha de uma opção errada, que poderá gerar muitas páginas de lixo.

De preferência quando for instalar um filtro de conversão, você poderá tentar deixar como padrão um filtro texto, que detecte o tipo de arquivo e execute o filtro adequado para conversão. Ferramentas como `file` poderão ajudá-lo a resolver isto. Claro que você terá dificuldades em diferenciar entre *alguns* tipos de arquivos e é claro que você estará providenciando o filtro correto para convertê-los.

A Coleção de Programas do FreeBSD tem um filtro de texto capaz de realizar automaticamente a conversão definida chamada `apsfilter`. Ele também pode detectar texto puro, PostScript e DVI, executar as conversões necessárias, e a seguir imprimir o resultado.

#### 11.4.1.5. Filtros de Saída

O sistema de spooling **LPD** suporta um outro tipo de filtro que ainda não foi explorado: o filtro de saída. Um filtro de saída tem apenas o objetivo de imprimir texto puro, como outros filtros de texto, mas com muitas simplificações. Se você estiver utilizando um filtro de saída mas não um filtro texto, então:

- O **LPD** inicia o filtro de saída apenas uma única vez para o trabalho de impressão ao invés de uma vez para cada arquivo no trabalho de impressão.
- O sistema **LPD** não faz nenhuma previsão para identificar o início ou o final de arquivos dentro do trabalho de impressão para o filtro de saída.
- O sistema **LPD** não repassa o nome do usuário ou a máquina para o filtro, então não consegue fazer a contabilização do uso. De fato, ele só utiliza dois argumentos:

nome do filtro -*wlargura* -*lcomprimento*

Onde *largura* é retirado da opção *pw* e o *comprimento* é retirado da opção *pl* referente a impressora utilizada

Não fique seduzido pela facilidade do uso dos filtros de saída. Se você gostaria de ter cada arquivo do trabalho de impressão iniciando em diferente páginas um filtro de saída *não iria funcionar*. Use um filtro de texto (também conhecido como filtro de entrada); veja a seção [Instalando um Filtro Texto](#). Além disso, um filtro de saída é atualmente *mais complexo* porque ele examina um *stream* de bytes inicialmente enviado para ele com um *flag* especial e deve ser enviado sinais para ele próprio em benefício do sistema **LPD**.

Entretanto, um filtro de saída é *necessario* se você quiser cabeçalho de pá e precisa enviar sequências de caracteres escape ou outro conjunto de caracteres inicialização para abilitar o cabeçalho das páginas. (mas isso é também *fútil* se você deseja carregar cabeçalhos de página para a contabilidade do usuário uma vez que o sistema **LPD** não fornece nenhuma informação do usuário ou do computador para o filtro de saída).

Para uma única impressora, o sistema **LPD** permite tanto filtros de saída e de texto ou de outro tipo. Na maioria dos casos o sistema **LPD** irá iniciar o filtro de saída para imprimir a página de cabeçalho ( veja a seção [Páginas de Cabeçalho](#)) somente. O sistema **LPD** então agurada o filtro de saída *terminar* pelo envio de uma sequência de dois bytes para o filtro: os caracteres ASCII 031 seguido pelo 001. Quando o filtro ve esses dois bytes (031, 001), ele deve parar com um envio do sinal *SIGSTOP* para si mesmo. Quando o sistema **LPD** termina a execução de outros filtros, ele irá reinicializar o filtro de saída pelo envio do sinal *SIGCONT* para ele.

Se existir um filtro de saída mas *não* um filtro texto e o **LPD** estiver trabalhando em um trabalho de impressão de texto puro, o **LPD** utiliza o filtro de saída para realizar a tarefa de impressão. Como dito antes, o filtro de saída irá imprimir cada arquivo do trabalho de impressão na sequência sem o intervalo de alimentação de folha ou outro avanço de papel, e isso você provavelmente *não* deseja. Na maioria dos casos você necessitaria de um filtro de texto.

O programa `lpf`, o qual nos introduzimos recentemente como um filtro texto, pode também ser executado como um filtro de saída. Se você necessitar de um filtro de saída rápido e sem muita qualidade, mas não quer ter o trabalho de escrever um detector de byte e um código para enviar sinais, tente o `lpf`. Você também pode colocar o `lpf` dentro de um script em shell para manipular os códigos de inicialização que a impressora pode necessitar.

#### 11.4.1.6. `lpf`: Um Filtro de Texto

O programa `/usr/libexec/lpr/lpf` que vem com a distribuição do FreeBSD é um filtro de texto ( filtro de entrada ) e pode identar a saída ( o trabalho de impressão enviado com o comando `lpr -i`), permitindo que caracteres literais possam ser enviados (trabalho de impressão enviado com o comando `lpr -l`), ajustar a posição de impressão para caracteres de retorno (backspace) e tabulações no trabalho de impressão,

e fazer a contabilidade das páginas impressas. Também pode agir como um filtro de saída.

O comando `lpf` é adequado para a maioria dos ambientes de impressão. E ainda que não possua capacidade para enviar sequências de inicialização para as impressoras, isto é fácil ser escrito em script de shell para fazer as inicializações necessárias e então executar o `lpf`.

Para que o `lpf` faça a estatística corretamente, ele necessita dos valores preenchidos corretamente nas opções `pw` e `pl` no arquivo `/etc/printcap`. Ele utiliza esses valores para determinar como o texto poderá caber na página que quantas páginas estarão no trabalho de impressão do usuário. Para mais informações, veja a seção [Estatísticas do Uso da Impressora](#).

## 11.4.2. Páginas de Cabeçalhos

Se você tiver *muitos* usuários, todos eles usando várias impressoras, você provavelmente deverá considerar as *páginas de cabeçalho* como um mal necessário.

Páginas de Cabeçalho, também conhecida como *banner* ou *burst pages* identifica de quem é o trabalho de impressão depois dele ser impresso. Ela é geralmente impressa grande, com letras em negrito, e as vezes com bordas decorativas, para que em um pilha de impressão eles possam ser identificados e entregues ao usuário correto. As páginas de cabeçalho são impressas uma por trabalho de impressão e não uma por arquivo impresso.

O sistema **LPD** pode providenciar automaticamente páginas de cabeçalho para suas saídas se sua impressora puder imprimir texto puro. Se você tiver uma impressora PostScript, você necessitava de um programa externo para gerar essas páginas; veja a seção [Páginas de Cabeçalho em Impressoras PostScript](#).

### 11.4.2.1. Habilitando Páginas de Cabeçalho

Na seção [Configuração Simples da Impressora](#), nos desligamos a impressão das páginas de cabeçalho, através da opção `sh` ( que significa ``suppres header") no arquivo `/etc/printcap`. Para habilitar a impressão, basta a remoção da opção `sh`.

Isso parece tão fácil, correto?

E você esta certo. Você *deve* ter providenciado um filtro de saída para enviar os dados de inicialização para a impressora. Aqui um está um exemplo de filtro de saída para uma impressora Hewlett Packard PCL-compatível:

```
#!/bin/sh
#
hpof - Output filter for Hewlett Packard PCL-compatible printers
Installed in /usr/local/libexec/hpof

printf "\033&k2G" || exit 2
exec /usr/libexec/lpr/lpf
```



especificando odiretório para o filtro de saída na opção *of*. veja na seção [Filtros de Saída](#) para maiores informações.

Aqui um exemplo do arquivo `/etc/printcap` para a impressora *teak* que foi introduzida recentemente; nos habilitamos a página de cabeçalho e adicionamos o filtro de saída descrito acima:

```
#
/etc/printcap for host orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
 :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:\
 :if=/usr/local/libexec/hpif:\
 :vf=/usr/local/libexec/hpvf:\
 :of=/usr/local/libexec/hpof:
```

Agora, quando os usuários imprimirem os seus trabalhos para a impressora *teak*, eles terão uma página de cabeçalho para cada job impresso. Se os usuários querem perder tempo procurando os seus trabalhos de impressora, eles podem suprimir essas páginas com o comando `lpr -h`; veja a seção [Opções de Páginas de Cabeçalho](#) para outras opções veja [lpr\(1\)](#).

**Nota:** O LPD imprime o caracter de alimentação de página (form feed) logo apos a página de cabeçalho. Se sua impressora utiliza um caracter diferente ou sequencia de caracteres para ejetar uma página, especifique-o na opção *ff* no arquivo `/etc/printcap`.

### 11.4.2.2. Controlando Páginas de Cabeçalho

Uma vez habilitada a página de cabeçalho, o LPD irá produzir um *longo cabeçalho*, uma página completa com letras grandes identificando o usuário, máquina, e o trabalho de impressão. Aqui está um exemplo (kelly imprimiu um trabalho de impressão i entitulado de outline a partir da máquina rose):

```

k ll ll
k l l
k l l
k k eeee l l y y
k k e e l l y y
k k eeeeee l l y y
kk k e l l y y
k k e e l l y yy
k k eeee ll ll yyy y
k k eeee ll ll yyy y
 y
 Y Y
 YYY Y

 ll
 l i
 t l
 t l
o o u u ttttt l ii n nnn eeee
o o u u t l i nn n e e
o o u u t l i n n e
o o u uu t t l i n n e e
```

oooo      uuuu      tt      lll      iii      n      n      eeee

```
r rrr oooo sss eeee
rr r o o s s e e
r o o s eeeeee
r o o s e
r o o s s e e
r oooo sss eeee
```

Job: outline  
Date: Sun Sep 17

11:04:58 1995

O **LPD** adiciona uma alimentação de página logo após o texto ser impresso, então inicia em uma nova página o trabalho de impressão (a menos que se tenha definido a opção *sf* (suprimir a alimentação de página) na impressora destinada no arquivo `/etc/printcap`).

Se você preferir, o **LPD** pode fazer uma *página pequena*; especificando a opção *sb* (short banner) no arquivo `/etc/printcap`. A página de cabeçalho irá parecer como esta:

```
rose:kelly Job: outline Date: Sun Sep 17 11:07:51 1995
```

Também por padrão, o **LPD** imprime a página de cabeçalho primeiro, e então o trabalho de impressão. Para reverter isso, especifique a opção *hl* (header last) no arquivo `/etc/printcap`.

### 11.4.2.3. Estatísticas para Páginas de Cabeçalho

Usando o **LPD** para imprimir as páginas de cabeçalho, leva-nos a um paradigma, nos devemos contar as páginas de cabeçalho para fazer as estatísticas, ou devemos *deix-la de fora*.

Por que?

Por que o filtro de saída é somente um programa externo que terá controle quando a página de cabeçalho for impressa e não pode diferenciar isto do total para que possa fazer a estatística corretamente, e isso não é fornecido por qualquer *usuário ou máquina* ou qualquer arquivo de estatística, então ele não tem ideia para quem contabilizar o uso da impressora. E também não é justo "adicionar uma página" para o filtro texto ou

qualquer outro filtro de conversão ( os quais têm informações sobre os usuários e máquinas) uma vez que os usuários podem suprimir as páginas de cabeçalho com o comando `lpr -h`. À eles então seriam atribuídos os custos ( estatísticas ) de páginas de cabeçalho que efetivamente não imprimiram. Basicamente, o `lpr -h` irá ser uma opção preferida no âmbito dos usuários, mas você não deverá incentivar o seu uso.

Isso *não é o suficiente* para cada filtro gerar sua própria página de cabeçalho ( entretanto estará apto a contabiliza-las). Se os usuários quiserem a opção de supressão de páginas de cabeçalho com o comando `lpr -h`, eles ainda terão e serão contabilizado uma vez que o **LPD** não tem conhecimento da `-h` para nenhum filtro.

Então, qual é a sua opção?

Você pode:

- Aceitando o paradigma do **LPD** e liberando as páginas de cabeçalho.
- Instalando alternativas ao **LPD**, tais como: **LPRng**. Seção [Alternativas Ao Spooler Padrão](#) diz mais sobre os programas que podem substituir o **LPD**.
- Escrevendo um filtro de saída *inteligente*. Normalmente um filtro de saída não precisa de muita coisa a além de inicializar a impressora ou fazer algumas conversões de caracteres. Ele terá que se adaptar para páginas de cabeçalho ou para trabalhos de impressão em texto puro ( quando ele não for um filtro de texto (entrada)). Mas, se existir um filtro de texto para trabalhos de impressão em texto puro, então **LPD** irá iniciar o filtro de saída somente para a página de cabeçalho. E o filtro de saída poderá analisar a página de cabeçalho que o **LPD** gerou para determinar qual usuário e máquina para poder contabiliza-la. O único problema com esse método é que o filtro de saída continua sem ter conhecimento do arquivo de estatística ( isso não é passado pela opção `af`), mas se você tem o conhecimento do arquivo de estatística, voce poderá coloca-lo direto no código do filtro de saída. Para facilitar o passo de análise, use a opção `sh` (página curta) do arquivo `/etc/printcap`. Então depois de tudo feito, os usuários irão apreciar o quão generoso o administrador do sistema é, liberando das estatísticas de utilização a página de cabeçalho.

#### 11.4.2.4. Páginas de Cabeçalho em Impressoras PostScript

Como descrito acima, o **LPD** pode gerar as páginas de cabeçalho em texto puro, funcional para a maioria das impressoras. Claro que as impressoras PostScript não podem imprimir diretamente texto puro, então a vantagem do **LPD** poder imprimir as páginas de cabeçalho torna-se imprestável--.

Uma maneira óbvia de ter páginas de cabeçalho é gerar através de cada filtro de conversão e de filtros de texto, as páginas de cabeçalho. Os filtros devem usar os argumentos de nome de usuário e da máquina, para gerar uma página de cabeçalho otimizada. a desvantagem deste método é que o usuário sempre terá a página de cabeçalho mesmo que ele utilize o comando `lpr -h` para imprimir seus trabalhos de impressões.

Iremos explorar este método. O *script* abaixo manipula três argumentos ( nome do usuário (login), nome da máquina, e o nome do trabalho de impressão) e faz uma página simples de cabeçalho em PostScript:

```
#!/bin/sh
#
make-ps-header - make a PostScript header page on stdout
Installed in /usr/local/libexec/make-ps-header
#
#
These are PostScript units (72 to the inch). Modify for A4 or
whatever size paper you are using:
#
page_width=612
page_height=792
border=72
#
Check arguments
#
if [$# -ne 3]; then
 echo "Usage: `basename $0` <user> <host> <job>" 1>&2
 exit 1
fi
#
Save these, mostly for readability in the PostScript, below.
#
user=$1
host=$2
job=$3
date=`date`
#
Send the PostScript code to stdout.
#
exec cat <<EOF
%!PS

%
% Make sure we do not interfere with user's job that will follow
%
save

%
% Make a thick, unpleasant border around the edge of the paper.
%
$border $border moveto
$page_width $border 2 mul sub 0 rlineto
0 $page_height $border 2 mul sub rlineto
currentscreen 3 -1 roll pop 100 3 1 roll setscreen
$border 2 mul $page_width sub 0 rlineto closepath
0.8 setgray 10 setlinewidth stroke 0 setgray

%
% Display user's login name, nice and large and prominent
%
/Helvetica-Bold findfont 64 scalefont setfont
```

```

$page_width ($user) stringwidth pop sub 2 div $page_height 200 sub
moveto
($user) show

%
% Now show the boring particulars
%
/Helvetica findfont 14 scalefont setfont
/y 200 def
[(Job:) (Host:) (Date:)] {
200 y moveto show /y y 18 sub def }
forall

/Helvetica-Bold findfont 14 scalefont setfont
/y 200 def
[($job) ($host) ($date)] {
 270 y moveto show /y y 18 sub def
} forall

%
% That is it
%
restore
showpage
EOF

```

Agora, cada chamada aos filtros de conversão e aos filtros de texto, irá ser gerada uma página de cabeçalho e então será impresso o trabalho de impressão do usuário. Segue um exemplo de um filtro de conversão de DVI, mostrado anteriormente neste documento, modificado para poder imprimir a página de cabeçalho.

```

#!/bin/sh
#
psdf - DVI to PostScript printer filter
Installed in /usr/local/libexec/psdf
#
Invoked by lpd when user runs lpr -d
#

orig_args="$@"

fail() {
 echo "$@" 1>&2
 exit 2
}

while getopts "x:y:n:h:" option; do
 case $option in
 x|y) ;; # Ignore
 n) login=$OPTARG ;;
 h) host=$OPTARG ;;
 *) echo "LPD started `basename $0` wrong." 1>&2
 exit 2
 ;;
 esac
done

["$login"] || fail "No login name"
["$host"] || fail "No host name"

```

```
(/usr/local/libexec/make-ps-header $login $host "DVI File"
 /usr/local/bin/dvips -f) | eval /usr/local/libexec/lprps $orig_args
```

Observe como o filtro faz a análise da lista de argumentos para que possa determinar os nomes de máquina e de usuário. A análise para os outros filtros de conversão é idêntica. O filtro de texto possui uma sutil diferença no conjunto de argumentos passados ( veja a seção [Como os filtros funcionam](#) ).

Como foi mencionado, o esquema acima é muito simples, desabilita a opção "supressão da página de cabeçalho" ( a opção `-h` ) para o comando `lpr`. Se os usuários quiserem economizar árvores ( ou dinheiro, caso seja cobrada a impressão da páginas de cabeçalho). ele não poderam faze-lo, uma vez que os próprios filtros se encarregam deste serviço.

Para permitir os usuários desligarem as páginas de cabeçalho, você deverá usar o truque mostrado na seção [Contabilizando as Páginas de Cabeçalho](#) : para escrever um filtro de saída que analise a página de cabeçalho gerada pelo LPD e produzir uma versão em PostScript. Se o usuário enviar o trabalho de impressão com o comando `lpr -h`, então o **LPD** não irá gerar a página de cabeçalho, e nada ira para o seu filtro de saída. Caso contrario, seu filtro de saída ira ler o texto que o **LPD** gerou e irá enviar a página de cabeçalho apropriada em código PostScript.

Se voce tiver uma impressora PostScript na porta serial, você pode utilizar o comando `lprps`, o qual vem com um filtro da saída, `psmf`, Note que o `psmf` não contabiliza as páginas de cabeçalho

### 11.4.3. Impressão em Rede

O FreeBSD suporta impressão via rede; enviando trabalhos de impressão para impressoras remotas. Impressão em rede geralmente nos conduz a duas coisas diferentes:

- Acessando a impressora conectada a uma máquina remota. Você instala a impressora normalmente como se fosse uma impressora serial ou paralela na máquina. Então, você configura o **LPD** para habilitar o acesso a impressora de outras máquinas na rede. Veja a seção [Impressoras Instaladas em Máquinas Remotas](#) mostra como faze-lo.
- Acessando a impressora conectada diretamente a rede. A impressora possui uma interface de rede, além da porta paralela ou serial. Como a impressora que iremos utilizar a seguir:
  - Ela deve entender o protocolo **LPD** e possa manipular filas de trabalhos de impressão de uma máquina remota. Nesse caso, Ela age como uma máquina remota comum rodando **LPD**. Seguindo os mesmos passos na seção [Instalando Impressoras em Máquinas Remotas](#) para configurar a impressora.
  - Ela deve suportar conexão de rede do tipo *stream*. Nesse caso, você conecta a impressora em uma máquina na rede que ficará responsável pelo *spooling* dos trabalhos de impressão e envia-los para a impressora. Na seção [Impressoras com Interfaces de Rede do Tipo Data Stream](#) Dá sugestões de como instalar essas impressoras.

### 11.4.3.1. Impressoras Instaladas em um Computador Remoto

O sistema **LPD** foi construído com suporte para enviar trabalhos de impressão para outras máquinas na rede, rodando **LPD** ( ou um algo compatível com ele ). Essa característica possibilita a instalação de impressoras em uma máquina que possa ser acessada por outras. Isso também funciona para impressora que possuem interfaces de rede capaz de compreender o protocolo **LPD**.

Para habilitar essa impressora remota, primeiro devemos instalar a impressora em uma máquina, que será o *servidor de Impressão*, usando os passos descritos na seção [Simples Configuração de Impressora](#) . Para configuração avançada veja a seção [Configuração Avançada de Impressora](#) . Tenha certeza que o teste de impressão funcionou corretamente para esta impressora. Tenha certeza que o *computador local* tenha autorização para acessar a *máquina remota* e o serviço do **LPD**, veja a seção [Restringindo os Trabalhos de Impressão para Máquinas Remotas](#) ).

Se estiver utilizando uma impressora com interface de rede compatível com **LPD**, então *servidor de impressão* será a própria impressora, e o *nome da impressora* será o nome que você configurou para a impressora. Veja a documentação que acompanha a sua impressora e/ou a interface de rede dela.

**Dica:** Se você está utilizando uma Hewlett Packard Laserjet então o nome da impressora fará automaticamente a conversão do LF para o CRLF para você, e então você não necessitará do filtro de texto `hpif` para isso.

Então, em outros servidores que você quiser ter acesso, deverá adicionar uma entrada no arquivo `/etc/printcap` como segue:

1. Nomeia a entrada como quiser. Por simplicidade, você usará o mesmo nome/apelido que estiver no servidor da impressora.
2. Deixe a opção `lp` em branco, faça isso explicitamente (`:lp=:`).
3. Crie um diretório de spool e especifique a sua localização com a opção `sd`. O **LPD** irá armazenar os trabalhos de impressão lá antes de entregar ao servidor de impressão
4. Coloque o nome do servidor da impressora na opção `rm`.
5. Coloque o nome da impressora que está no servidor de impressão não opção `rp`

Feito isso, você não precisará de filtros de entrada, dimensionamento de páginas, ou outra coisa qualquer no arquivo `/etc/printcap`.

Aqui está um exemplo. O servidor `rose` possui duas impressoras, *bamboo* e *rattan*. Nos iremos permitir que os usuários da máquina `orchid` possa imprimir nessas impressoras. Aqui está o arquivo `/etc/printcap` para a máquina `orchid` (veja a seção anterior [Habilitando Páginas de Cabeçalho](#) ). Ele já possui a entrada para a impressora *teak*; nos iremos adicionar a entrada para as duas impressoras no servidor `rose`:

```
#
/etc/printcap for host orchid - added (remote) printers on rose
#
#
```

```

teak is local; it is connected directly to orchid:
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
 :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:\
 :if=/usr/local/libexec/ifhp:\
 :vf=/usr/local/libexec/vfhp:\
 :of=/usr/local/libexec/ofhp:

#
rattan is connected to rose; send jobs for rattan to rose:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
 :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan:

#
bamboo is connected to rose as well:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
 :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo:

```

Então, nos apenas criamos o diretório de *spool* na máquina *orchid*:

```

mkdir -p /var/spool/lpd/rattan /var/spool/lpd/bamboo
chmod 770 /var/spool/lpd/rattan /var/spool/lpd/bamboo
chown daemon:daemon /var/spool/lpd/rattan /var/spool/lpd/bamboo

```

Agora, os usuários na máquina *orchid* podem imprimir para as impressoras *rattan* e *bamboo*. Se, por exemplo, o usuário na máquina *orchid* digitar:

```
% lpr -P bamboo -d sushi-review.dvi
```

O sistema **LPD** na máquina *orchid* irá copiar o trabalho de impressão do diretório de *spool* /var/spool/lpd/bamboo e irá notar que é uma impressão em DVI. Tão logo o servidor *rose* tiver espaço no diretório de *spool* da impressora *bamboo* os dois sistemas **LPDs** irão transferir o arquivo para o servidor *rose*. O arquivo ficará aguardando na fila do servidor *rose* até ser finalmente impresso. Ele então irá converter do formato DVI para PostScript deste que a impressora *bamboo* consiga interpretar PostScript

### 11.4.3.2. Impressoras com Interfaces de Rede *Data Stream*

Geralmente, quando você compra uma interface de rede para a sua impressora, você pode ter duas versões: uma que emula um *spooler* ( a versão mais cara), ou uma que permita que você envie dados para ela como se você estivesse utilizando uma porta paralela ou serial ( a versão mais barata). Essa seção mostra como você irá utilizar essa versão mais barata. Para a versão mais cara, veja a seção anterior [Impressoras Instaladas em uma Máquina Remota](#) .

O formato do arquivo /etc/printcap permite que você especifique qual a interface, seria ou paralela, a ser usada e ( se você estiver usando uma interface serial), qual a taxa de transferencia ( *baud rate*), Se está usando controle de fluxo, expansão de tabulação, conversão de novas linhas, e mais. Mas existe um meio de especificar a conexão com a impressora que e ouvir sobre TCP/IP ou outra porta da rede.



Para enviar dados para uma impressora na rede, você deverá desenvolver um programa de comunicação que possa ser chamado pelo filtro de texto e/ou de conversão. Aqui está um exemplo: O script `netprint` pega todos os dados da entrada padrão e envia para a impressora de rede. Nos especificamos o nome da máquina onde a impressora está conectada como o primeiro argumento e o número da porta para a conexão como o segundo argumento para o `netprint`. Note que esse suporte é de uma única direção ( Impressoras para o FreeBSD ); muitas impressoras de rede suportam comunicação bidirecional, você pode ter vantagem em utilizar essa comunicação para obter status da impressora, fazer contabilidade, etc.

```
#!/usr/bin/perl
#
netprint - Text filter for printer attached to network
Installed in /usr/local/libexec/netprint
#
$#ARGV eq 1 || die "Usage: $0 <printer-hostname> <port-number>";

$printer_host = $ARGV[0];
$printer_port = $ARGV[1];

require 'sys/socket.ph';

($ignore, $ignore, $protocol) = getprotobyname('tcp');
($ignore, $ignore, $ignore, $ignore, $address)
 = gethostbyname($printer_host);

$sockaddr = pack('S n a4 x8', &AF_INET, $printer_port, $address);

socket(PRINTER, &PF_INET, &SOCK_STREAM, $protocol)
 || die "Can't create TCP/IP stream socket: $!";
connect(PRINTER, $sockaddr) || die "Can't contact $printer_host: $!";
while (<STDIN>) { print PRINTER; }
exit 0;
```

Nos podemos utilizar este script em vários filtros. Supondo que temos uma impressora Diabolo 750-N conectada à rede. A impressora aceita dados para impressão na porta de número 5100. O nome da máquina é `scrivener`. Aqui está um filtro de texto para a impressora.

```
#!/bin/sh
#
diabolo-if-net - Text filter for Diabolo printer `scrivener'
listening
on port 5100. Installed in /usr/local/libexec/diabolo-if-net
#
exec /usr/libexec/lpr/lpr "$@" | /usr/local/libexec/netprint scrivener
5100
```

#### 11.4.4. Restringindo o Uso da Impressora

Essa seção dá informações de como restringir o uso da impressora. O sistema **LPD** permite que você controle quem pode acessar a impressora localmente ( direto na máquina) ou remotamente, eles podem imprimir multiplas cópias, tão grandes quando se queira os trabalhos de impressão, tão grandes que podem ser as filas de impressão.

##### 11.4.4.1. Restringindo Multiplas Cópias

O sistema **LPD** faz isso com facilidade para os usuários poderem imprimir multiplas cópias de um arquivo. Os usuários podem imprimir trabalhos de impressão com o comando `lpr -#5`, por exemplo e ter cinco cópias de um mesmo arquivo em um trabalho de impressão. Isso será um boa ideia se estiver habilitado.

Se você perceber que isso pode causar um desgaste desnecessário da impressora, você poderá desabilitar a opção `-#` para [lpr\(1\)](#) apenas adicionando a opção `sc` no arquivo `/etc/printcap`. Quando os usuários utilizarem o comando com a opção `-#` para imprimir seus trabalhos de impressão, ele verá:

```
lpr: multiple copies are not allowed
```

Note que se você tem configurado o acesso a impressora remota ( veja a seção [Impressora Instaladas em Máquinas Remotas](#) ), você necessitará da opção `sc` no arquivo `/etc/printcap` da máquina remota, caso contrário, os usuários poderiam enviar trabalhos de impressão com multiplas cópias, usando uma outra máquina na rede.

Aqui tem um exemplo. Esse é um arquivo `/etc/printcap` para a máquina `rose`. A impressora `rattan` parruda, que aguenta multiplas cópias, mas a impressora à laser `bamboo` é um pouco mais delicada ,que não, então será desabilitada a multiplas cópias, adicionando a opção `sc` :

```
#
/etc/printcap for host rose - restrict multiple copies on bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
 :sh:sd=/var/spool/lpd/rattan:\
 :lp=/dev/lpt0:\
 :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
 :sh:sd=/var/spool/lpd/bamboo:sc:\
 :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
 :if=/usr/local/libexec/psif:\
 :df=/usr/local/libexec/psdf:
```

Agora, nós necessitaremos de adicionar a opção `sc` no arquivo `/etc/printcap` na máquina `orchid` para dessabilitar as cópias multiplas para a impressora `teak`):

```
#
/etc/printcap for host orchid - no multiple copies for local
printer teak or remote printer bamboo
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
 :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:sc:\
 :if=/usr/local/libexec/ifhp:\
 :vf=/usr/local/libexec/vfhp:\
 :of=/usr/local/libexec/ofhp:

rattan|line|diablo|lp|Diablo 630 Line Printer:\
 :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
 :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo:sc:
```

Usando a opção *sc*, nos podemos prever o uso do comando `lpr -#`, mas isso não prevê usuários utilizando [lpr\(1\)](#) várias vezes, ou de enviar o mesmo arquivo várias vezes em um único trabalho de impressão, como segue:

```
% lpr forsale.sign forsale.sign forsale.sign forsale.sign forsale.sign
```

Existem várias maneiras de se prever esse abuso ( inclusive ignorando ele) que você pode explorar livremente.

#### 11.4.4.2. Restringindo o Acesso as Impressoras

Você pode controlar quem pode imprimir em qual impressora utilizando o mecanismo de grupos do UNIX® e a opção *rg* no `/etc/printcap`. Apenas colocando os usuários que você quer que tenham acesso a impressora diferentes grupos, e então definindo na opção *rg* o grupo desejado.

Os usuários fora do grupo ( incluindo o `root`) irão receber a seguinte mensagem:  
``lpr: Not a member of the restricted group" Se eles tentarem imprimir em uma impressora controlada.

Como a opção *sc* ( suprime multiplas cópias), você precisará de especificar a opção *rg* na máquina remota (servidor) para ter também o acesso à impressora, veja a seção [Impressoras Instaladas em Servidores Remotos](#) ).

Por exemplo, nós iremos permitir o acesso de todos a impressora *rattan*, mas somente os usuários do grupo *artists* poderão utilizar a impressora *bamboo*. Novamente o arquivo `/etc/printcap` é semelhante ao da máquina *rose*:

```
#
/etc/printcap for host rose - restricted group for bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
 :sh:sd=/var/spool/lpd/rattan:\
 :lp=/dev/lpt0:\
 :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
 :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:\
 :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
 :if=/usr/local/libexec/psif:\
 :df=/usr/local/libexec/psdf:
```

Permita-nos mostrar um outro exemplo do arquivo `/etc/printcap` (para a máquina *orchid*) sozinha. Claro, qualquer um na máquina *orchid* pode imprimir para a impressora *bamboo*. Neste caso, nos deveremos permitir acesso à apenas de alguns usuários na máquina *orchid*, de qualquer maneira, nos estamos controlando o acesso a impressora.

**Nota:** Só podemos restringir apenas um grupo por impressora

### 11.4.4.3. Controlando o Tamanho do Arquivos Enviados

Se você possui muitos usuários acessando a sua impressora, você provavelmente necessitará de limitar o tamanho máximo dos arquivos enviados para a impressora. Além de que, o espaço livre em disco é limitado, você deverá prever o envio de vários trabalhos de impressão simultaneos, que poderão consumir uma grande quantidade de espaço em disco.

O sistema **LPD** permite que você limite o tamanho máximo de um trabalho de impressão através da opção *mx*. A unidade padrão é em blocos de *BUFSIZ*, que são 1024 bytes. Se você colocar o valor como zero para esta opção, você não estará limitando o tamanho do arquivo, entretanto se você não especificar valor, o padrão será utilizado, que é de : 1000 blocos.

**Nota:** O limite é aplicado para os *arquivos* dentro de um trabalho de impressão, e *não* ao tamanho total do trabalho de impressão

O sistema **LPD** não irá recusar o arquivo que estiver acima do limite imposto por você. Ele colocara na fila de impressão apenas a quantidade de bytes até o limite. O resto será descartado. Será esse o comportamento correto.

Iremos utilizar outro exemplo de limites para as impressoras *rattan* e *bamboo*. Uma vez que arquivos artísticos em PostScript tentem a ser muito grande, nós iremos limitar-los a 5 megabytes. Nós iremos deixar sem limite a impressora de texto puro (matricial):

```
#
/etc/printcap for host rose
#

#
No limit on job size:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
 :sh:mx#0:sd=/var/spool/lpd/rattan:\
 :lp=/dev/lpt0:\
 :if=/usr/local/libexec/if-simple:

#
Limit of five megabytes:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
 :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:mx#5000:\
 :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
 :if=/usr/local/libexec/psif:\
 :df=/usr/local/libexec/psdf:
```

Relembrando, os limites são aplicados somente aos usuários locais. Se você permite acesso remoto, usuários remotos não terão este limite. Você deverá especificar na opção *mx* no arquivo remoto */etc/printcap*. Veja a seção [Impressoras Instaladas em Servidores Remotas](#) para mais detalhes.

Existem outros meios especializados para limitar o tamanho dos trabalhos de impressão de impressão remota; veja a seção [Restringindo Impressão Remota](#) .

#### 11.4.4.4. Restringindo a Impressão Remota

O sistema de spooling **LPD** possui vários meios de restringir o acesso dos trabalhos de impressão enviados de uma máquina remota:

##### Restrição à Máquina

Você pode controlar de qual máquina remota o servidor local **LPD** aceitará pedidos de impressão com a ajuda dos arquivos `/etc/hosts.equiv` e `/etc/hosts.lpd`. O **LPD** verifica esses arquivos para saber se o pedido foi feito por uma das máquinas listada em um dos arquivos. Se não estiver em nenhum deles o **LPD** recusará o pedido

O formato desses arquivos é muito simples: o nome da máquina remota, um por linha. Note que o arquivo `/etc/hosts.equiv` é também utilizado pelo protocolo [ruserok\(3\)](#), e afeta programas como [rsh\(1\)](#) e [rcp\(1\)](#), então, seja cuidadoso.

Por exemplo, aqui está um arquivo `/etc/hosts.lpd` para a máquina `rose`:

```
orchid
violet
madrigal.fishbaum.de
```

Isso significa que a máquina `rose` irá aceitar pedidos das máquinas `orchid`, `violet`, e `madrigal.fishbaum.de`. Se qualquer outra máquina tentar acessar o **LPD** na máquina `rose`, esse trabalho de impressão será rejeitado.

##### Restrição de Tamanho

Você pode controlar o quanto de espaço livre é necessário no sistema de arquivos onde o diretório de spooling está. Crie um arquivo chamado `minfree` dentro do diretório do spooling para a impressora local. Insira no arquivo o número representando quantos blocos ( 512 bytes ) de espaço livre deve ter o disco para que o trabalho de impressão remoto seja aceito.

Isso permite que você garanta que o seu sistema de arquivos não fique lotado pelos seus usuários remotos. Você também pode usar isso para dar uma certa prioridade aos usuários locais: Eles estão aptos a enviar grandes trabalhos de impressão mesmo que o limite fique abaixo do especificado no arquivo `minfree`.

Por exemplo, vamos adicionar um arquivo `minfree` para a impressora `bamboo`. Nós examinaremos o arquivo `/etc/printcap` para encontrar o diretório de spooling para essa impressora; aqui está a entrada:

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
:sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:mx#5000:\
:lp=/dev/ttyd5:ms#-parenb cs8 clocal
crtsets:rw:mx#5000:\
:if=/usr/local/libexec/psif:\
:df=/usr/local/libexec/psdf:
```

O diretório de spooling é dado na opção *sd*. Nós iremos criar uma reserva de 3 megabytes ( que são 6144 blocos do disco ) para o **LPD** poder aceitar pedidos de trabalhos de impressão remotos:

```
echo 6144 > /var/spool/lpd/bamboo/minfree
```

#### Restrições de Usuários

Para que você possa controlar qual o usuário remoto pode acessar a impressora local, voce deve utilizar a opção *rs* do arquivo */etc/printcap*. Quando a opção *rs* é utilizada localmente, o **LPD** irá somente aceitar os trabalhos de impressão das máquinas remotas *se* o usuário que o tiver enviando, também tiver conta com o mesmo *login* na máquina local. Caso contrário, o **LPD** irá recusar o trabalho de impressão.

Essa opção é particularmente poderosa para o ambiente onde existam, por exemplo, diferentes departamentos compartilhando a rede, e alguns usuários transcendem os limites departamentais. Dando a eles contas em seus sistemas, eles também poderão imprimir nas impressoras desses departamentos. Se você quiser que apenas a utilização desses usuários sejam restritas a utilização das impressoras, você deve utilizar contas do tipo ``token'', sem diretório de usuário e com um shell sem utilização como */usr/bin/false*.

### 11.4.5. Contabilizando o Uso da Impressora

Então, você precisa contabilizar as impressões. Por que não? Afinal papel e tinta custam dinheiro. E também o custo de manutenção da impressora com as partes móveis que podem quebrar. Você deve examinar suas impressoras, use um padrão e uma taxa de manutenção que deve conter o custo de uma página ( ou por metro, por pés, ou qualquer outra coisa ). Agora como você irá inicializar a contabilidade para as impressões?

Bem, a má notícia é que o sistema de spooling **LPD** não provê muita ajuda nesse departamento. A contabilidade é altamente dependente da maneira de como a impressora é utilizada, o formato que é impresso, como *voce* necessita que a impressora seja contabilizada.

Para implantar a contabilidade, você deve modificar o filtro para texto puro ( para contabilizar trabalhos de impressão em texto puro) e os filtros de conversão ( para contabilizar outros formatos de impressão), para contar as páginas ou as requisições para a impressora das páginas impressas. Você não deve utilizar filtros simples, pois eles não fornecem dados para a contabilidade. Veja a seção [Filtros](#).

Geralmente, existe duas maneiras para realizar a contabilidade:

- *Contabilidade Periódica* é feita pela maneira mais simples. A toda hora é impresso algo, e o filtro armazena o usuário, a máquina, e o número de páginas impressas no arquivo de contabilidade. A cada mes, semestre, ano ou a qualquer período que você preferir, você pode coletar a contagem de páginas, através do arquivo de contabilidade para várias impressoras. Então você poderá zerar o arquivo de contabilidade para que se faça novo período de contabilidade.

- *Contabilidade temporizada* é menos comum, provavelmente porque é mais difícil. Esse método tem filtros que contabilizam as impressões tão logo que se use as impressoras. Como a limitação de disco, a contabilidade é imediata. Você pode impedir que os usuários imprimam quando suas cotas cheguem ao vermelho, e devam providenciar outro meio para os usuários verificarem e ajustar suas ``cotas de impressão". Mas esse método requer algum código de banco de dados para rastrear os usuários e suas cotas.

O sistema de spool **LPD** suporta ambos os métodos facilmente: deste que você providencie os filtros ( na maioria das vezes ), você também deve providenciar o código para a contabilidade. Mas existe o lado bom: você terá uma enorme flexibilidade em seus métodos de contabilidade. Por exemplo, você escolhe como irá fazer a contabilidade, periodicamente ou em determinado tempo. Você escolhe qual a informação que irá armazenar: nome do usuário, nome da máquina, tipo do trabalho de impressão, páginas impressas, metragem quadrada do papel utilizado, quanto tempo a impressora foi utilizada para a impressão em muito mais. E você só terá que modificar o filtro para armazenar essas informações.

### 11.4.5.1. Uma Contabilidade Rápida e Rasteira

O FreeBSD vem com dos programas que podem ajuda-lo com uma contabilidade simples e periodica. Eles são filtros texto (`lpf`), descrito na seção [lpf: um Filtro Texto](#), e [pac\(8\)](#), um programa para recolher e totalizar as entradas do arquivo de contabilidade das impressoras.

Como mencionado na seção sobre filtros ([Filtros](#)), O **LPD** inicia os filtros de texto e de conversão com o nome do arquivo de contabilidade para ser utilizado na linha de comando do filtro. Os filtros podem usar esse argumento para saber onde escrever os dados da contabilidade. O nome desse arquivo deve ser colocado na opção *af* no arquivo `/etc/printcap`, e se não estiver especificado com o diretório absoluto, será utilizado de forma relativa ao diretório de spooling.

O **LPD** inicia o `lpf` com a largura e o comprimento da página como os argumentos ( da opções *pw* e *pl*). O `lpf` usa esses argumentos para determinar quanto de papel será utilizado. Depois do envio do arquivo para impressora, ele então escreve a entrada da contabilidade para o arquivo de contabilidade. Uma entrada seria semelhante a essa:

```
2.00 rose:andy
3.00 rose:kelly
3.00 orchid:mary
5.00 orchid:mary
2.00 orchid:zhang
```

Você deve usar arquivos de contabilidade separados para cada impressora, como `lpf` não tem lógica de bloqueio interna, e dois acessos via `lpf` pode comrromper os dados das entradas concorrentes. O meio mais fácil de separar os arquivos de contabilidade é utilizando a opcao *af=acct* no arquivo `/etc/printcap`. Então, cada arquivo de contabilidade irá ficar no diretório de *spooling* da impressora desejada, com o nome de `acct`.

Quando você estiver pronto para cobrar dos usuários pelas impressões, execute o programa [pac\(8\)](#). Para cada impressora que você deseja coletar os dados, você deverá ir ao seu diretório de *spooling* e então executar o programa *pac*. Você terá um relatório dos custos, semelhante à:

Login	pages/feet	runs	price
orchid:kelly	5.00	1	\$ 0.10
orchid:mary	31.00	3	\$ 0.62
orchid:zhang	9.00	1	\$ 0.18
rose:andy	2.00	1	\$ 0.04
rose:kelly	177.00	104	\$ 3.54
rose:mary	87.00	32	\$ 1.74
rose:root	26.00	12	\$ 0.52
total	337.00	154	\$ 6.74

Esses são os argumentos para o comando [pac\(8\)](#):

*-P*Impressora

Onde *Impressora* será a impressora ser avaliada. Essa opção funciona somente se existir o caminho absoluto na opção *af* no arquivo `/etc/printcap`.

*-c*

Ordena a saída pelo custo ao invés da ordem alfabética dos nomes dos usuários.

*-m*

Ignora o nome da máquina no arquivo de contabilidade. Com essa opção, o usuário `smith` na máquina `alpha` é o mesmo usuário `smith` na máquina `gamma`. Sem isso serão diferentes.

*-ppreço*

Computa o uso com o *preço* em dolar por página ou por pés, ao invés do preço estipulado na opção *pc* no arquivo `/etc/printcap`, or 3 centavos (o valor padrão). Você pode especificar o *preço* como um número de ponto flutuante.

*-r*

Ordena de modo reverso.

*-s*

Cria um arquivo com o resumo da estatística e trunca o arquivo de contabilidade.

*nome ...*



Imprime as informações de contabilidade de um dado usuário com o nome *names* somente.

No resumo padrão que o comando [pac\(8\)](#) gera, você vê o número de páginas impressas por cada usuário de várias máquinas. Se, no seu sítio, as máquinas não são importantes (porque os usuários podem usar qualquer uma delas), então execute o comando `pac -m`, para gerar a seguinte saída:

Login	pages/feet	runs	price
andy	2.00	1	\$ 0.04
kelly	182.00	105	\$ 3.64
mary	118.00	35	\$ 2.36
root	26.00	12	\$ 0.52
zhang	9.00	1	\$ 0.18
total	337.00	154	\$ 6.74

Para computar o custo em dolar, o comando [pac\(8\)](#) usa a opção *pc* do arquivo `/etc/printcap` (padrão é 200, ou 2 centavos por página). Especifique, em centenas de centavos, o preço por página ou por pés se você possuir esse tipo de impressão. Você pode sobrescrever este valor quando você executa o comando [pac\(8\)](#) com a opção *-p*. A unidade para esta opção é o dollar e não a centena de centavos. Por exemplo:

```
pac -p1.50
```

Faz o custo de cada página ter 1,50 dolares. Você pode realmente ter um lucro usando essa opção.

Finalmente, executando o comando `pac -s`, você economizará informações no resumo da contabilidade armazenado em arquivo, o qual terá o mesmo nome que o arquivo de estatística da impressora, mas com um sufixo acrescido ao nome (*\_sum*). Ele então trunca o arquivo de estatísticas. Quando você executa o comando [pac\(8\)](#) novamente, ele re-le o resumo do arquivo de resumo para inicial a totalização, então adiciona as informações no arquivo normal de estatísticas.

### 11.4.5.2. Como Contar Páginas Impressas?

De modo a apurar mesmo uma contagem pouco precisa, você deve ser capaz de determinar quanto papel um trabalho usa. Este é um dos problemas essenciais da contabilidade em relação à impressão.

Para trabalhos ou *jobs* de texto simples o problema não é muito difícil: você conta quantas linhas têm no trabalho e compara isso com quantas linhas por página sua impressora suporta. Não se esqueça de levar em consideração *backspaces* no arquivo que provoquem sobrescrita de linhas e linhas longas que se transformam em uma ou mais linhas adicionais.

O filtro de texto `lpf` (cuja introdução pode ser vista em [lpf: Um Filtro de Texto](#)) leva em consideração estas coisas quando faz a contagem. Se você estiver escrevendo um filtro de texto que precise realizar contabilidade pode querer examinar o código fonte do `lpf`.

E como você lida com outros formatos?

Bem, para conversões de DVI para LaserJet ou DVI para PostScript você pode fazer com que seu filtro trate a saída do comando `dvilj` ou `dvips` e verifique quantas páginas foram convertidas. Você deve conseguir fazer isso também com outros formatos e programas de conversão.

Esses métodos, porém, não resolvem o problema de que a impressora pode não imprimir todas essas páginas. Por exemplo, o papel pode prender, a tinta acabar ou a impressora explodir, e o usuário ainda seria cobrado pelas páginas.

Então, o que você pode fazer?

Existe somente um meio *correto* para fazer uma contabilidade *precisa*. Deixe a impressora dizer para você quanto de papel ela usou, e a conecte via porta serial ou a uma conexão de rede. Quase todas as impressoras PostScript suportam essa manipulação. Outras marcas e modelos também podem, por exemplo, impressoras de rede Imagen Laser. Modifique os filtros para essas impressoras para pegar a utilização das páginas após a impressão de cada trabalho, para então poder armazenar as informações da contabilidade baseado apenas neste valor. Não existe contagem de linha e nem exame de erros no arquivo.

Claro, você pode ser generoso e deixar as impressões gratuitas.

---

## 11.5. Usando as Impessoras

Essa seção mostra à você como utilizar as impressoras que você configurou com o FreeBSD. Aqui tem uma visão geral dos comandos a nível de usuário:

[lpr\(1\)](#)

Imprime os trabalhos de impressão

[lpq\(1\)](#)

Verifica a fila da impressora

[lprm\(1\)](#)

Remove os trabalhos de impressão da fila da impressora

Existe também o comando administrativo, [lpc\(8\)](#), descrito na seção [Administrando o Sistema de Spooler LPD](#), usado para controlar as impressoras e suas filas.

Todos esses três comandos [lpr\(1\)](#), [lprm\(1\)](#), e [lpq\(1\)](#) aceitam a opção `-P Nome-da-Impressora` para especificar qual a impressora/fila irá operar, como listado no arquivo

`/etc/printcap`. Ele habilita que você possa enviar, remover e checar os trabalhos de impressão para várias impressoras. Se você não usa a opção `-P`, então esses comandos usam a impressora especificada na variável de ambiente `PRINTER`. Finalmente se você não tiver a variável de ambiente `PRINTER`, esses comandos por padrão usam a impressora `lp`.

No futuro, a terminologia *impressora padrão* será o nome que estiver na variável de ambiente `PRINTER`, ou a impressora chamada de `lp` quando não existir a variável de ambiente `PRINTER`.

### 11.5.1. Trabalhos de Impressão

Para imprimir arquivos, digite:

```
% lpr filename ...
```

Ele imprime cada arquivo listado na impressora padrão. Se sua lista não contém, [lpr\(1\)](#) lê os dados para serem impressos da entrada padrão. Por exemplo, esse comando imprime alguns importantes arquivos do sistema:

```
% lpr /etc/host.conf /etc/hosts.equiv
```

Para selecionar uma impressora específica, digite:

```
% lpr -P nome-da-impressora nome-do-arquivo ...
```

Esse exemplo imprime uma listagem do diretório corrente para a impressora `rattan`:

```
% ls -l | lpr -P rattan
```

Como não há arquivos a serem impressos pelo comando [lpr\(1\)](#), O comando `lpr` irá ler da entrada padrão, a qual é saída do comando `ls -l`.

O comando [lpr\(1\)](#) também pode aceitar uma grande variedade de opções para controlar a formatação, aplicando conversão de arquivos, gerando múltiplas cópias, e muito mais. Para mais informação, veja a seção [Opções de Impressão](#).

### 11.5.2. Verificando a Impressão

Quando você imprime com o comando [lpr\(1\)](#), os dados que você gostaria que saíssem juntos em um pacote chamado de "trabalho de impressão", o qual é enviado para o sistema de spooling **LPD**. Cada impressora tem sua própria fila de impressão para os trabalhos de impressão, e seus trabalhos de impressão aguardam na fila a impressão de outros trabalhos seus ou de outros usuários. A impressora imprime esses trabalhos de impressão na seguinte ordem (FIFO) primeiro a chegar é o primeiro a ser impresso.

Para visualizar a fila para a impressora padrão, digite: [lpq\(1\)](#). Para especificar a impressora, use a opção `-P`. option. Por exemplo, o comando

```
% lpq -P bamboo
```

mostrará a fila da impressora chamada de *bamboo*. Aqui está um exemplo da saída do comando `lpq`:

```
bamboo is ready and printing
Rank Owner Job Files Total Size
active kelly 9 /etc/host.conf, /etc/hosts.equiv 88 bytes
2nd kelly 10 (standard input) 1635 bytes
3rd mary 11 ... 78519 bytes
```

Ele mostra três trabalhos de impressão na fila da impressora *bamboo*. O primeiro trabalho, enviado pela usuária *kelly*, tem atribuído um ``número de fila" 9. Cada trabalho de impressão tem um número único. A maioria das vezes você pode ignorar este número, mas você irá precisar dele se quiser cancelar o trabalho de impressão; veja na seção [Removendo Trabalhos de Impressão](#) para mais detalhes.

O número de trabalho de Impressão 9 consiste em dois arquivos; múltiplos arquivos dado em um comando `lpr(1)` é tratado como parte de um simples trabalho de impressão. Ele é o trabalho de impressão atual ( note a palavra *active* na coluna ``Rank"), a qual diz qual deve ser o trabalho de impressão ativo. O segundo trabalho de impressão consiste de dados passados pela entrada padrão para o comando `lpr(1)`. O terceiro vem do usuário *mary*; o qual é um trabalho de impressão muito grande. O nome do diretório do seu arquivo é tão grande que não cabe, então o comando `lpq(1)` mostra os pontos da árvore.

A primeira linha da saída do comando `lpq(1)` é igualmente importante: ela informa o que a impressora está fazendo ( ou o que o **LPD** pensa que ela fazendo em última instância ).

O comando `lpq(1)` suporta também a opção `-l` para gerar uma longa lista detalhada. Aqui está um exemplo `lpq -l`:

```
waiting for bamboo to become ready (offline ?)
kelly: 1st [job 009rose]
 /etc/host.conf 73 bytes
 /etc/hosts.equiv 15 bytes

kelly: 2nd [job 010rose]
 (standard input) 1635 bytes

mary: 3rd [job 011rose]
 /home/orchid/mary/research/venus/alpha-regio/mapping 78519 bytes
```

### 11.5.3. Removendo a Trabalho de Impressão

Você pode remover um trabalho de impressão da fila com o comando `lprm(1)`. Frequentemente, você irá utilizar o comando `lprm(1)` para remover o trabalho de impressão ativo, mas muitas das vezes ou todas as vezes, a impressão continua.

Para remover o trabalho de impressão da impressora padrão, você deverá primeiro encontrar o número do trabalho de impressão com o comando `lpq(1)`, então digite:

```
% lprm job-number
```

Para remover o trabalho de impressão de uma impressora específica, você deve utilizar a opção *-P*. O comando a seguir remove o trabalho de impressão de número 10 da fila de impressão da impressora *bamboo*:

```
% lprm -P bamboo 10
```

O comando [lprm\(1\)](#) possui alguns atalhos:

*lprm -*

Remove todos os trabalhos de impressão ( para a impressora padrão )  
pertencente a você.

*lprm usuário*

Remove todos os trabalhos de impressão ( para a impressora padrão )  
pertencentes ao usuário *usuário*. Somente o super-usuário pode remover os  
trabalhos de impressão de outros usuários, e você só pode remover os seus  
próprios.

*lprm*

Sem o número do trabalho de impressão, ou com a opção *-* aparecendo na linha  
de comando, [lprm\(1\)](#) irá remover o trabalho de impressão ativo ( corrente ) da  
impressora padrão se este for pertencente a você, caso contrário só o super-  
usuário poderá remove-lo.

Utilize a opção *-P* para poder especificar a impressora sem ser a padrão. Por exemplo, o  
seguinte comando remove todos os trabalhos de impressão para o usuário corrente na  
fila da impressora chamada de *rattan*:

```
% lprm -P rattan -
```

**Nota:** Se você está trabalhando em um ambiente de rede, o comando [lprm\(1\)](#) permitirá  
à você remover os trabalhos de impressão apenas enviados pela máquina que os enviou.  
Veja a sequência de comandos abaixo para demonstrar isso:

```
% lpr -P rattan myfile
% rlogin orchid
% lpq -P rattan
Rank Owner Job Files Total Size
active seeyan 12 ... 49123 bytes
2nd kelly 13 myfile 12 bytes
% lprm -P rattan 13
rose: Permission denied
% logout
% lprm -P rattan 13
dfA013rose dequeued
cfA013rose dequeued
```

## 11.5.4. Além do Texto Puro: Opções de Impressão

O comando `lpr(1)` suporta um número de opções que controla a formatação do texto, converte `graphico` e outros formatos de arquivos, produzindo múltiplas cópias, manipulando o trabalho de impressão, e muito mais. Essa seção descreve as opções.

### 11.5.4.1. Opções de Formatação e Conversão

As seguintes opções do comando `lpr(1)` controla a formatação dos arquivos no trabalho de impressão. Use essas opções se o trabalho de impressão não contiver texto puro ou se você quiser uma saída formatada de texto puro através do utilitário `pr(1)`.

Por exemplo, o comando a seguir imprime um arquivo DVI ( de um sistema **TeX**) chamado de `fish-report.dvi` para a impressora chamada de *bamboo*:

```
% lpr -P bamboo -d fish-report.dvi
```

Essas opções é aplicada em cada arquivo do trabalho de impressão, você não pode misturar (digamos) arquivos nos formatos DVI e ditroff no mesmo trabalho de impressão. Sendo assim envie separadamente os arquivos, usando opções de conversão diferentes para cada um deles.

**Nota:** Para todas essas opções, exceto as `-p` e `-T`, requerem um filtro conversor instalado para a impressora de destino. Por exemplo, a opção `-d` precisa do filtro de conversão para DVI. A seção [Filtros Conversores](#) dá mais detalhes.

`-c`

imprime arquivos do tipo `cifplot`.

`-d`

Imprime arquivos do tipo DVI.

`-f`

Imprime arquivos do tipo FORTRAN.

`-g`

Imprime arquivos do tipo `plot`.

`-i número`

Para alinhar a saída por um *número* de colunas; se você omitir *número*, o alinhamento será de 8 colunas. Essa opção trabalha somente com alguns filtros conversores.

**Nota:** Não coloque nenhum espaço entre a opção `-i` e o número.

-l

Imprime arquivo do tipo literal, incluindo caracteres de controle.

-n

Imprime arquivo do tipo ditroff (troff independente do dispositivo).

-p

Formata o texto puro com o comando [pr\(1\)](#) antes da impressão. Veja [pr\(1\)](#) para maiores informações.

-T *título*

Usa o *título* no cabeçalho do comando [pr\(1\)](#) ao invés do nome do arquivo. Essa opção afeta somente quando usado em conjunto com a opção *-p*.

-t

Imprime arquivo do tipo troff.

-v

Imprime arquivo do Tipo raster.

Aqui um exemplo: Esse comando imprime uma versão da página de manual do comando [ls\(1\)](#) na impressora padrão:

```
% zcat /usr/share/man/man1/ls.1.gz | troff -t -man | lpr -t
```

O comando [zcat\(1\)](#) descompacta o código fonte da página do manual [ls\(1\)](#) e transfere para o comando [troff\(1\)](#), o qual formata o código e faz a saída do GNU troff ir para [lpr\(1\)](#), o qual envia o trabalho de impressão para o *spooler* do **LPD**. Por causa do uso da opção *-t* para o comando [lpr\(1\)](#), o spooler irá converter a saída do GNU troff em um formato que a impressora possa compreender e então será impresso o trabalho de impressão.

#### 11.5.4.2. Opções de Manipulação de Impressão

As opções a seguir para o comando [lpr\(1\)](#) informa ao **LPD** como manipular os trabalhos de impressão:

-# *cópias*

Produz o número de *cópias* de cada arquivo no trabalho de impressão ao invés de uma única cópia. O administrador pode desabilitar essa opção para reduzir o desgaste da impressora e insentivar a fotocópia. Veja a seção [Restringindo o Número de Cópias](#) .

Esse exemplo imprime tres cópias do arquivo `parser.c` seguido de tres cópias do arquivo `parser.h` na impressora padrão:

```
% lpr -#3 parser.c parser.h
```

-m

Envia um e-mail apos o término da impressão do trabalho de impressão. Com essa opção, o sistema **LPD** irá enviar um e-mail para a sua conta quando ele terminar de gerenciar seu trabalho de impressão. Na mensagem, ele ira informar à você se a impressão terminou com sucesso ou com falhas e, geralmente, qual foi o erro ocorrido.

-s

Não faz cópias de os arquivos para o diretório de spooling, mas ao inves disso, ele cria um link simbólico.

Se você está imprimindo um trabalho de impressão grande, provavelmente essa opção irá ajuda-lo. Isso economiza espaço no diretório de spooling, você pode ter um consumo maior que o seu espaço disponível no sistema do arquivo onde fica esse diretório. Isso economiza tempo porque o **LPD** não precisa copiar o arquivo cada vez que o seu trabalho de impressão for impresso.

Entretanto, existe um custo: deste que o **LPD** irá acessar o arquivo original diretamente, você não poderá modifica-lo ou remove-lo até que tenha terminado a sua impressão.

**Nota:** Se você está imprimindo para uma impressora remota, o **LPD** irá eventualmeten que copiar os arquivos da máquina local para a máquina remota, então a opção `-s` irá apenas economizar espaço na máquina local, não na máquina remota. Ainda assim é uma opção muito útil.

-r

Remove os arquivos do trabalho de Impressão apos terem sido copiados para o diretório de spool, ou apos a impressão deles com a opção `-s`. Seja cuidadoso com essa opção!

### 11.5.4.3. Opções de Páginas de Cabeçalho

Essa opção do [lpr\(1\)](#) ajusta o texto que normalmente aparece no cabeçalho do trabalho de impressão. Se o cabeçalho for suprimido para a impressora de destino, essa opção não terá efeito algum. Veja a seção [Páginas de Cabeçalho](#) para informar-se sobre a configuração.

-C *texto*

Troca o nome da máquina na página de cabeçalho pelo *texto*. O nome da máquina é normalmente o da máquina que enviou o trabalho de impressão.



-J *texto*

Troca o nome do trabalho de impressão na página de cabeçalho pelo *texto*. O nome do trabalho de impressão normalmente é o nome do primeiro arquivo do trabalho de impressão, ou `stdin` se você estiver imprimindo da entrada padrão.

-h

Não imprime a página de cabeçalho.

**Nota:** Em alguns locais, essa opção pode não ter efeito algum, uma vez que a páginas de cabeçalho são sempre gerada. Veja a seção [Páginas de Cabeçalho](#) para maiores detalhes.

### 11.5.5. Administrando Impressoras

Como um administrador para suas impressoras, você deve tê-las instaladas, configuradas e testadas. Usando o comando [lpc\(8\)](#), você pode interagir com suas impressoras de vários meios, com o comando [lpc\(8\)](#), você pode

- Iniciar ou parar a impressora
- Habilitar ou desabilitar as filas de impressão
- Rearrumar a ordem dos trabalhos de impressão em cada fila.

Primeiro, uma nota sobre a terminologia: se a impressora estiver *parada*, Ela não poderá imprimir nada que estiver na fila. Usuários poderão enviar os seus trabalhos de impressão, os quais aguardarão na fila até que a impressora seja *inicializada* ou a fila seja limpa.

Se a fila estiver *desabilitada*, nenhum usuário ( exceto o `root`) poderá enviar os trabalhos de impressão para a impressão. Quando a fila for *abilitada* permitirá que os usuários possam enviar seus trabalho de impressão. A impressora pode ser *inicializada* para que a fila seja desabilitada, nesse caso ela continuará imprimindo o trabalho de impressão até que a fila esteja vazia.

Em geral, você deve ter privilégios de `root` para usar o comando [lpc\(8\)](#). Usuários normais podem utilizar o comando [lpc\(8\)](#) somente para pegar o status da impressora e para reinicializar uma impressora travada.

Aqui está um resumo do comando [lpc\(8\)](#). A maioria dos comandos tem o *nome da impressora* como argumentopara informar qual é a impressora que queremos operar. Você pode usar *all* para o *nome da impressora* para ver/operar em todas as impressoras descritas no arquivo `/etc/printcap`.

```
abort nome da impressora
```

Cancela o trabalho corrente e para a impressora. Usuários podem enviar trabalhos de impressão se a fila estiver habilitada.

```
clean nome da impressora
```

Remove arquivos antigos que fazem parte de um trabalho de impressão que não foi removido corretamente pelo **LPD**, particularmente se existir erros durante a impressão ou uma grande atividade de administração. Esse comando encontra arquivos que não residem mais no diretório de spool e os remove.

`disable nome da impressora`

Desabilita a fila de impressão para novos trabalhos de impressão. Se a impressora estiver em funcionamento, ela irá continuar a imprimir até que a fila fique vazia. O superusuário (`root`) sempre pode enviar trabalhos para a impressora, mesmo com a fila desabilitada.

Esse comando é muito útil quando você está testando uma nova impressora ou a instalação de um filtro: desabilita a fila e envia os trabalhos de impressão como `root`. Outros usuários não poderão enviar trabalhos de impressão até que você complete seus testes e reabilite a fila com o comando `enable`.

`down nome da impressora mensagem`

Coloca a impressora inativa. Equivalente ao comando `disable` seguido do comando `stop`. A *mensagem* aparece como o status da impressora quando o usuário checar a fila da impressora com o comando `lpq(1)` ou o status com o comando `lpc status`.

`enable nome da impressora`

Abilita a fila para a impressora. Os usuários poderão enviar trabalhos de impressão, mas a impressora não irá imprimir nada até que ela seja inicializada.

`help nome do comando`

MOstra um texto de ajuda sobre o comando *nome do comando*. Sem o *nome do comando*, mostra um resumo dos comandos disponíveis.

`restart nome da impressora`

Inicializa a impressora. Usuários comuns podem usar este comando se alguma circunstância extraordinária trave o **LPD**, mas ele não pode iniciar a impressora parada pelos comandos `stop` ou `down`. O comando `restart` é equivalente aos comandos `abort` seguido por `start`.

`start nome da impressora`

Inicializa a impressora. A impressora irá imprimir os trabalhos de impressão que estiverem na fila.

`stop nome da impressora`

Para a impressora. A impressora irá terminar a impressão do trabalho de impressão atual e não irá imprimir mais nada que estiver na fila. Mesmo que ela esteja parada, os usuários podem enviar seus trabalhos de impressão.

```
topq nome da impressora trabalho de impressão ou nome do usuário
```

Reorganiza a fila para *nome da impressora* por colocar o trabalho de impressão com o número de *trabalho de impressão* ou o trabalho de impressão do *nome do usuário* no topo da fila. Para esse comando você não pode utilizar *all* como o *nome da impressora*.

```
up nome da impressora
```

Coloca a impressora no ar; oposto ao comando *down*. Equivale aos comandos *start* seguido por *enable*.

o comando [lpc\(8\)](#) aceita os comandos acima na linha de comando. Se você não colocar comandos como argumentos, o [lpc\(8\)](#) entrará no modo interativo, onde você poderá entrar com os comandos até você digitar *exit*, *quit*, ou fim de arquivo ( EOF ).

---

## 11.6. Alternativas para o *Spooler* Padrão

Se você estiver lendo este manual todo, você deve ter aprendido tudo sobre o sistema de impressão **LPD** que vem com o FreeBSD. E você provavelmente irá ter em mente uma pergunta: "Quais são os outros sistemas de impressão disponíveis e que funcionem no FreeBSD? "

### LPRng

O **LPRng**, que significa simplesmente "LPR: a próxima Geração (Next Generation)" é uma completamente reescritura do PLP. Patrick Powell e Justin Mason (o principal mantenedor do PLP) colaboraram para fazer o **LPRng**. O sítio principal dele é <http://www.lprng.org/>.

### CUPS

O **CUPS**, Sistema Comum de impressão para o UNIX (the Common UNIX Printing System), provê um nível (camada) protável de impressão para os sistemas operacionais baseados em UNIX®. Ele foi desenvolvido pela Easy Software Products para promover uma solução padrão para todo e qualquer usuário e vendedor de UNIX.

O **CUPS** utiliza o protocolo de impressão via internet (IPP) como base para o gerenciamento dos trabalhos de impressão e filas. O processo de impressora de linha ( Line Printer Daemon LPD), o Servidor de Bloco de Mensagens (SMB), e o protocolo AppSocket (a.k.a. JetDirect) também são suportados com funcionalidades redizadas. O CUPS adiciona visualização de impressoras na rede

e a descrição de impressoras PostScript (PPD) e dá suporte para a impressão do mundo real sob o UNIX.

O sítio principal para o CUPS é <http://www.cups.org/>.

---

## 11.7. Correção de Erros

Apos realizar um teste simples com o comando `lpctest(1)`, você pode ter obtido um desses resultados ao invés do funcionamento correto da saída:

Ela funciona por um tempo e para; ou não ejeta a folha toda.

A impressora parece imprimir, mas você aguarda e ela não faz nada. De fato, você deve pressionar o botão de PRINT REMAINING ou ALIMENTAÇÃO de folha para obter algum resultado.

Se esse é o caso, a impressora provavelmente está aguardando para ver se recebe mais algum dado para seu trabalho de impressão antes de imprimir qualquer outra coisa. Para corrigir este problema, você deve ter um filtro de texto para enviar o caracter FORM FEED ( alimentação de folha ) ou qualquer outro necessário para a impressora. Isso é suficiente para a maioria dos casos, pois faz com que a impressora imprima o restante das informações presentes no buffer interno. Isso também é útil para ter certeza que o trabalho de impressão terminou e o próximo irá iniciar em uma nova folha, e não colado com o anterior.

O shell script a seguir imprime o caracter de alimentação de folha na impressora; o arquivo é: `/usr/local/libexec/if-simple`

```
#!/bin/sh
#
if-simple - Simple text input filter for lpd
Installed in /usr/local/libexec/if-simple
#
Simply copies stdin to stdout. Ignores all filter arguments.
Writes a form feed character (\f) after printing job.

/bin/cat && printf "\f" && exit 0
exit 2
```

Ela produz o efeito ``escada."

Você pode ter o seguinte resultado:

```
! "$%&'()*+,-./01234
 "$%&'()*+,-./012345
 "$%&'()*+,-./0123456
```

Você pode vir ser mais uma vítima do *efeito escada*, que é causado pelo conflito de interpretação dos caracteres que deveriam indicar uma nova linha. O estilo

dos sistemas operacionais UNIX® utilizam um único caractere para representar uma nova linha: o caractere ASCII código 10, o alimentação de linha (LF - line feed). O MS-DOS®, OS/2®, e outros sistemas operacionais usam o par de caracteres com os códigos ASCII 10 e 13 ( o caractere de retorno de carro (CR)). Muitas impressoras usam a convenção do MS-DOS para representar uma nova linha.

Quando você imprimir com o FreeBSD, você deverá usar em seu texto o caractere de alimentação de linha, que avança o papel para próxima linha, mas mantém a mesma posição horizontal na página para a impressão do próximo caractere. É isso que o caractere de retorno de carro faz (CR); move para o canto esquerdo a posição da próxima impressão do caractere.

Aqui está o que o FreeBSD espera que sua impressora faça:

A impressora Recebeu CR A impressora Imprimiu CR

A impressora Recebeu LF A impressora imprimiu CR + LF

Aqui, algumas maneiras para solucionar isto:

- Use as chaves de configuração, no painel de controle da impressora para alterar a interpretação desses caracteres. Verifique o manual da sua impressora para encontrar como fazer isso.

**Nota:** Se você utiliza um outro sistema operacional além do FreeBSD, você deve *reconfigurar* a impressora para usar a interpretação dos caracteres CR e LF desse sistema. Você deve dar preferência a outras soluções, como mostrada abaixo.

- Ter um driver serial para FreeBSD converter o caractere LF em CRLF. Claro, ele trabalha *somente* com impressoras na porta serial. Para habilitar essa função, use a opção *ms#* e coloque o modo *onlcr* no arquivo `/etc/printcap`.
- Enviar um *código escape* para a impressora para ter um tratamento temporário diferente do caractere LF. Consulte o manual de sua impressora para conhecer as sequências escape que sua impressora pode suportar. Quando você encontrar o código correto, modifique o filtro texto para enviar este código inicialmente, para após enviar o arquivo a ser impresso.

Aqui tem um exemplo de filtro texto para as impressoras que reconhecem os códigos da linguagem PCL da Hewlett Packard. Esse filtro faz a impressora tratar os caracteres LF como sendo uma sequência de LF + CR; então ele envia o arquivo; então ele envia um caractere FF ( alimentação de formulário ) para ejetar a última página do arquivo. Ele deve funcionar com todas as impressoras da Hewlett Packard.

```
#!/bin/sh
#
hpif - Simple text input filter for lpd for HP-PCL based
printers
Installed in /usr/local/libexec/hpif
#
Simply copies stdin to stdout. Ignores all filter
arguments.
Tells printer to treat LF as CR+LF. Ejects the page
when done.

printf "\033&k2G" && cat && printf "\033&l0H" && exit 0
exit 2
```

Aqui um exemplo do `/etc/printcap` de um computador chamado `orchid`. Ele tem uma única impressora ligada a ele na primeira porta paralela, uma Hewlett Packard LaserJet 3Si, chamada de *teak*. Ele esta usando o script acima como um filtro de texto:

```
#
/etc/printcap for host orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
 :lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak:mx#0:\
 :if=/usr/local/libexec/hpif:
```

A Impressora Sobreescreve as linhas.

A impressora nunca avança para linha abaixo. Todas as linhas do texto são impressas no topo da página e uma sobre as outras.

Esse problema é o "oposto" do efeito escada, descrito acima, e muito mais raro. Em algum lugar, o caracter LF que é usado pelo FreeBSD como final de linha, é tratado como o caracter CR, que retorna a cabeça de impressão para o inicio da linha, mas não move uma linha para baixo, como deveria ser.

Use as chaves de configuração da impressora ou o painel de controle para forçar a seguinte interpretação dos caracteres LF e CR.

### **Impressora Recebe Impressora Imprime**

CR	CR
LF	CR + LF

A Impressora Perde Caracteres.

Enquanto está imprimindo, a impressora não imprime alguns caracteres po linha. O problema pode estar na configuração *raw*, perdendo mais e mais caracteres.

O problema é que a impressora não pode manter a velocidade a qual o computador está usando para enviar os dados pela porta serial ( esse problema não deve ocorrer na porta paralela). Existem duas maneiras de solucionar o problema:

- Se a impressora suportar o controle por XON/XOFF, o FreeBSD irá usa-lo se for especificado o modo *ixon* na opção *ms#*.

- Se a impressora suportar *carrier flow control*, especifique o modo *crtcts* na opção *ms#*. Tenha certeza que o cabo que conecta a impressora ao computador esteja corretamente feito para *carrier flow control*,

### A Impressora Imprime Lixo.

A impressora aparentemente imprime lixo randômico e não o texto solicitado.

Esse é outro sintoma de parâmetros incorretos da comunicação com a impressora serial. Verifique novamente os valores nas opções da taxa bps (*br*) e da paridade (*ms#*); tenha certeza das configurações especificadas no arquivo `/etc/printcap`.

### Nada Acontece.

Se nada acontecer, o problema, provavelmente, está dentro do FreeBSD e não no *hardware*. Adicione a opção (*lf*) no arquivo `/etc/printcap` na impressora que você está verificando. No nosso exemplo, estamos utilizando a impressora *rattan*, com a opção *lf*:

```
rattan|line|diablo|lp|Diablo 630 Line Printer:\
:sh:sd=/var/spool/lpd/rattan:\
:lp=/dev/lpt0:\
:if=/usr/local/libexec/if-simple:\
:lf=/var/log/rattan.log
```

Então, tente novamente. Verifique o arquivo de *log* ( no nosso exemplo, `/var/log/rattan.log`) para ver qualquer mensagem de erro que possa aparecer. Baseado nessas mensagens, você poderá corrigir o problema.

Se você não especificar a opção *lf* o **LPD** usará `/dev/console` como padrão.

---

## Capítulo 12. Armazenamento

### Índice

- 12.1. [Sinótese](#)
- 12.2. [Nomes de Dispositivos](#)
- 12.3. [Adicionando Discos](#)
- 12.4. [RAID](#)
- 12.5. [Criando e utilizando mídias ópticas \(CDs & DVDs\)](#)
- 12.6. [Criação e utilização de disquetes](#)
- 12.7. [Criando e utilizando fitas DAT](#)
- 12.8. [Backups em Disquetes](#)
- 12.9. [Considerações básicas sobre backup](#)
- 12.10. [Rede, Memória, e Sistemas de Arquivos em Imagens](#)
- 12.11. [Snapshots de sistemas de arquivos](#)
- 12.12. [Cotas de uso para sistemas de arquivos](#)
- 12.13. [Encryptando partições de um disco](#)

## 12.1. Sinótese

Este capítulo demonstra como fazer uso de discos no FreeBSD. Incluindo discos de memória, discos conectados através da rede e os dispositivos de armazenamento dos padrões SCSI/IDE.

Após ler este capítulo você saberá:

- A terminologia utilizada pelo FreeBSD para descrever a organização dos dados num disco rígido (partições e slices).
- Como adicionar um disco rígido ao seu sistema.
- Como configurar um sistema de arquivos virtual, como os alocados em memória.
- Como fazer utilização de cotas para limitar o uso do espaço de armazenamento em disco.
- Como criptografar discos para assegurar contra possíveis ataques.
- Como criar e gravar CDs e DVDs no FreeBSD.
- As várias opções de mídias para cópias de segurança (Backup).
- Como utilizar os programas disponíveis no FreeBSD para realizar cópias de segurança (Backup).
- Como fazer cópias de segurança em disquetes.
- O que são *snapshots* e como fazer uso deles de maneira eficiente.

---

## 12.2. Nomes de Dispositivos

Temos aqui uma lista dos dispositivos físicos de armazenamento suportados pelo FreeBSD, e os nomes associados a eles.

**Tabela 12-1. Convenções de Nomenclatura de Discos Físicos**

<b>Tipo de Controladora</b>	<b>Nome do Dispositivo</b>
Discos rígidos IDE	<i>ad</i>
CDROM IDE	<i>acd</i>
Discos rígidos SCSI e dispositivos de armazenamento em massa USB	<i>da</i>
CDROM SCSI	<i>cd</i>
Controladores de CDROM fora de padrões	<i>mcd</i> para CD-ROM Mitsumi, <i>scd</i> para CD-ROM Sony, <i>matcd</i> para CD-ROM Matsushita/Panasonic <a href="#">[a]</a>
Controladoras de disquete	<i>fd</i>
Controladoras de tape SCSI	<i>sa</i>
Controladoras de tape IDE	<i>ast</i>



<b>Tipo de Controladora</b>	<b>Nome do Dispositivo</b>
Controladoras de Flash	<i>fla</i> para dispositivos de Flash DiskOnChip®
Controladoras RAID	<i>aacd</i> para Adaptec® AdvancedRAID, <i>mlx</i> e <i>mlyd</i> para Mylex®, <i>amrd</i> para AMI MegaRAID®, <i>idad</i> para Compaq Smart RAID, <i>twed</i> para 3ware® RAID.
Notas:	
a. O suporte a controladora <a href="#">mated(4)</a> foi removido no FreeBSD 4.X desde 5 de outubro de 2002 e não é suportada no FreeBSD 5.0 e 5.1. Entretanto, o suporte foi novamente adicionado no FreeBSD 5.X desde 16 de junho de 2003.	

## 12.3. Adicionando Discos

*Contribuição original de David O'Brien.*

Vamos supor que desejamos adicionar um novo disco SCSI numa máquina que atualmente possui apenas um. Primeiro desligamos o computador e conectamos o novo disco ao computador seguindo as instruções fornecidas pelos fabricantes do computador, da controladora e do disco. Devido a este procedimento variar bastante, os detalhes desta operação estarão fora do escopo deste documento.

Realize login como usuário `root`. Após você ter instalado o disco, verifique o arquivo `/var/run/dmesg.boot` para certificar-se de que o novo disco foi encontrado. Continuando com o nosso exemplo, o novo disco adicionado será o `da1` e nós vamos montá-lo em `/1` (se você estiver adicionando um disco IDE, o nome do dispositivo será `wd1` em sistemas pre-4.0, ou `ad1` ma maioria dos sistemas 4.X).

Devido ao FreeBSD rodar em computadores compatíveis com IBM-PC, ele terá que respeitar o particionamento especificado pela BIOS do PC. Estas por sua vez são diferentes das partições tradicionais BSD. Um disco de um PC pode ter quatro entradas de partições na BIOS. Se o disco será totalmente dedicado ao FreeBSD você terá a opção de usar o modo *dedicated*. Ou então, o FreeBSD terá de viver como uma das partições da BIOS do PC. O FreeBSD chama as partições da BIOS do PC de *slices* ou *fatias*, então não as confunda com as partições BSD tradicionais. Você também pode usar *slices* num disco dedicado ao FreeBSD, mas usá-los em um computador que possua um outro sistema operacional instalado. Isso para não causar confusões com o utilitário `fdisk` de outros sistemas operacionais.

No caso do slice a unidade será adicionada como `/dev/dals1e`. Isto é lido como: disco SCSI, unidade número 1 (segundo disco SCSI), slice 1 (Partição 1 da BIOS do PC), e `e` é uma partição BSD. Em um caso dedicado, a unidade será adicionada simplesmente como `/dev/dale`.

## 12.3.1. Utilizando o sysinstall(8)

### 1. Navegando no Sysinstall

Você pode utilizar o `/stand/sysinstall` para particionar e definir os rótulos ou *label* que serão utilizados no novo disco de maneira fácil através de seus menus. Estando logado como usuário `root` ou utilizando o comando `su`. Execute `/stand/sysinstall` e entre no menu *Configure*. Dentro do *FreeBSD Configuration Menu*, vá descendo nas opções e selecione a opção *Fdisk*.

### 2. fdisk Editor de Partições

Uma vez no **fdisk**, nós podemos teclar `A` para usar todo o disco para o FreeBSD. Quando for perguntado se você quer "manter a possibilidade de uso cooperativo com outro sistema operacional no futuro" responda *YES*. Grave as alterações feitas no disco teclando `w`. Agora saia do FDISK teclando `q`. Em seguida, você será perguntado sobre o Registro Mestre de Inicialização(MBR, Master Boot Record). Desde que você esteja adicionando um disco num sistema em execução, selecione *None*.

### 3. Editor de Rótulo(Label) de Disco

Em seguida, você precisará sair do **sysinstall** e iniciá-lo novamente. Siga a mesma direção citada nos menus, porém agora selecione a opção *Label*. Você entrará então no *Disk Label Editor* (Editor de Rótulos de Discos). Onde são criadas as partições BSD tradicionais. Um disco pode ter oito partições, rotuladas de *a-h*. Alguns rótulos de partições tem um uso especial. A partição *a* é utilizada para a partição raiz ou root (`/`). Somente o disco do sistema (ex, o disco de onde seu sistema realiza o boot) deve ter uma partição *a*. A partição *b* é utilizada para partições de memória virtual(swap), e você pode ter vários discos com partições swap. A partição *c* endereça todo o disco no modo dedicado, ou toda a slice do FreeBSD no modo slice. As outras partições são para uso geral.

O editor de rótulos do **sysinstall** prefere usar a partição *e* quando esta não for uma partição raiz ou swap. Uma vez no Editor de Rótulos, crie um sistema de arquivos único teclando `c`. Quando perguntado se este será um Sistema de Arquivos (FS, file system) ou swap, escolha *FS* e digite um ponto de montagem (ex, `/mnt`). Quando você estiver adicionando um disco após a instação do sistema, o **sysinstall** não criará as entradas no `/etc/fstab` para você, então o ponto de montagem definido não tem muita importância.

Você agora está pronto para gravar o novo rótulo no disco e criar o sistema de arquivos. Faça isso teclando `w`. Ignore qualquer erro do **sysinstall** que não poderá montar a nova partição. Saia do Editor de Rótulos e do **sysinstall**.

## 4. Finalizando

Como último passo, edite o `/etc/fstab` para adicionar a entrada referente ao novo disco.

### 12.3.2. Utilizando utilitários de linha de comando

#### 12.3.2.1. Utilizando Slices ou Fatias

Esta configuração permitirá que ao seu disco funcionar corretamente com um outro sistema operacional que possa estar instalado em seu computador e não se confundirá com o utilitário `fdisk` de outros sistemas. É recomendado o uso deste método para instalar novos discos. Somente use o modo *dedicated* se você tiver uma boa razão!

```
dd if=/dev/zero of=/dev/dal bs=1k count=1
fdisk -BI dal #Inicializa seu novo disco
disklabel -B -w -r dals1 auto #Rotula ele
disklabel -e dals1 #Edita o rótulo do disco somente para criar e
adicionar alguma partição
mkdir -p /l
newfs /dev/dals1e #Repita isso em cada uma das partições criadas
mount /dev/dals1e /l #Monta a partição
vi /etc/fstab #Adiciona uma entrada apropriada no seu /etc/fstab.
```

Se você possuir um disco IDE, substitua o `ad` por `da`. Nos sistemas pre-4.X use `wd`.

#### 12.3.2.2. Dedicado

Caso você não vá compartilhar este novo disco com outros sistemas operacionais, você poderá usar o modo *dedicated* ou *dedicado*. Lembre-se que este modo pode causar confusão nos sistemas operacionais Microsoft; entretanto, nenhum dano será causado por ele. O OS/2® da IBM, entretanto, irá tornar qualquer partição que ele não entenda numa partição "apropriada".

```
dd if=/dev/zero of=/dev/dal bs=1k count=1
disklabel -Brw dal auto
disklabel -e dal #Cria a partição `e'
newfs -d0 /dev/dale
mkdir -p /l
vi /etc/fstab #Adicione a entrada para /dev/dale
mount /l
```

Um método alternativo é:

```
dd if=/dev/zero of=/dev/dal count=2
disklabel /dev/dal | disklabel -BrR dal /dev/stdin
newfs /dev/dale
mkdir -p /l
vi /etc/fstab #Adicione a entrada para /dev/dale
mount /l
```

**Nota:** Desde o FreeBSD 5.1-RELEASE, o utilitário [bsdlable\(8\)](#) substitui o velho programa [disklabel\(8\)](#). No [bsdlable\(8\)](#) algumas opções e parâmetros obsoletos foram retirados; no exemplo anterior temos a opção `-r` que deve ser removida com o [bsdlable\(8\)](#). Para ter mais informações, por favor, visite a página de manual do [bsdlable\(8\)](#).

---

## 12.4. RAID

### 12.4.1. RAID feito em Software

#### 12.4.1.1. Configuração do Controlador de Discos Concatenados(CCD)

*Trabalho original de Christopher Shumway. Revisado por Jim Brown.*

Quando estiver escolhendo uma solução de armazenamento em massa os fatores mais importantes a considerar são a velocidade, a confiança e o custo. É raro ter todos os três num equilíbrio; normalmente um rápido e confiável dispositivo de armazenamento em massa é caro, e para baixar o preço tanto a velocidade como a confiança do dispositivo podem ser sacrificadas.

No desenvolvimento do sistema descrito abaixo, o custo é escolhido como o fator mais importante, seguido da velocidade e então a confiança. A velocidade da transferência de dados deste sistema determinada é relacionada pela rede. E enquanto a confiança é muito importante, o controlador CCD descrito abaixo disponibiliza dados que já tiveram cópias de segurança geradas em CD-Rs e que podem facilmente ser substituídos.

Definir seus próprios requerimentos é o primeiro passo na escolha de uma solução de armazenamento em massa. Se suas necessidades preferirem a velocidade e a confiança ao invés do custo, sua solução irá ser diferente do sistema descrito nesta seção.

##### 12.4.1.1.1. Instalando o Hardware

Em relação ao sistema de discos IDE, temos três discos IDE Western Digital 30GB de 5400 RPM para o núcleo do disco CCD descrito abaixo, disponibilizando aproximadamente 90GB de armazenamento. De maneira ideal, cada disco IDE teria sua própria controladora e cabo, mas para minimizar custos não utilizaremos controladoras IDEs adicionais. Uma vez que os discos são configurados com jumpers, temos que cada controladora IDE possui um mestre(master) e um escravo(slave).

Ao reiniciar, a BIOS do sistema terá que detectar automaticamente os discos conectados. Ainda mais importante, o FreeBSD terá que detectar os discos ao reiniciar:

```
ad0: 19574MB <WDC WD205BA> [39770/16/63] at ata0-master UDMA33
ad1: 29333MB <WDC WD307AA> [59598/16/63] at ata0-slave UDMA33
ad2: 29333MB <WDC WD307AA> [59598/16/63] at ata1-master UDMA33
ad3: 29333MB <WDC WD307AA> [59598/16/63] at ata1-slave UDMA33
```

**Nota:** Se o FreeBSD não detectar todos os discos, confira se os discos foram jumpeados corretamente. A maioria das controladores também possuem a opção de jumpeamento "Cable Select". Este *não* é o jumper para a relação master/slave. Consulte a documentação da controladora para ter ajuda de como identificar a maneira correta de configurar.

Em seguida, consideraremos como serão conectados como parte do sistema de arquivos. Você deve realizar uma pesquisa entre o [vinum\(8\)](#) ([Capítulo 13](#)) e o [ccd\(4\)](#). Nesta configuração em particular, o [ccd\(4\)](#) foi o escolhido.

#### 12.4.1.1.2. Configurando o CCD

O controlador [ccd\(4\)](#) permite que você pegue diversos discos iguais e concatenê-os em um sistema de arquivos lógico. Para fazer o uso do [ccd\(4\)](#) você precisará de um kernel contruído com o suporte ao [ccd\(4\)](#). Adicione esta linha no seu arquivos de configuração do kernel, recompile e reinstale o kernel:

```
pseudo-device ccd 4
```

Nos sistemas 5.X você deve usar a linha à seguir:

```
device ccd
```

**Nota:** No FreeBSD 5.X não é necessário especificar um número de dispositivos [ccd\(4\)](#), como o dispositivo do controlador [ccd\(4\)](#) agora se auto-replica -- novas instâncias do dispositivo serão criadas automaticamente sobre demanda.

O suporte ao [ccd\(4\)](#) pode também ser carregado como um módulo de kernel carregável no FreeBSD 3.0 ou anterior.

Para configurar o [ccd\(4\)](#) você deve primeiro usar o [disklabel\(8\)](#) para rotular os discos:

```
disklabel -r -w ad1 auto
disklabel -r -w ad2 auto
disklabel -r -w ad3 auto
```

Isto criará um rótulo de disco para o `ad1c`, `ad2c` e `ad3c` que farão uso do disco todo.

**Nota:** Desde o FreeBSD 5.1-RELEASE, o utilitário [bsdlable\(8\)](#) substitui o antigo programa [disklabel\(8\)](#). No [bsdlable\(8\)](#) algumas opções e parâmetros obsoletos foram retirados; no exemplo que segue temos a opção `-r` que deve ser retirada. Para mais informações, visite a página de manual do [bsdlable\(8\)](#).

O próximo passo é mudar o tipo do rótulo. Você pode usar o [disklabel\(8\)](#) para editar os discos:

```
disklabel -e ad1
disklabel -e ad2
disklabel -e ad3
```

Isto abre o rótulo atual de cada disco com o editor especificado pela variável de ambiente `EDITOR`, que geralmente é o [vi\(1\)](#).

Um rótulo de disco não modificado parecerá com algo como isto:

```
8 partitions:
size offset fstype [fsize bsize bps/cpg]
 c: 60074784 0 unused 0 0 0 # (Cyl. 0 -
59597)
```

Adicione uma nova partição *e* para que o [ccd\(4\)](#) use. Isto pode usualmente ser copiado da partição *c*, mas a opção *fstype* tem de ser `4.2BSD`. O rótulo do disco agora deve parecer com algo como isto:

```
8 partitions:
size offset fstype [fsize bsize bps/cpg]
 c: 60074784 0 unused 0 0 0 # (Cyl. 0 -
59597)
 e: 60074784 0 4.2BSD 0 0 0 # (Cyl. 0 -
59597)
```

#### 12.4.1.1.3. Construindo o Sistema de Arquivos

O dispositivo `ccd0c` pode não existir ainda, para criá-lo execute os seguintes comandos:

```
cd /dev
sh MAKEDEV ccd0
```

**Nota:** No FreeBSD 5.0, o [devfs\(5\)](#) gerenciará automaticamente os dispositivos listados em `/dev`, o que dispensa o uso do `MAKEDEV`.

Agora que você tem todos os discos rotulados você deve contruir o [ccd\(4\)](#). Para fazer isso, use o [ccdconfig\(8\)](#) com opções similares as que seguem:

```
ccdconfig ccd0 ① 32 ② 0 ③ /dev/ad1e ④ /dev/ad2e /dev/ad3e
```

O uso e o significado de cada opção está listado abaixo:

①

O primeiro argumento é o dispositivo que será configurado, neste caso, o `/dev/ccd0c`. A porção `/dev/` é opcional.

②

O intercalamento para o sistema de arquivos. O intercalamento define o tamanho da faixa(stripe) em blocos de disco, que é normalmente 512 bytes. Ebntão, um intercalamento de 32 seria 16,384 bytes.

③

Parâmetros para o [ccdconfig\(8\)](#). Se você deseja ativar o controle de espelhamento você pode especificar o parâmetro aqui. Esta configuração não provê espelhamento para o [ccd\(4\)](#), então esta está sendo definida por 0 (zero).

#### 4

As argumentos finais para o [ccdconfig\(8\)](#) são os dispositivos para colocar na matriz. Utilize o caminho e nome completo para cada dispositivo.

Após executar o [ccdconfig\(8\)](#) o [ccd\(4\)](#) é configurado. Um sistema de arquivo pode ser instalado. Use o [newfs\(8\)](#) para mais opções, ou simplesmente execute:

```
newfs /dev/ccd0c
```

#### 12.4.1.1.4. Fazendo tudo automaticamente

Geralmente, você irá querer montar o [ccd\(4\)](#) em cada reinício do sistema. Para fazer isso você deve configurar isso primeiro. Escreva sua configuração corrente no `/etc/ccd.conf` utilizando os seguintes comandos:

```
ccdconfig -g > /etc/ccd.conf
```

Durante o reinício do sistema o script `/etc/rc` executa `ccdconfig -C` caso exista o `/etc/ccd.conf`. Isto automaticamente configura o [ccd\(4\)](#) e então este poderá ser montado.

**Nota:** Se você estiver inicializando o sistema em modo monousuário, antes você poder dar [mount\(8\)](#) no [ccd\(4\)](#), você precisa definir os seguintes comandos para configurar a matriz:

```
ccdconfig -C
```

Para montar automaticamente o [ccd\(4\)](#) coloque uma entrada para o [ccd\(4\)](#) em `/etc/fstab` e então este será montado no início do sistema:

```
/dev/ccd0c /media ufs rw 2 2
```

#### 12.4.1.2. O gerenciador de volumes Vinum

O gerenciador de volumes Vinum é um controlador de dispositivo de bloco que implementa controladores de discos virtuais. Este isola o disco físico da interface de dispositivos de bloco e mapeia os dados em meios que resultam num aumento de flexibilidade, performance e confiança comparáveis as tradicionais slices de armazenamento em disco. O [vinum\(8\)](#) implementa os modelos RAID-0, RAID-1 e RAID-5, tanto individualmente como combinando.

Visite [Capítulo 13](#) para mais informações sobre [vinum\(8\)](#).

## 12.4.2. RAID feito em Hardware

O FreeBSD também suporta uma variedade de controladoras RAID. Estes dispositivos controlam um subsistema RAID sem precisar de um software específico do FreeBSD para gerenciar a matriz.

Utilizando uma BIOS na placa, a placa controla a maioria das operações em disco. À seguir temos uma configuração descrevendo o uso de uma controladora Promise IDE RAID. Quando esta placa é instalada e o sistema é iniciado, é apresentado um prompt pedindo informações. Siga as instruções para entrar na tela de configuração da placa. Aqui, você tem a habilidade para combinar todos os discos conectados. Após ter feito a configuração os disco parecerão um único disco para o FreeBSD. Outros níveis de RAID podem ser configurados como descrito.

## 12.4.3. Reconstruindo Matrizes de ATA RAID1

O FreeBSD permite que você troque um disco falhando em uma matriz. Isto requer que você pegue-o antes de reiniciar o sistema.

Você provavelmente verá algo como o seguinte em `/var/log/messages` ou na saída do [dmesg\(8\)](#):

```
ad6 on monster1 suffered a hard error.
ad6: READ command timeout tag=0 serv=0 - resetting
ad6: trying fallback to PIO mode
ata3: resetting devices .. done
ad6: hard error reading fsbn 1116119 of 0-7 (ad6 bn 1116119; cn 1107
tn 4 sn 11) status=59 error=40
ar0: WARNING - mirror lost
```

Utilizando o [atacontrol\(8\)](#), busque por informações:

```
atacontrol list
ATA channel 0:
 Master: no device present
 Slave: acd0 <HL-DT-ST CD-ROM GCR-8520B/1.00> ATA/ATAPI rev 0

ATA channel 1:
 Master: no device present
 Slave: no device present

ATA channel 2:
 Master: ad4 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
 Slave: no device present

ATA channel 3:
 Master: ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
 Slave: no device present

atacontrol status ar0
ar0: ATA RAID1 subdisks: ad4 ad6 status: DEGRADED
```

1. Você primeiramente precisará desconectar o disco da matriz para então poder removê-lo seguramente:



2. # atacontrol detach 3
  3. Substitua o disco.
  4. Reconecte o disco como um reserva:
  5. # atacontrol attach 3
  6. Master: ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
  7. Slave: no device present
  8. Reconstrua a matriz:
  9. # atacontrol rebuild ar0
  10. O comando de reconstrução executará até estar completo. Entretanto, é possível abrir outro terminal (utilizando **Alt+Fn** ) e checar o progresso da reconstrução com o seguinte comando:
  11. # dmesg | tail -10
  12. [output removed]
  13. ad6: removed from configuration
  14. ad6: deleted from ar0 disk1
  15. ad6: inserted into ar0 disk1 as spare
  - 16.
  17. # atacontrol status ar0
  18. ar0: ATA RAID1 subdisks: ad4 ad6 status: REBUILDING 0% completed
  19. Espere até que esta operação esteja completa.
- 

## 12.5. Criando e utilizando mídias ópticas (CDs & DVDs)

*Contribuição de Mike Meyer.*

### 12.5.1. Introdução

Os CDs tem um número diferenciado de funcionalidades dos discos convencionais. Inicialmente, eles não são graváveis aos usuários. Eles são criados o que permite serem lidos continuamente sem esperas para mover o cabeçote entre trilhas. Eles também são facilitadores no transporte de informações entre sistemas com tamanhos de mídias similares e podendo ser feitas ao mesmo tempo.

Os CDs tem trilhas, mas isto se refere a uma seção de dados que são lidos continuamente e não são uma propriedade física do disco. Para produzir um CD no FreeBSD, você deve preparar os arquivos de dados que comporão as trilhas, para que então sejam escritas como trilhas do CD.

O sistema de arquivos ISO 9660 foi desenvolvido para criar um acordo nestas diferenças. Isto infelizmente codifica o sistema de arquivos com limites inerentes ao padrão. Afortunadamente, este provê um mecanismo de extensões que permite gravar CDs apropriadamente para exceder estes limites enquanto estiver trabalhando com sistemas que que não suportam estas extensões.

O programa [sysutils/mkisofs](#) é utilizado para produzir um arquivo contendo um sistema de arquivos ISO 9660. Este possui opções que suportam diversas extensões, estas estão descritas abaixo. Você pode instalar o port [sysutils/mkisofs](#).

Qual utilitário usar para gravar o CD dependerá se seu gravador de CD é ATAPI ou algum outro. Gravadores de CD ATAPI usam o programa [burncd](#) que é uma parte do sistema. Os gravadores SCSI e USB devem usar o [cdrecord](#) do port [sysutils/cdrtools](#).

`burncd` has a limited number of supported drives. To find out if a drive is supported, see the [CD-R/RW supported drives](#) list.

**Nota:** Se você usa a versão FreeBSD 5.X, FreeBSD 4.8-RELEASE ou acima, será possível utilizar o [cdrecord](#) e outros utilitários para controladores SCSI em controladores ATAPI com o [ATAPI/CAM module](#).

## 12.5.2. mkisofs

O [sysutils/mkisofs](#) produz um sistema de arquivos ISO 9660 que é uma imagem da árvore de diretórios no espaço de nomes do sistema de arquivos UNIX®. A maneira mais simples de usá-lo é:

```
mkisofs -o imagefile.iso /path/to/tree
```

Este comando criará o arquivo *imagefile.iso* contendo um sistema de arquivos ISO 9660 que é uma cópia da árvore de diretórios e arquivos em */path/to/tree*. No processo, serão mapeados os nomes de arquivos de acordo com as limitações do padrão ISO 9660, e serão excluídos os arquivos que tiverem nomes não característicos de sistemas de arquivos ISO.

Algumas opções estão disponíveis para exceder as restrições. Em particular, a opção *-R* ativa as extensões Rock Ridge, que são comuns aos sistemas UNIX, a opção *-J* ativa o uso de extensões Joliet, que são utilizadas em sistemas da Microsoft, e a opção *-hfs* pode ser utilizada na criação de sistemas de arquivos HFS, que são utilizados no Mac OS®.

Para CDs que vão ser utilizados apenas em sistemas FreeBSD a opção *-U* pode ser utilizada para desativar todas as limitações de nomes de arquivos. Quando utilizada com a opção *-R*, esta produz um sistema de arquivos idêntico à árvore de diretórios do FreeBSD que você utilizou na criação da imagem. Porém isso viola de diversas maneiras o padrão ISO 9660.

A última opção de uso geral é a *-b*. Ela é utilizada para especificar o local de uma imagem de boot que será utilizada na geração de um CD bootável "El Torito". Esta opção pede um argumento que é o caminho da imagem de boot do topo da árvore de diretórios que será escrita no CD. Então, fornecer `/tmp/myboot` cria um sistema FreeBSD bootável com a imagem de boot em `/tmp/myboot/boot/cdboot`, você poderia produzir a imagem de um sistema de arquivos ISO 9660 em `/tmp/bootable.iso` como:

```
mkisofs -U -R -b boot/cdboot -o /tmp/bootable.iso /tmp/myboot
```

Após ter feito o indicado, se você possuir um dispositivo `vn` (FreeBSD 4.X), ou `md` (FreeBSD 5.X) configurado em seu kernel, você poderá montar o sistema de arquivos com:

```
vnconfig -e vn0c /tmp/bootable.iso
mount -t cd9660 /dev/vn0c /mnt
```

Isso no FreeBSD 4.X, já no FreeBSD 5.X faça:

```
mdconfig -a -t vnode -f /tmp/bootable.iso -u 0
mount -t cd9660 /dev/md0 /mnt
```

Até agora você pode verificar que o `/mnt` e o `/tmp/myboot` são idênticos.

Existem diversas outras opções que podem ser utilizadas com o [sysutils/mkisofs](#) para ajustar seu comportamento. Em particular, modificações para um layout ISO 9660 e na criação de discos Joliet e HFS. Visite a página de manual do [mkisofs\(8\)](#) para ter mais detalhes.

### 12.5.3. burncd

Se você tiver um gravador de CD ATAPI, você poderá utilizar o comando `burncd` para gravar uma imagem ISO num CD. O `burncd` é parte do sistema, é instalado em `/usr/sbin/burncd`. Usá-lo é muito simples, e este possui poucas opções:

```
burncd -f cddevice data imagefile.iso fixate
```

Isto irá gravar uma cópia do `imagefile.iso` no `cddevice`. O dispositivo padrão é o `/dev/acd0c`. Visite a página de manual do [See burncd\(8\)](#) para ter mais opções de como definir a velocidade de gravação, para ejetar o CD após a gravação e para gravar CDs de áudio.

### 12.5.4. cdrecord

Se você não possuir um gravador de CDs ATAPI, você terá que usar o `cdrecord` para gravar seus CDs. O `cdrecord` não faz parte do sistema; você deve instalá-lo através do port [sysutils/cdrtools](#) ou pelo pacote apropriado. Mudanças no sistema podem fazer que algumas versões binárias deste programa venham resultar em falhas. Você deve atualizar este port quando atualizar o seu sistema, ou se você já estiver [tracking - STABLE](#), atualize o port quando uma nova versão estiver disponível.

Enquanto o `cdrecord` tem diversas opções, o uso básico dele é mais simples do que o uso do `burncd`. Para gravar uma imagem ISO 9660 basta:

```
cdrecord dev=device imagefile.iso
```

O segredo da utilização do `cdrecord` é encontrar a opção *dev* que deve ser utilizada. Para encontrar a configuração correta, use o parâmetro *-scanbus* do `cdrecord`, que produzirá um resultado parecido com este:

```
cdrecord -scanbus
Cdrecord 1.9 (i386-unknown-freebsd4.2) Copyright (C) 1995-2000 Jörg
Schilling
Using libscg version 'schily-0.1'
scsibus0:
 0,0,0 0) 'SEAGATE ' 'ST39236LW ' '0004' Disk
 0,1,0 1) 'SEAGATE ' 'ST39173W ' '5958' Disk
 0,2,0 2) *
 0,3,0 3) 'iomega ' 'jaz 1GB ' 'J.86' Removable
Disk
 0,4,0 4) 'NEC ' 'CD-ROM DRIVE:466' '1.26' Removable
CD-ROM
 0,5,0 5) *
 0,6,0 6) *
 0,7,0 7) *
scsibus1:
 1,0,0 100) *
 1,1,0 101) *
 1,2,0 102) *
 1,3,0 103) *
 1,4,0 104) *
 1,5,0 105) 'YAMAHA ' 'CRW4260 ' '1.0q' Removable
CD-ROM
 1,6,0 106) 'ARTEC ' 'AM12S ' '1.06' Scanner
 1,7,0 107) *
```

Esta listagem demonstra qual o valor apropriado que a opção *dev* deverá ter. Localize seu gravador de CDs, e use os três números separados por vírgulas como valor para a opção *dev*. Neste caso, o dispositivo CRW é o 1,5,0, então o valor apropriado seria *dev=1,5,0*. Há maneiras mais fáceis de especificar este valor; visite a página de manual do [cdrecord\(1\)](#) para mais detalhes. Que é onde estão descritas as informações de como gravar trilhas de áudio, controlar a velocidade entre outras possibilidades.

### 12.5.5. Duplicação de CDs de Áudio

Você pode duplicar um CD de áudio extraíndo seus dados do CD em uma série de arquivos, e então gravar estes arquivos num CD virgem. Este processo é um pouco diferente entre controladores ATAPI e SCSI.

## Controladores SCSI

1. Utilize o `cdda2wav` para extrair o áudio.
2. `% cdda2wav -v255 -D2,0 -B -Owav`
3. Utilize o `cdrecord` para gravar os arquivos `.wav`.
4. `% cdrecord -v dev=2,0 -dao -useinfo *.wav`

Certifique-se que o `2,0` é a configuração apropriada como descrito em [Seção 12.5.4](#).

## Controladores ATAPI

1. A controladora de CD ATAPI torna cada trilha disponível como `/dev/acd0t $nn$` , onde  $d$  é o número da controladora, e  $nn$  é o número da trilha escrita em dois dígitos decimais, precedidos com zero quando preciso. Então a primeira trilha no disco é a `/dev/acd0t01`, a segunda é a `/dev/acd0t02`, a terceira é a `/dev/acd0t03`, e assim segue.

Certifique-se de que os arquivos apropriados existam em `/dev`.

```
cd /dev
sh MAKEDEV acd0t99
```

**Nota:** No FreeBSD 5.0, [devfs\(5\)](#) criará automaticamente e gerenciará as entradas em `/dev` para você, portanto, não é necessário utilizar o `MAKEDEV`.

2. Extraia cada trilha utilizando o [dd\(1\)](#). Você deve especificar também um tamanho específico de bloco quando estiver extraindo os arquivos.
3. `# dd if=/dev/acd0t01 of=track1.cdr bs=2352`
4. `# dd if=/dev/acd0t02 of=track2.cdr bs=2352`
5. ...
6. Grave os arquivos extraídos para o disco utilizando o `burncd`. Você deve especificar que são arquivos de áudio, e que o `burncd` deve executar o *fixate* no disco quando terminar a gravação.
7. `# burncd -f /dev/acd0c audio track1.cdr track2.cdr ... fixate`

### 12.5.6. Duplicando CDs de Dados

Você pode copiar um CD de dados em um arquivo de imagem que funcionalmente é equivalente a uma imagem criada com o [sysutils/mkisofs](#), e você usar este recurso para duplicar qualquer CD de dados. O exemplo a seguir assume que o seu dispositivo de CDROM é o `acd0`. Substitua pelo dispositivo de CDROM correto no seu sistema. Um `c` deve ser adicionado ao final do nome do dispositivo para indicar a partição toda, ou, no caso de CDROMs, o disco inteiro.

```
dd if=/dev/acd0c of=file.iso bs=2048
```

Agora que você possui a imagem, você pode gravá-la num CD como descrito anteriormente.

## 12.5.7. Utilizando CDs de Dados

Agora que você criou um CD de dados padrão, você provavelmente irá querer montar a unidade e ler dados dele. Por padrão, o [mount\(8\)](#) irá assumir que o tipo de sistema de arquivos é o *ufs*. Se você tentar algo como:

```
mount /dev/cd0c /mnt
```

Você conseguirá a seguinte mensagem ``Incorrect super block'', e o disco não será montado. O CDROM não é um sistema de arquivos *UFS*, portanto, tentar montá-lo com esta configuração ocasionará em falha. Você só precisa indicar ao [mount\(8\)](#) que o sistema de arquivos é do tipo *ISO9660*, e tudo funcionará. Você faz isso especificando a opção *-t cd9660* ao [mount\(8\)](#). Por exemplo, se você quer montar o dispositivo de CDROM `/dev/cd0c`, sob o `/mnt`, você deve executar:

```
mount -t cd9660 /dev/cd0c /mnt
```

Note que seu nome de dispositivo (`/dev/cd0c` neste exemplo) poderá ser diferente, dependendo da interface que seu CDROM utilizar. Também, a opção *-t cd9660* apenas executa o [mount cd9660\(8\)](#). O exemplo acima pode ser reduzido para:

```
mount_cd9660 /dev/cd0c /mnt
```

Você geralmente pode utilizar CDROMs de dados de qualquer fabricante com este método. Entretanto, discos com algumas extensões do ISO 9660 podem ter um comportamento ímpar. Por exemplo, disco com extensão Joliet armazenam todos os nomes de arquivos em caracteres Unicode de dois bytes. O kernel do FreeBSD não entende Unicode (ainda!), então os caracteres que não pertencerem ao idioma Inglês serão mostrados como sinais de interrogação. (Se você estiver rodando o FreeBSD 4.3 ou superior, o driver CD9660 inclui ganchos para carregar uma tabela de conversão Unicode dinamicamente. Módulos para algumas das codificações comuns estão disponíveis no port [sysutils/cd9660\\_unicode](#).)

Ocasionalmente, você poderá receber um ``Device not configured'' quando tentar montar um CDROM. Isto geralmente significa que o drive de CDROM acha que não há um disco, ou este drive não é visível no barramento. Pode levar alguns segundos para que um drive de CDROM detectar que está com um disco, portanto seja paciente.

Algumas vezes, um CDROM SCSI pode não ser inicializado devido a não haver tempo suficiente para responder durante um reinício do barramento. Se você possui um CDROM SCSI, por favor, adicione a seguinte opção ao seu arquivo de configuração do kernel e e [recompile-o](#).

```
options SCSI_DELAY=15000
```

Isto diz para o barramento SCSI pausar durante 15 segundos durante a inicialização, para dar chance de resposta ao seu drive de CDROM durante o reinício do barramento SCSI.

## 12.5.8. Gravando CDs de Dados em modo Raw

Você pode querer gravar um arquivo diretamente no CD, sem criar um sistema de arquivos ISO 9660. Algumas pessoas fazem isso no propósito de backup. E este modo de gravação é mais rápido:

```
burncd -f /dev/acd1c -s 12 data archive.tar.gz fixate
```

Para poder acessar os dados que serão gravados no CD você deverá ler os dados de um dispositivo raw:

```
tar xzvf /dev/acd1c
```

Você não pode montar este disco como faria com um CDROM normal. Como um CDROM não poderia ser lido sob nenhum outro sistema operacional, excepto o FreeBSD. Se você quer estar apto a montar os CDs, ou a compartilhar os dados com outros sistemas operacionais, você deve usar o [sysutils/mkisofs](#) como já citado.

## 12.5.9. Utilizando o Driver ATAPI/CAM

Este driver permite que dispositivos ATAPI (CD-ROM, CD-RW, DVD-ROM, etc...) possam ser acessados através do sub-sistema SCSI, permitindo o uso de aplicações como o [sysutils/cdrdao](#) ou o [cdrecord\(1\)](#).

Para utilizar este driver, você precisará adicionar as seguintes linhas ao seu arquivo de configuração do kernel:

```
device atapicam
device scbus
device cd
device pass
```

Você também precisará adicionar as seguintes linhas:

```
device ata
device atapicd
```

Ambas deverão estar presentes.

Então recompile, instale o novo kernel e reinicie seu sistema. Durante o processo de inicialização, seu gravador de CDs deverá retornar algo como:

```
acd0: CD-RW <MATSHITA CD-RW/DVD-ROM UJDA740> at ata1-master PIO4
cd0 at ata1 bus 0 target 0 lun 0
cd0: <MATSHITA CDRW/DVD UJDA740 1.00> Removable CD-ROM SCSI-0 device
cd0: 16.000MB/s transfers
cd0: Attempt to query device size failed: NOT READY, Medium not
present - tray closed
```

O drive agora pode ser acessado através do nome de dispositivo `/dev/cd0`, por exemplo, para montar o CDROM em `/mnt` faça o seguinte:

```
mount -t cd9660 /dev/cd0c /mnt
```

Como usuário `root` você pode executar o seguinte comando para saber o endereço SCSI do gravador:

```
camcontrol devlist
<MATSHITA CDRW/DVD UJDA740 1.00> at scbus1 target 0 lun 0
(pass0,cd0)
```

Temos que o `1,0,0` será o endereço SCSI para utilizar no [cdrecord\(1\)](#) e para outras aplicações SCSI.

Para maior informações sobre ATAPI/CAM e sistema SCSI, visite as páginas de manual do [atapicam\(4\)](#) e do [cam\(4\)](#).

---

## 12.6. Criação e utilização de disquetes

*Trabalho original de Julio Merino. Reescrito por Martin Karlsson.*

Armazenar dados em disquetes pode ser útil algumas vezes, por exemplo, quando você não tem nenhum outro tipo de mídia de armazenamento removível ou quando você precisa transferir pequena quantidade de dados para um outro computador.

Esta seção explicará como utilizar disquetes no FreeBSD. Primeiramente, cobrirá a formatação e o uso de disquetes DOS de 3.5 polegadas, mas os conceitos são similares em relação ao uso de outros formatos.

### 12.6.1. Formatando disquetes

#### 12.6.1.1. O Dispositivo

Disquetes são acessados através de entradas em `/dev`, assim como outros dispositivos. Para acessar um disquete gravado em raw no 4.X ou versões anteriores use `/dev/fdN`, onde o `N` indica o número da controladora, usualmente 0, ou `/dev/fdNX`, onde `X` indica a letra do dispositivo.

Na versão 5.0 ou acima, use simplesmente o `/dev/fdN`.

##### 12.6.1.1.1. O tamanho do disco no 4.X e nas versões anteriores

Estes também possuem os dispositivos `/dev/fdN.size`, onde `size` é o tamanho do disquete em kilobytes. Estas entradas são utilizadas em formatação de baixo-nível, para determinar o tamanho do disco. 1440kB é o tamanho que será utilizado no exemplo à seguir.

Algumas vezes as entradas no `/dev` terão de ser (re)criadas. Para fazer isso:



```
cd /dev && ./MAKEDEV "fd*"
```

#### **12.6.1.1.2. O tamanho do disco na 5.0 e versões superiores**

Na 5.0, o [devfs\(5\)](#) administrará automaticamente os dispositivos em /dev, o que torna o uso do MAKEDEV desnecessário.

O tamanho desejado do disco é passado para o [fdformat\(1\)](#) através da opção `-f`. Os tamanhos suportados são listados no [fdcontrol\(8\)](#), mas esteja avisado que o 1440kB é o que funciona melhor.

### **12.6.1.2. Formatando**

Um disquete precisa ser formatado em baixo-nível antes de estar utilizável. Isto geralmente é realizado pelos fabricantes, mas este tipo de formatação é uma ótima maneira de verificar a integridade da mídia. E ainda é possível forçar tamanhos de discos maiores (ou menores), mas a maioria dos discos são desenvolvidos para o tamanho 1440kB.

Para formatar um disquete em baixo-nível você precisa utilizar o [fdformat\(1\)](#). Este utilitário necessita do nome do dispositivo como argumento.

Anote qualquer mensagem de erro, isso pode ajudar a determinar se o disco está bom ou ruim.

#### **12.6.1.2.1. Formatando no 4.X ou versões anteriores**

Utilize o dispositivo `/dev/fdN.size` para formatar o disquete. Insira um disquete novo de 3.5 polegadas no seu drive e execute:

```
/usr/sbin/fdformat /dev/fd0.1440
```

#### **12.6.1.2.2. Formatando no 5.0 e versões superiores**

Utilize o dispositivo `/dev/fdN` para formatar o disquete. `devices to format the floppy.` Insira um disquete novo de 3.5 polegadas no seu drive e execute:

```
/usr/sbin/fdformat -f 1440 /dev/fd0
```

### **12.6.2. O rótulo do disco**

Depois da formatação de baixo-nível você precisará colocar um rótulo no disquete. Este rótulo será destruído mais tarde, mas é necessário mais tarde para o sistema determinar o tamanho e a geometria do disco.

O novo rótulo do disco indicará o disco todo, e conterá todas as informações próprias sobre a geometria do disquete. Os valores de geometria para rótulos de disquetes estão listados em `/etc/disktab`.

Você pode executar agora o [disklabel\(8\)](#):

```
/sbin/disklabel -B -r -w /dev/fd0 fd1440
```

**Nota:** Desde o FreeBSD 5.1-RELEASE, o utilitário [bsdlable\(8\)](#) substitui o velho programa [disklabel\(8\)](#). Com o [bsdlable\(8\)](#), um número de opções e parâmetros obsoletos foram retirados; no exemplo a opção *-r* deveria ser retirada. Para mais informações, por favor, visite a página de manual do [bsdlable\(8\)](#).

### 12.6.3. O sistema de arquivos

Agora o disquete está pronto para ser formatado em alto-nível. Isto estabelecerá um novo sistema de arquivos no disquete, que permitirá ao FreeBSD ler e gravar nele. Depois de criar um novo sistema de arquivos, o rótulo do disquete é destruído, portanto, se você desejar reformatar o disco será necessário recriar o rótulo do disquete.

O sistema de arquivos do disquete pode ser UFS ou FAT. FAT é geralmente a melhor escolha para disquetes.

Para estabelecer um novo sistema de arquivos no disquete, faça:

```
/sbin/newfs_msdos /dev/fd0
```

O disquete agora está pronto para ser utilizado.

### 12.6.4. Utilizando o disquete

Para utilizar o disquete, monte-o como [mount\\_msdos\(8\)](#) (no 4.X e versões anteriores) ou como [mount\\_msdfs\(8\)](#) (no 5.0 e versões superiores). Pode-se também utilizar o [emulators/mttools](#) da coleção de ports.

---

## 12.7. Criando e utilizando fitas DAT

As principais mídias de fitas são 4mm, 8mm, QIC, mini-cartucho e DLT.

### 12.7.1. 4mm (DDS: Digital Data Storage)

As fitas 4mm estão substituindo as QIC na escolha de mídias para backups em estações de trabalho. Esta tendência cresce de maneira acelerada devido a compra da Archive pela Conner, fabricante líder de controladoras QIC, e pela parada na produção de controladoras QIC. As controladoras 4mm são menores e mais silenciosas porém não possuem a reputação de confiabilidade que é dada as controladoras 8mm. Os cartuchos são mais baratos e menores (3 x 2 x 0.5 polegadas, 76 x 51 x 12 mm) que os cartuchos 8mm. As fitas 4mm, como as 8mm, tem comparativamente o mesmo tempo de vida no cabeçote de leitura e gravação, ambas usam *helical scan*.

As taxas de transferências nestas controladoras iniciam em ~150 kB/s, chegando a ~500 kB/s. Sua capacidade inicia em 1.3 GB chegando até 2.0 GB. Compressão de

Hardware está disponível para a maioria destas controladoras, o que chega a aproximadamente dobrar a capacidade. Uma controladora com múltiplas unidades pode ter até 6 controladoras em um único gabinete com mudança de fita automática. Tendo capacidade no alcance de 240 GB.

O padrão DDS-3 agora suporta fitas com capacidade maiores que 12 GB (ou 24 GB com compressão).

As controladoras 4mm, assim como as 8mm, utilizam *helical-scan*. Todos os benefícios e desvantagens disso estão aplicados em ambas as controladoras, 4mm e 8mm.

As fitas devem ser inutilizadas após 2,000 passagens ou 100 backups completos.

### 12.7.2. 8mm (Exabyte)

As fitas 8mm são as mais comuns em controladoras de fita SCSI; são a melhor escolha para as trocas de fitas. Quase todos possuem uma controladora de fitas Exabyte 2 GB 8mm. As controladoras 8mm são confiáveis, convenientes e silenciosas. Os cartuchos não são caros e são pequenos (4.8 x 3.3 x 0.6 polegadas; 122 x 84 x 15 mm). Um ponto negativo para as controladoras 8mm é a vida curta do cabeçote de leitura e gravação e das fitas devido a alta frequência de movimentação da fita sobre o cabeçote.

As taxas de transferências vão de ~250 kB/s até ~500 kB/s. Tendo capacidade iniciando em 300 MB indo até 7 GB. Compressão de Hardware esta disponível na maioria das controladoras, o que aproximadamente dobra a capacidade delas. Estas controladoras estão disponíveis em unidades ou em multi-controladoras com 6 controladoras e 120 fitas num único gabinete. A mudança de fita é automática por unidade. Chegando a capacidades no alcance de 840+ GB.

O modelo Exabyte "Mammoth" suporta 12 GB em uma fita (24 GB with compression) e custa aproximadamente duas vezes mais que as fitas convencionais.

Os dados são gravados na fita utilizando *helical-scan*, as cabeças de leitura e gravação são posicionadas em um ângulo para a mídia (aproximadamente 6 graus). O envólucro da fita está a 270 graus do carretel que segura a cabeça de leitura e gravação. O carretel gira enquanto a fita desliza sobre ele. O resultado é uma alta densidade de dados e trilhas empacotadas bem próximas na fita de uma borda até a outra.

### 12.7.3. QIC

As fitas e controladoras QIC-150 são, talvez, as mais comuns por ai. As controladoras QIC são as mais baratas controladoras de backup consideradas "sérias". O ponto negativo é o preço das mídias. Fitas QIC são caras se comparadas as 8mm ou 4mm, mais ou menos 5 vezes o preço por GB de dados armazenados. Mas, se suas necessidades podem ser satisfeitas com metade das fitas, QIC pode ser a escolha certa. QIC é o tipo *mais* comum de controladora de fitas. Cada localidade possui uma controladora QIC de alguma densidade ou de outra. Nisso está a vantagem, QIC tem um número grande de densidades de fitas fisicamente similares (algumas vezes idênticas). As controladoras QIC não são silenciosas. Estas controladoras buscam audivelmente antes de iniciar a gravação dos dados e são audíveis enquanto estão lendo, escrevendo e

buscando os dados. As medidas das fitas QIC(6 x 4 x 0.7 polegadas; 15.2 x 10.2 x 1.7 mm). [Mini-cartridges](#), que também fazem vasto uso de fitas de 1/4" são discutidas separadamente. Não estão disponíveis multi-controladoras ou troca de fitas automática.

As taxas de transferência de dados vão de ~150 kB/s até ~500 kB/s. A capacidade de armazenamento vai de 40 MB até 15 GB. Compressão de Hardware está disponível na maioria das novas controladoras QIC. As controladoras QIC são menos frequentemente instaladas; elas estão sendo substituídas pelas controladoras DAT.

Os dados são gravados nas fitas em trilhas. As trilhas giram em torno da linha central da fita de um final ao outro. O número de trilhas e conseqüentemente a largura delas variam com a capacidade de armazenamento da fita. A maioria senão todas as controladoras provêm compatibilidade com as antigas, pelo menos para leitura ( e frequentemente para gravação também). QIC tem uma boa reputação considerando a segurança dos dados (a mecânica é simples e mais robusta que as controladoras com *helical scan* ).

As fitas devem ser inutilizadas após 5,000 backups.

#### **12.7.4. XXX\* Mini-Cartridge ou Mini-Cartucho**

#### **12.7.5. DLT**

DLT tem a mais rápida taxa de transferência de dados de todas as controladoras listadas aqui. A fita de 1/2" (12.5mm) é colocada em um cartucho de carretel único (4 x 4 x 1 polegadas; 100 x 100 x 25 mm). O cartucho tem uma porta balançante sobre uma lado todo. O mecanismo da controladora abre esta porta para extrair o líder. O líder da fita tem um buraco oval que a controladora usa para "engancha" a fita. O carretel auxiliar é localizado dentro da controladora de fita. Todos os outros cartuchos de fita listados aqui (9 fitas de trilhas são as únicas exceções) tem ambos, o carretel de fornecimento e o carretel auxiliar localizado dentro do próprio cartucho da fita.

As taxas de transferência são aproximadamente 1.5 MB/s, três vezes a das controladoras 4mm, 8mm, ou QIC. A capacidade vai de 10 GB até 20 GB para uma única controladora. Controladoras multi-fitas e com troca automática estão disponíveis contendo de 5 a 900 fitas sobre 1 a 20 controladoras, disponibilizando de 50 GB a 9 TB de armazenamento.

Com compressão, o formato DLT Type IV suporta a capacidade de 70 GB.

Os dados são gravados na fita em trilhas paralelas na direção de viagem (assim como em fitas QIC). Duas trilhas são escritas de uma vez. As vida útil das cabeças de leitura e gravação são relativamente longas; uma vez que a fita para de mover, não há movimentação entre a fita e a cabeça de leitura e gravação.

#### **12.7.6. AIT**

AIT é um novo formato da Sony, e pode armazenar 50 GB (com compressão) por fita. Estas fitas possuem chips de memória que armazenam índices do conteúdo da fita. Este índice pode ser rapidamente lido pela controladora para determinar a posição dos

arquivos na fita, ao invés dos vários minutos exigidos por outros tipos de controladoras de fitas. Programas como **SAMS:Alexandria** podem operar quarenta ou mais controladoras AIT, comunicando diretamente com os chips de memória das fitas e apresentando o conteúdo na tela, permitindo determinar em que fita os arquivos estão, localizando a fita correta, carregando ela, e restaurando os dados da fita.

Multi-controladoras como a citada acima custam por volta de \$20,000, um preço um pouco fora do mercado hobbista.

### 12.7.7. Utilizando uma fita nova pela primeira vez

A primeira vez que você tenta ler ou escrever em uma fita nova, completamente vazia, a operação irá falhar. A mensagem que você receberá será similar a:

```
sa0(ncr1:4:0): NOT READY asc:4,1
sa0(ncr1:4:0): Logical unit is in process of becoming ready
```

A fita não contém um Bloco Identificador (bloco número 0). Todas as controladoras de fita QIC desde que adotaram o padrão QIC-525 escrevem um Bloco Identificador na fita. Há duas soluções:

- `mt fsf 1` faz a escrita do Bloco Identificador na fita.
- Utilize o botão na frente do painel para ejetar a fita.

Reinsira a fita e faça o `dump` dos dados para a fita.

`dump` reportará ```DUMP: End of tape detected``` e a console mostrará:  
```HARDWARE FAILURE info:280 asc:80,96```.

Rebobine a fita usando: `mt rewind`.

As operações subsequentes resultarão em sucesso.

12.8. Backups em Disquetes

12.8.1. Posso utilizar disquetes para realizar backup dos meus dados?

Disquetes não são realmente boas mídias para backup devido a:

- A mídia não é confiável, especialmente se tem longo tempo de uso.
- Realizar o backup e restaurar os dados é um processo muito lento.
- Eles tem uma capacidade de armazenamento muito limitada (os dias de realizar um backup de um disco rígido inteiro podem somar uma dúzia e uma quantidade enorme de disquetes a medida que este tempo passar).

Entretanto, se você não tiver outra maneira de realizar backup dos seus dados, disquetes são melhores do que nada, ou nenhum backup.

Se você tiver que fazer uso de disquetes tenha certeza da boa qualidade deles. Disquetes que foram utilizados por anos no escritório são uma péssima escolha. Idealmente, use disquetes novos de um fabricante com boa reputação.

12.8.2. Mas, como eu realizo backup dos meus dados em disquetes?

A melhor maneira de realizar backups em disquetes é utilizando o [tar\(1\)](#) com a opção *-M* (multi volume), que permitirá que o backup ocupe múltiplos disquetes.

Para realizar backup de todos os arquivos no diretório corrente e subdiretórios deste, faça isso (como `root`):

```
# tar Mcvf /dev/fd0 *
```

Quando o primeiro disquete estiver cheio o [tar\(1\)](#) pedirá para que você insira o próximo volume (devido ao [tar\(1\)](#) fazer com que as mídias sejam independentes referindo-se a um volume; neste contexto, volumes são os disquetes).

```
Prepare volume #2 for /dev/fd0 and hit return:
```

Isto se repetirá (com o número do volume sendo incrementado) até que todos os arquivos especificados estejam arquivados.

12.8.3. Eu posso usar compressão em meus backups?

Desafortunadamente, o [tar\(1\)](#) não permitirá o uso da opção *-z* com a especificação de arquivos sendo gerados com volumes múltiplos. Você poderia, claro, usar o [gzip\(1\)](#) em todos os arquivos, e então usar o [tar\(1\)](#) para arquivá-los nos disquetes, e então ao restaurar utilizar o [gunzip\(1\)](#) para obter os arquivos novamente!

12.8.4. Como eu restauro os dados dos meus backups?

Para restaurar o arquivo todo use:

```
# tar Mxvf /dev/fd0
```

Há duas maneiras que podem ser utilizadas para a restauração de um arquivos em específico. Primeiro, você precisar iniciar com o primeiro disquete e executar:

```
# tar Mxvf /dev/fd0 filename
```

O utilitário [tar\(1\)](#) irá pedir a inserção dos disquetes subsequentes até encontrar o arquivo requerido.

Alternativamente, se você sabe em qual disquete o arquivo está, você pode simplesmente inseri-lo na unidade de disquetes e utilizar o comando citado anteriormente. Note que se o arquivo for o primeiro e este ocupar outro disquete o [tar\(1\)](#) irá alertá-lo que este não pode ser restaurado, mesmo se você não tenha pedido para que ele fizesse isso!

12.9. Considerações básicas sobre backup

Os três mais utilizados programas de backup são [dump\(8\)](#), [tar\(1\)](#), e [cpio\(1\)](#).

12.9.1. Dump e Restore

Os programas tradicionais de backup do UNIX® são o `dump` e o `restore`. Eles operam a controladora como se esta fosse uma coleção de blocos de discos, abaixo da abstração de arquivos, links e diretórios que são criadas pelos sistemas de arquivos. O `dump` realiza o backup de todo um sistema de arquivos em um dispositivo. Ele não tem habilidade de realizar backup somente de uma parte de um sistema de arquivos ou uma árvore de diretórios que envolva mais de um sistema de arquivos. O `dump` não grava os arquivos em diretórios numa fita, ele grava tudo de maneira direta (raw) em blocos de dados que compõe os arquivos e diretórios.

Nota: Se você utilizar o `dump` no seu diretório root, você não realizaria o backup do `/home`, `/usr` e muitos outros subdiretórios desde que estes sejam tipicamente pontos de montagem para outros sistemas de arquivos, ou ainda links entre estes sistemas de arquivos.

O `dump` tem uma artimanha que vem desde primeiros dias na Versão 6 do AT&T UNIX (aproximadamente 1975). Os parâmetros padrões são adequados para fitas de 9 trilhas (6250 bpi), não para as mídias de alta-densidade disponíveis hoje em dia (up to 62,182 ftpi). Estes padrões devem ser sobreescritos na linha de comando para que seja utilizada a capacidade das atuais controladoras de fita.

Também é possível realizar backup dos dados através da rede para um dispositivo de backup conectado em um outro computador com o `rdump` e o `rrestore`. Ambos os programas tem sua confiança baseadas no `rcmd` e no `ruserok` para o acesso à controladora de fita remota. Portanto, o usuário que irá realizar o backup deverá estar listado no arquivo `.rhosts` no computador remoto. Os argumentos para o `rdump` e para o `rrestore` devem ser adequados para o computador remoto. Quando estiver fazendo o `rdump` de um computador com FreeBSD para uma controladora de fita Exabyte conectada numa estação Sun chamada `komodo`, use:

```
# /sbin/rdump 0dsbfu 54000 13000 126 komodo:/dev/nsa8 /dev/da0a 2>&1
```

Cuidado: há implicações na segurança do sistema permitir a autenticação pelo `.rhosts`. Avalie a sua situação cuidadosamente.

Também é possível utilizar o `dump` e o `restore` numa confecção mais segura através do `ssh`.

Exemplo 12-1. Utilizando o `dump` sobre o `ssh`

```
# /sbin/dump -0uan -f - /usr | gzip -2 | ssh1 -c blowfish \
```

```
targetuser@targetmachine.example.com dd of=/mybigfiles/dump-usr-10.gz
```

12.9.2. tar

O [tar\(1\)](#) também tem sua data de volta a Versão 6 of AT&T UNIX (aproximadamente 1975). O `tar` opera em cooperação com o sistema de arquivos; o `tar` grava os arquivos e diretórios na fita. O `tar` não suporta a gama enorme de opções disponíveis no [cpio\(1\)](#), mas o `tar` não requer o pouco usual esquema de comandos com redirecionamento que o `cpio` usa.

A maioria das versões do `tar` não suportam backups através da rede. A versão GNU do `tar`, que é a versão utilizada pelo FreeBSD, suporta dispositivos remotos utilizando a mesma sintaxe do `rdump`. Para utilizar o `tar` de um computador com FreeBSD para uma controladora de fita Exabyte conectada numa estação Sun chamada `komodo`, use:

```
# /usr/bin/tar cf komodo:/dev/nsa8 . 2>&1
```

Para as versões sem suporte a dispositivos remotos, é possível usar um redirecionamento com o `rsh` para enviar os dados para uma controladora de fita remota.

```
# tar cf - . | rsh hostname dd of=tape-device obs=20b
```

Se você está preocupado com a segurança no que envolve realizar backups através da rede você deveria utilizar o `ssh` ao invés do `rsh`.

12.9.3. cpio

O [cpio\(1\)](#) é o programa original do UNIX para a troca de arquivos entre máquinas por meio de mídias magnéticas. O `cpio` tem opções (entre muitas outras) de realizar *byte-swapping*, gravando um bom número de sistemas de arquivos de diferentes formatos, e permitindo o redirecionamento dos dados para outros programas. Esta última funcionalidade faz do `cpio` uma excelente escolha para criar mídias de instalações. O `cpio` não sabe como andar nos diretórios da árvore de diretórios e uma lista dos arquivos deve ser passada através da `stdin`.

O `cpio` não suporta a realização de backups através da rede. Você pode utilizar um redirecionamento e o `rsh` para enviar os dados para uma controladora de fita remota.

```
# for f in directory_list; do
find $f >> backup.list
done
# cpio -v -o --format=newc < backup.list | ssh user@host "cat >
backup_device"
```

Onde o `directory_list` é a lista de diretórios que você quer adicionar ao seu backup, `user@host` é a combinação usuário/computador que irá realizar o backup, e `backup_device` é onde os backups devem ser escritos (e.g., `/dev/nsa0`).

12.9.4. `pax`

O `pax(1)` é a resposta do IEEE/POSIX® para o `tar` e o `cpio`. Sobre os anos de várias versões do `tar` e do `cpio` estão algumas incompatibilidades. Então para sair da briga pela padronização completa deles, o POSIX criou um novo utilitário de arquivamento. O `pax` tenta ler e gravar nos mais variados formatos tanto do `cpio` quanto do `tar`, adicionado novos formatos por si, inclusive. Sua maneira de comandar é mais semelhante ao `cpio` que ao `tar`.

12.9.5. **Amanda**

O **Amanda** (Advanced Maryland Network Disk Archiver) é um sistema de backup cliente/servidor, melhor que um único programa. Um servidor **Amanda** irá realizar numa única controladora de fita o backup de qualquer número de computadores que tenham o cliente do **Amanda** e uma conexão de com o servidor **Amanda**. Um problema comum em locais com um grande número de discos é que a quantidade de tempo requerida para o backup dos dados diretamente na fita excede a quantidade de tempo para a tarefa. O **Amanda** resolve este problema. O **Amanda** pode utilizar um disco auxiliar para realizar o backup de diversos sistemas de arquivos ao mesmo tempo. O **Amanda** cria ``conjuntos de arquivos ``: um grupo de fitas utilizadas sobre o tempo para criar os backups completos de todos os sistemas de arquivos listados no arquivo de configuração do **Amanda**. O ``conjunto de arquivos `` também pode conter backups incrementais (ou diferenciais) noturnos de todos os sistemas de arquivos. Para restaurar um sistema de arquivos é necessário o backup completo mais recente e os incrementais.

O arquivo de configurações provê um controle total da realização dos backups e do tráfego de rede que o **Amanda** gera. O **Amanda** utilizará qualquer um dos programas de backup citados para gravar os dados nas fitas. O **Amanda** está disponível também como um port ou como pacote, porém ele não é instalado por padrão.

12.9.6. Não fazer nada

``Não fazer nada`` não é um programa de computador mas sim a mais utilizada estratégia de backup. Não há custos iniciais. Não há um agendamento de backup a ser cumprido. Apenas diga não. Se alguma coisa acontecer aos seus dados dê um sorriso malicioso e dê cria neles!

Se seu tempo e seus dados são um pouquinho mais que nada, então ``Não fazer nada`` é o mais adequado programa de backup para o seus computadores. Mas cuidado, o UNIX é uma ferramenta útil, você deve encontrar num intervalo de seis meses uma coleção de arquivos que tenham valor para você.

``Não fazer nada`` é o método de backup correto para `/usr/obj` e outras árvores de diretórios que podem ser criadas exatamente como eram no seu computador. Um exemplo são os arquivos HTML ou PostScript® das versões deste Handbook. Estes documentos foram criados de arquivos no formato SGML. Criar backups deles não é necessário. Os arquivos SGML tem backups realizados regularmente.

12.9.7. Qual o melhor programa de backup?

A *vez* é do [dump\(8\)](#), pelos resultados obtidos do torturantes testes que Elizabeth D. Zwicky fez a todos os programas discutidos aqui. A escolha limpa para preservar todos os seus dados e suas peculiaridades do sistema de arquivos UNIX é o `dump`. Elizabeth criou um sistema de arquivos contendo uma grande variedade de condições não usuais (e algumas nem tão pouco usuais assim) e realizou testes em cada um dos programas realizando backup e restaurando o sistema de arquivos criado. As peculiaridades incluídas: arquivos com buracos, arquivos buracos e blocos nulos, arquivos com caracteres engraçados e estranhos em seus nomes, arquivos que não podem ser lidos ou gravados, arquivos de dispositivos, arquivos que mudam de tamanho durante o backup, arquivos que são criados e apagados durante o backup entre outros. Ela apresentou os resultados na LISA V em outubro de 1991. Visite [torture-testing Backup and Archive Programs](#).

12.9.8. Procedimento de restauração de emergência

12.9.8.1. Antes do desastre

Existem pelo menos quatro passos que são necessários para preparar caso ocorra um desastre..

Primeiro, imprima os rótulos de discos de cada um dos discos que você possui (e.g. `disklabel da0 | lpr`), sua tabela de sistemas de arquivos (`/etc/fstab`) e todas as mensagens de inicialização, isso tudo em duas cópias.

Segundo, determine quais serão os disquetes de boot e de manutenção (`boot.flp` e `fixit.flp`), tendo todos os arquivos para os seus dispositivos. A maneira mais fácil de verificar é reiniciar sua máquina com o disquete de boot na unidade e checar as mensagens emitidas. Se todos os arquivos de dispositivos são listados e funcionais, passe ao passo três.

Senão, você terá que criar dois disquetes de boot personalizados contendo um kernel que possa montar todos os seus discos e acessar a fita de backup na controladora. Este disquetes devem conter: `fdisk`, `disklabel`, `newfs`, `mount`, e qualquer programa de backup, de preferência o que você usa. Este programas deverão ser gerados de maneira estática. Se você utilizar o `dump`, o disquete deverá conter o `restore`.

Terceiro, crie backups nas fitas regularmente. Qualquer mudança que for feita após o último backup será perdida. Proteja as fitas de backup ativando a trava contra gravação.

Quanto, teste os disquetes (mesmo o `boot.flp` e o `fixit.flp` ou od dois disquetes de boot personalizados que você fez no segundo passo.) e as fitas de backup. Tome nota do processo. Guarde estas notas com o disquete bootável, as etiquetas e as fitas de backup. Você poderá estar tão perturbado no momento da restauração que as notas irão prevenir que você destrua as fitas de backup (Como? no lugar de `tar xvf /dev/sa0`, você acidentalmente digita `tar cvf /dev/sa0` e sobrescreve sua fita de backup).

Para adicionar segurança ao processo, faça disquetes inicializáveis e duas fitas de backup por vez. Armazene cada uma delas em locais remotos diferentes. Um local remoto não é no subsolo do mesmo prédio. Um certo número de empresas no World Trade Center aprendeu esta lição do modo mais difícil. Um local remoto deve ser fisicamente separado de computadores e discos por uma distância significativa.

Exemplo 12-2. Um Script para a criação de disquetes de inicialização

```
#!/bin/sh
#
# Criar um disquete para restauração
#
# Formatar o disquete
#
PATH=/bin:/sbin:/usr/sbin:/usr/bin

fdformat -q fd0
if [ $? -ne 0 ]
then
    echo "Disquete problemático, insira um novo"
    exit 1
fi

# Colocar os blocos de boot no disquete
#
disklabel -w -B /dev/fd0c fd1440

#
# Criar a partição no disquete
#
newfs -t 2 -u 18 -l 1 -c 40 -i 5120 -m 5 -o space /dev/fd0a

#
# Montar o novo disquete
#
mount /dev/fd0a /mnt

#
# Criar diretórios necessários
#
mkdir /mnt/dev
mkdir /mnt/bin
mkdir /mnt/sbin
mkdir /mnt/etc
mkdir /mnt/root
mkdir /mnt/mnt          # Para a partição root
mkdir /mnt/tmp
mkdir /mnt/var

#
# Popular os diretórios
#
if [ ! -x /sys/compile/MINI/kernel ]
then
    cat << EOM
O kernel MINI não existe, por favor, crie um.
Veja aqui um exemplo para o arquivo de configuração:
#
# MINI -- Um kernel para usar o FreeBSD num disquete.
#
```

```

machine      "i386"
cpu          "I486_CPU"
ident       MINI
maxusers    5

options      INET                # needed for _tcp _icmpstat
_ipstat     #                    #          _udpstat _tcpstat

_u db
options     FFS                  #Berkeley Fast File System
options     FAT_CURSOR          #block cursor in syscons or
pccons
options     SCSI_DELAY=15      #Be pessimistic about Joe SCSI
device
options     NCONS=2            #1 virtual consoles
options     USERCONFIG        #Allow user configuration with
-c XXX

config      kernel  root on da0 swap on da0 and da1 dumps on da0

device      isa0
device      pci0

device      fdc0      at isa? port "IO_FD1" bio irq 6 drq 2 vector
fdintr
device      fd0 at fdc0 drive 0

device      ncr0

device      scbus0

device      sc0 at isa? port "IO_KBD" tty irq 1 vector scintr
device      npx0      at isa? port "IO_NPX" irq 13 vector npxintr

device      da0
device      da1
device      da2

device      sa0

pseudo-device  loop                # required by INET
pseudo-device  gzip                # Exec gzipped a.out's
EOM
    exit 1
fi

cp -f /sys/compile/MINI/kernel /mnt

gzip -c -best /sbin/init > /mnt/sbin/init
gzip -c -best /sbin/fsck > /mnt/sbin/fsck
gzip -c -best /sbin/mount > /mnt/sbin/mount
gzip -c -best /sbin/halt > /mnt/sbin/halt
gzip -c -best /sbin/restore > /mnt/sbin/restore

gzip -c -best /bin/sh > /mnt/bin/sh
gzip -c -best /bin/sync > /mnt/bin/sync

cp /root/.profile /mnt/root

cp -f /dev/MAKEDEV /mnt/dev
chmod 755 /mnt/dev/MAKEDEV

```

```

chmod 500 /mnt/sbin/init
chmod 555 /mnt/sbin/fsck /mnt/sbin/mount /mnt/sbin/halt
chmod 555 /mnt/bin/sh /mnt/bin/sync
chmod 6555 /mnt/sbin/restore

#
# Criar os n&ocirc;s de dispositivos
#
cd /mnt/dev
./MAKEDEV std
./MAKEDEV da0
./MAKEDEV da1
./MAKEDEV da2
./MAKEDEV sa0
./MAKEDEV pty0
cd /

#
# Criar a tabela de sistema de arquivos m&iacute;nimo
#
cat > /mnt/etc/fstab <<EOM
/dev/fd0a    /      ufs      rw  1  1
EOM

#
# Criar um arquivo passwd m&iacute;nimo
#
cat > /mnt/etc/passwd <<EOM
root:*:0:0:Charlie &:/root:/bin/sh
EOM

cat > /mnt/etc/master.passwd <<EOM
root::0:0::0:0:Charlie &:/root:/bin/sh
EOM

chmod 600 /mnt/etc/master.passwd
chmod 644 /mnt/etc/passwd
/usr/sbin/pwd_mkdb -d/mnt/etc /mnt/etc/master.passwd

#
# Desmontar o disquete e informar ao usu&aacute;rio
#
/sbin/umount /mnt
echo "O disquete foi desmontado e esta pronto para o uso."

```

12.9.8.2. Ap&osil; o disastre

A quest&ao; chave é: seu equipamento sobreviveu? Vocé tem realizado backups regulares para n&ao; se preocupar com problemas de software.

Se o equipamento foi danificado, as partes danificadas devem ser trocadas antes de tentar usar o computador.

Se seu equipamento est&aa; funcional, verifique seus disquetes. Caso vocé esteja utilizando um disquete inicializ&avel; personalizado, entre no sistema em modo mono-usu&ario; (digite -s no prompt boot :). Passe adiante no pr&oximo; par&agrafo;.

Se você estiver utilizando os disquetes `boot.flp` e `fixit.flp`, continue lendo. Insira o disquete `boot.flp` na primeira unidade de disquetes e realize o boot no computador. O menu de instalação original aparecerá na tela. Selecione a opção *Fixit--Repair mode with CDRom or floppy.* Insira o disquete `fixit.flp` quando for solicitado. O `restore` e outros programas que você precisa estão localizados em `/mnt2/stand`.

Restaurar cada sistema de arquivos separadamente.

Tente o `mount` (e.g. `mount /dev/da0a /mnt`) na partição raiz do seu primeiro disco. Se o seu rótulo de disco foi danificado, use o `disklabel` para reparticionar e rotular o disco da mesma maneira que você tem salva e impressa. Use o `newfs` para recriar os sistemas de arquivos. Remonte a partição raiz do disquete com direitos de gravação (`mount -u -o rw /mnt`). Utilize seu programa e fitas de backup para restaurar os dados dos sistemas de arquivos (e.g. `restore vrf /dev/sa0`). Desmonte o sistema de arquivos (e.g. `umount /mnt`). Repita o processo para cada um dos sistemas de arquivos danificados.

Uma vez que seu sistema está em execução, realize backup dos seus dados em novas fitas. Qualquer parada ou perda de dados pode acontecer novamente sem que a causa seja conhecida. Gastar mais uma hora agora pode salvá-lo do estresse mais tarde.

12.10. Rede, Memória, e Sistemas de Arquivos em Imagens

Reorganizado e melhorado por Marc Fonvieille.

Além dos discos que você insere fisicamente ao seu computador: disquetes, CDs, discos rígidos, ou qual seja; outras formas de discos são entendidas pelo FreeBSD - os *discos virtuais*.

Insto inclui sistemas de arquivos em rede como o [Network File System](#) e o Coda, sistemas de arquivos baseados em memória e sistema de arquivos em imagens.

Dependendo da versão de FreeBSD que você usa, você terá que utilizar ferramentas diferentes para criar e utilizar sistemas de arquivos em imagens e baseados em memória.

Nota: O usuários do FreeBSD 4.X utilizarão [MAKEDEV\(8\)](#) para criar os dispositivos necessários. O FreeBSD 5.0 e versões superiores utilizarão o [devfs\(5\)](#) para alocar os nós de dispositivos de maneira transparente para o usuário.

12.10.1. Sistemas de arquivos em arquivo de imagem no FreeBSD 4.X

O utilitário [vnconfig\(8\)](#) configura e disponibiliza o `vnode` que é um dispositivo para pseudo-discos. Um `vnode` é uma representação de um arquivo e é o foco para a atividade com os arquivos. Isto significa que o [vnconfig\(8\)](#) utiliza arquivos arquivos

para criar e operar sistemas de arquivos. Uma das utilizações possíveis seria a montagem de arquivos de imagens de CDs ou de disquetes.

Para utilizar o [vnconfig\(8\)](#) você precisa do suporte [vn\(4\)](#), para adicioná-lo ao seu arquivo de configuração de kernel coloque:

```
pseudo-device vn
```

Para montar uma imagem de um sistema de arquivos:

Exemplo 12-3. Utilizando o vnconfig para montar uma imagem de um sistema de arquivos no FreeBSD 4.X

```
# vnconfig vn0 diskimage
# mount /dev/vn0c /mnt
```

Para criar uma nova imagem de sistema de arquivos com o [vnconfig\(8\)](#):

Exemplo 12-4. Criando uma nova imagem de disco com o vnconfig

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# vnconfig -s labels -c vn0 newimage
# disklabel -r -w vn0 auto
# newfs vn0c
Warning: 2048 sector(s) in last cylinder unallocated
/dev/vn0c:      10240 sectors in 3 cylinders of 1 tracks, 4096 sectors
              5.0MB in 1 cyl groups (16 c/g, 32.00MB/g, 1280 i/g)
super-block backups (for fsck -b #) at:
   32
# mount /dev/vn0c /mnt
# df /mnt
Filesystem 1K-blocks      Used      Avail Capacity  Mounted on
/dev/vn0c      4927          1      4532      0%      /mnt
```

12.10.2. Sistemas de arquivos em imagem no FreeBSD 5.X

O utilitário [mdconfig\(8\)](#) é utilizado para configurar e ativar discos em memória, [md\(4\)](#) no FreeBSD 5.X. Para usar o [mdconfig\(8\)](#) você terá que carregar o módulo [md\(4\)](#) ou adicionar o suporte no arquivo de configuração do kernel:

```
device md
```

O comando [mdconfig\(8\)](#) suporta três tipos de discos em memória: discos de memória alocados com o [malloc\(9\)](#), discos de memória utilizando um arquivo e espaços de troca. Um possível uso é a montagem de arquivos de imagens de disquetes e CDs.

Para montar um sistema de arquivos existente em um arquivo de imagem:

Exemplo 12-5. Usando o mdconfig para montar um arquivo de imagem de um sistema de arquivos no FreeBSD 5.X

```
# mdconfig -a -t vnode -f diskimage -u 0
```

```
# mount /dev/md0c /mnt
```

Para criar uma nova imagem de sistema de arquivos com o [mdconfig\(8\)](#):

Exemplo 12-6. Criando um novo disco em arquivo de imagem com o `mdconfig`

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# mdconfig -a -t vnode -f newimage -u 0
# disklabel -r -w md0 auto
# newfs md0c
/dev/md0c: 5.0MB (10240 sectors) block size 16384, fragment size 2048
    using 4 cylinder groups of 1.27MB, 81 blks, 256 inodes.
super-block backups (for fsck -b #) at:
    32, 2624, 5216, 7808
# mount /dev/md0c /mnt
# df /mnt
Filesystem 1K-blocks      Used      Avail Capacity  Mounted on
/dev/md0c      4846           2       4458      0%      /mnt
```

Se você não especificar o número da unidade com a opção `-u`, o [mdconfig\(8\)](#) utilizará o [md\(4\)](#) para selecionar automaticamente um dispositivo não utilizado. O nome da unidade alocada será impresso na saída padrão como o `md4`. Para mais detalhes sobre o [mdconfig\(8\)](#) visite a página de manual dele.

Nota: Desde o FreeBSD 5.1-RELEASE, o utilitário [bsdlable\(8\)](#) substituiu o velho programa [disklabel\(8\)](#). Com o [bsdlable\(8\)](#) algumas opções e parâmetros obsoletos foram retirados; no exemplo que foi dado a opção `-r` deveria ser removida. Para mais informações, por favor, visite a página de manual do [bsdlable\(8\)](#).

O utilitário [mdconfig\(8\)](#) é muito útil, entretanto ele precisa de diversas linhas de comando para criar um arquivo de imagem de um sistema de arquivos. O FreeBSD 5.0 também vem com uma ferramenta chamada [mdmfs\(8\)](#), este programa configura um disco [md\(4\)](#) utilizando o [mdconfig\(8\)](#), e coloca um sistema de arquivos UFS em uso com o [newfs\(8\)](#), e monta-o utilizando o [mount\(8\)](#). Por exemplo, se você quer criar e montar a mesma imagem de sistema de arquivos citada anteriormente, simplesmente digite como segue abaixo:

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records in
5120+0 records out
# mdmfs -F newimage -s 5m md0 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md0      4846           2       4458      0%      /mnt
```

Se você usar a opção `md` sem um número de unidade, o [mdmfs\(8\)](#) utilizará o [md\(4\)](#) que com suas funcionalidades selecionará automaticamente um dispositivo não utilizado. Para mais detalhes sobre o [mdmfs\(8\)](#), por favor, visitem a página de manual dele.

12.10.3. Sistemas de arquivos baseados em memória no FreeBSD 4.X

O controlador [md\(4\)](#) é simples, eficiente o suficiente para criar sistemas de arquivos em memória no FreeBSD 4.X. O [malloc\(9\)](#) é utilizado para alocar memória.

Simplesmente, pegue um sistema de arquivos que você tenha preparado como, por exemplo, o [vnconfig\(8\)](#), e:

Exemplo 12-7. Discos de memória md no FreeBSD 4.X

```
# dd if=newimage of=/dev/md0
5120+0 records in
5120+0 records out
# mount /dev/md0c /mnt
# df /mnt
Filesystem 1K-blocks      Used    Avail Capacity  Mounted on
/dev/md0c          4927         1     4532      0%    /mnt
```

Para mais detalhes, por favor, visitem a página de manual do [md\(4\)](#).

12.10.4. Sistemas de arquivos baseados em memória no FreeBSD 5.X

As mesmas ferramentas são utilizadas para os discos baseados em memória e os baseados em arquivos de imagem: [mdconfig\(8\)](#) ou [mdmfs\(8\)](#). O armazenamento para os sistemas de arquivos baseados em memória é alocado com o [malloc\(9\)](#).

Exemplo 12-8. Criando um novo disco baseado em memória com o mdconfig

```
# mdconfig -a -t malloc -s 5m -u 1
# newfs -U md1
/dev/md1: 5.0MB (10240 sectors) block size 16384, fragment size 2048
      using 4 cylinder groups of 1.27MB, 81 blks, 256 inodes.
      with soft updates
super-block backups (for fsck -b #) at:
   32, 2624, 5216, 7808
# mount /dev/md1 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md1          4846     2  4458      0%    /mnt
```

Exemplo 12-9. Criando um novo disco baseado em memória com o mdmfs

```
# mdmfs -M -s 5m md2 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md2          4846     2  4458      0%    /mnt
```

Ao invés de utilizar o sistema de arquivos alocado com o [malloc\(9\)](#) é possível utilizar swap, para isso apenas substitua a opção *malloc* com *swap* na linha de comando do [mdconfig\(8\)](#). O utilitário [mdmfs\(8\)](#) por padrão (sem a opção *-M*) cria um disco baseado em swap. Para mais detalhes, por favor, visitem a página de manual do [mdconfig\(8\)](#) e do [mdmfs\(8\)](#).

12.10.5. Desconectando um disco de memória do sistema

Quando um disco baseado em memória ou um baseado em um arquivo de imagem não é utilizado, você deve liberar todos os recursos para o sistema. A primeira coisa a fazer é desmontar o sistema de arquivos, então use o [mdconfig\(8\)](#) para desconectar o disco do sistema e os recursos serão liberados.

Por exemplo, para desconectar e liberar os recursos utilizados pelo `/dev/md4`, faça:

```
# mdconfig -d -u 4
```

É possível listar informações sobre o dispositivo [md\(4\)](#) configurado utilizando o comando `mdconfig -l`.

Para o FreeBSD 4.X, o [vnconfig\(8\)](#) é utilizado para desconectar os dispositivos. Por exemplo, para desconectar e liberar todos os recursos utilizados pelo `/dev/vn4`, faça:

```
# vnconfig -u vn4
```

12.11. Snapshots de sistemas de arquivos

Contribuição de Tom Rhodes.

O FreeBSD 5.0 oferece uma nova funcionalidade em conjunto com o [Soft Updates](#): snapshots de sistemas de arquivos.

O snapshots permitem ao usuário criar imagens de um sistema de arquivos especificado. Os arquivos de snapshot devem ser criados no sistema de arquivos em que foi executada a ação, e um usuário não pode criar mais de 20 snapshots por sistema de arquivos. Snapshots ativos são gravados em superbloco, tornando-os persistentes sobre as operações de desmontar e remontar que ocorrem nas reinicializações do sistema. Quando um snapshot não é mais necessário ele pode ser removido com o comando padrão [rm\(1\)](#). Os snapshots podem ser removidos em qualquer ordem, entretanto, nem todo o espaço deles será adquirido pois outros snapshots possivelmente utilizarão parte dos blocos que seriam liberados.

Durante a criação inicial, a opção `schg` (veja a página de manual do [chflags\(1\)](#)) é configurada para assegurar que não será permitido escrever no snapshot, nem mesmo para o `root`. O comando [unlink\(1\)](#) faz uma exceção para arquivos de snapshots desde que eles permitam ser removidos com a opção `schg` sendo utilizada, portanto não é necessário limpar a opção `schg` antes de remover um arquivo de snapshot.

Os snapshots são criados com o comando [mount\(8\)](#). Para colocar um snapshot do `/var` no arquivo `/var/snapshot/snap` utilize o seguinte comando:

```
# mount -u -o snapshot /var/snapshot/snap /var
```

Uma vez que um snapshot foi criado, este possui diversas utilidades:

- Alguns administradores utilizarão arquivos de snapshot com o propósito de backup devido a um snapshot poder ser transferido através de CD ou fita.
- A integridade do arquivo, o [fsck\(8\)](#) pode ser executado para um snapshot. Assumindo que aquele sistema de arquivos foi limpo ao ser montado, você sempre deveria ter resultados limpos e não modificados. Isto é essencialmente o que o processo do [fsck\(8\)](#) faz em *background*.
- Execute o utilitário [dump\(8\)](#) no snapshot. Será gerada uma cópia que será consistente ao sistema de arquivos e tendo os arquivos no mesmo horário dos do snapshot. O [dump\(8\)](#) também pode pegar um snapshot, criar uma cópia em uma imagem e então remover o snapshot em um comando, utilizando a opção *-L*.
- Execute o [mount\(8\)](#) no snapshot como se fosse uma imagem de um sistema de arquivos. Para executar o [mount\(8\)](#) no snapshot `/var/snapshot/snap` faça:
 - `# mdconfig -a -t vnode -f /var/snapshot/snap -u 4`
 - `# mount -r /dev/md4 /mnt`

Você agora pode caminhar pela hierarquia de arquivos do seu sistema de arquivos `/var` montado no seu `/mnt`. Tudo estará no mesmo estado de quando o snapshot foi criado. A única exceção é que alguns snapshots aparecerão com arquivos com tamanho zero. Quando o uso de um snapshot foi encerrado, você pode desmontá-lo com:

```
# umount /mnt
# mdconfig -d -u 4
```

Para mais informações sobre a opção *softupdates* e sistemas de arquivos em snapshots, incluindo artigos técnicos, você pode visitar o website de Marshall Kirk McKusick's em <http://www.mckusick.com>.

12.12. Cotas de uso para sistemas de arquivos

O uso de cotas é uma funcionalidade do sistema operacional que permite limitar a quantidade de espaço de disco e/ou o número de arquivos que um usuário ou membros de um grupo podem alocar em um sistema de arquivos. Isto é mais frequente em sistemas de compartilhamento de tempo onde é desejável limitar a quantidade de recursos que qualquer usuário ou um grupo de usuários podem alocar. Isto prevenirá um usuário ou um grupo de usuários de consumir todo o espaço em disco disponível.

12.12.1. Configurando seu sistema para utilizar cotas de disco

Antes de tentar utilizar cotas de disco, é necessário ter certeza que o suporte a cotas esteja configurado em seu kernel. Isto é feito adicionando as seguintes linhas ao seu arquivo de configuração do kernel:

```
options QUOTA
```

O kernel `GENERIC` não possui isto ativado, portanto, você terá que configurar, recompilar e instalar um kernel personalizado caso deseje utilizar cotas de disco. Por favor, visite a seção [Capítulo 9](#) para ter mais informações sobre configuração de kernel.

Em seguida, você precisará ativar as cotas de disco no `/etc/rc.conf`. Isto é feito adicionando a linha:

```
enable_quotas="YES"
```

Para controlar o processo de inicialização de cotas, há uma variável de configuração adicional disponível. Normalmente, na inicialização é checada a integridade das cotas de cada um dos sistemas de arquivos pelo programa [quotacheck\(8\)](#). O [quotacheck\(8\)](#) facilita a certificação que dos dados das definições de cotas refletem adequadamente os dados no sistema de arquivos. Isto é um processo que consome tempo que irá afetar significativamente o tempo que o seu sistema leva para iniciar. Se você gostaria de passar adiante neste passo há uma variável com este propósito para ser configurada no `/etc/rc.conf`:

```
check_quotas="NO"
```

Se você estiver utilizando um sistema FreeBSD anterior a versão 3.2-RELEASE, a configuração é simples e consiste em apenas uma variável. Configure o `/etc/rc.conf` como segue:

```
check_quotas="YES"
```

Finalmente, você precisará editar o arquivos `/etc/fstab` para ativar as cotas de disco para cada um dos sistemas de arquivos. Aqui é onde se pode ativar o uso de cotas para usuário ou para grupos de usuários para todos os sistemas de arquivos.

Para ativar o uso de cotas por usuário em um sistema de arquivos, adicione a opção *userquota* no campo de opções de uma entrada no `/etc/fstab` para o sistema de arquivos que você quer ativar as cotas. Por exemplo:

```
/dev/dals2g    /home    ufs rw,userquota 1 2
```

Similar é o processo de ativação de cotas para grupos de usuários. Utilize a opção *groupquota* ao invés de *userquota*. Para ativar ambos os tipos de cotas, mude a entrada para:

```
/dev/dals2g    /home    ufs rw,userquota,groupquota 1 2
```

Por padrão, os arquivos de cota são armazenados no diretório raiz de cada sistema de arquivos com o nome de `quota.user` e `quota.group` para cotas de usuário e de grupos respectivamente. Veja a página de manual do [fstab\(5\)](#) para mais informações. Embora a página de manual do [fstab\(5\)](#) diga que você pode especificar um local alternativo para os arquivos de cota, isto não é recomendado devido aos vários utilitários de gerenciamento de cotas não suportam esta possibilidade, o que ocasionaria em erro devido a estes utilitários buscarem pelos arquivos na raiz do sistema de arquivos.

Neste ponto você deve reiniciar seu sistema com seu novo kernel. O `/etc/rc` automaticamente executará os comandos apropriados para criar os arquivos de cota iniciais para todos os sistemas de arquivos que tiveram cotas ativadas no `/etc/fstab`, portanto, não é necessário criar qualquer arquivo de cota com tamanho zero manualmente.

No curso normal das operações você não deveria ser solicitado a executar os comandos [quotacheck\(8\)](#), [quotaon\(8\)](#), ou [quotaoff\(8\)](#) manualmente. Entretanto, você pode querer ler as páginas de manuais somente para estar familiarizado com a operação destes comandos.

12.12.2. Configurando os limites das cotas

Uma vez que você configurou e ativou o sistema de cotas, verifique se este está realmente funcional. Uma maneira fácil de fazer isso é executando:

```
# quota -v
```

Você deve ver uma linha de sumário sobre a utilização de disco e sobre os limites de cotas para cada um dos sistemas de arquivos que as cotas estão ativas.

Você agora está pronto para iniciar a definição dos limites de cotas com o comando [edquota\(8\)](#).

Você tem diversas opções para auxiliar em como você define os limites de quantidade de espaço em disco que um usuário ou um grupo de usuários pode alocar, e quantos arquivos eles podem criar. Você deve limitar as alocações se baseando no espaço em disco (cotas de blocos) ou pelo número de arquivos (cotas por *inode*) ou uma combinação de ambos. Cada um destes limites pode ser atingido em duas categorias: limites físicos e lógicos.

Um limite físico não pode ser excedido. Uma vez que o usuário atinge seu limite físico ele não poderá fazer alocação alguma no sistema de arquivos em questão. Por exemplo, se o usuário tem um limite físico de 500 blocos num sistema de arquivos e está utilizando atualmente o 490 blocos, o usuário poderá alocar somente 10 blocos adicionais. A tentativa de alocar 11 blocos adicionais resultará em falha.

Limites lógicos, por outro lado, podem ser excedidos por uma quantidade limitada de tempo. Este período é conhecido como período de favor (*grace time*), que é de uma semana, por padrão. Se um usuário estiver acima do seu limite lógico por um tempo maior que o tempo de favor suas cotas tornarão-se cotas de limite físico e nenhuma outra alocação será permitida. Quando um usuário apaga seus arquivos liberando espaço suficiente para estar abaixo dos limites das suas cotas lógicas, o tempo de favor é reiniciado.

À seguir, temos um exemplo do que você pode ver quando executar o comando [edquota\(8\)](#). Quando o comando [edquota\(8\)](#) é chamado, você é direcionado ao editor especificado pela variável de ambiente `EDITOR` ou ao editor `vi` se a configuração da variável `EDITOR` não foi alterada, para permitir que você edite os limites de cotas.

```
# edquota -u test
Quotas for user test:
/usr: blocks in use: 65, limits (soft = 50, hard = 75)
      inodes in use: 7, limits (soft = 50, hard = 60)
/usr/var: blocks in use: 0, limits (soft = 50, hard = 75)
          inodes in use: 0, limits (soft = 50, hard = 60)
```

Você normalmente verá duas linhas para cada sistema de arquivos que tenham cotas ativadas. Uma linha para os limites de blocos e uma para os limites de *inode*. Simplesmente, modifique os valores que você desejar para modificar os limites de cotas. Por exemplo, para aumentar o limite de blocos deste usuário de 50 e o limite físico de 75 para, respectivamente, 500 e 600, mude:

```
/usr: blocks in use: 65, limits (soft = 50, hard = 75)
```

para:

```
/usr: blocks in use: 65, limits (soft = 500, hard = 600)
```

Os novos limites de cotas será definidos assim que você sair do editor.

Algumas vezes é desejável definir limites de cotas para uma faixa de UIDs. Isso pode ser feito pelo uso da opção *-p* no comando [edquota\(8\)](#). Primeiro, defina o limite de cotas desejado para um usuário, e então execute `edquota -p protouser startuid-enduid`. Por exemplo, se o usuário `test` tem os limites de cotas desejado, o seguinte comando pode ser utilizado para duplicar estes limites para os usuários com UIDs de 10,000 até 19,999, com:

```
# edquota -p test 10000-19999
```

Para mais informações veja a página de manual do [edquota\(8\)](#).

12.12.3. Verificando os limites de cotas e a utilização de disco

Você pode utilizar tanto o comando [quota\(1\)](#) quanto o [repquota\(8\)](#) para verificar os limites de cotas e a utilização do espaço de disco. O comando [quota\(1\)](#) pode ser utilizado para verificar individualmente as cotas e a utilização de disco de um usuário ou de um grupo de usuários. Um usuário pode apenas examinar suas cotas próprias, e as cotas para o grupo no qual ele é membro. Só o superusuário pode ver as cotas de todos os usuários e de todos os grupos de usuários. O comando [repquota\(8\)](#) pode ser utilizado para conseguir um sumário de todas as cotas e utilização de disco para os sistemas de arquivos com cotas ativas.

À seguir, temos um exemplo de saída do comando `quota -v` para um usuário que tem limites definidos em dois sistemas de arquivos.

```
Disk quotas for user test (uid 1002):
      Filesystem  blocks  quota  limit  grace  files  quota
limit  grace
      /usr        65*    50     75    5days    7     50
60
      /usr/var    0      50     75           0     50
60
```

No sistema de arquivos `/usr`, no exemplo dado, o usuário está acima do seu limite lógico de 50 blocos em 15 blocos e tem 5 dias restantes para que o tempo de favor acabe. Note o asterisco `*`, ele indica que o usuário está acima dos seus limites de cotas.

Normalmente, os sistemas de arquivos que o usuário não fez utilização de espaço em disco não serão mostrados na saída do comando `quota(1)`, mesmo se este tiver limites de cotas definidos neste sistema de arquivos. A opção `-v` irá listar estes sistemas de arquivos, como o `/usr/var` no exemplo em discussão.

12.12.4. Uso de cotas em sistemas de arquivos NFS

Cotas podem ser utilizadas pelo subsistema do servidor NFS. O processo em segundo plano `rpc.rquotad(8)` faz com que as informações de cota estejam disponíveis ao comando `quota(1)` nos clientes NFS, permitindo aos usuários ver suas estatísticas de cotas em suas máquinas.

Ative o `rpc.rquotad` no `/etc/inetd.conf` como abaixo:

```
rquotad/1      dgram rpc/udp wait root /usr/libexec/rpc.rquotad
rpc.rquotad
```

Agora, reinicie o `inetd`:

```
# kill -HUP `cat /var/run/inetd.pid`
```

12.13. Encriptando partições de um disco

Contribuição de Lucky Green.

O FreeBSD oferece excelentes proteções em tempo de uso para controlar acesso não autorizado aos dados. Permissões de arquivos e Controle de Acesso Mandatório (MAC) (Mandatory Access Control, veja [Seção 10.12](#)) ajudam a prevenir acessos não autorizados de terceiros tentando acessar os dados enquanto o sistema operacional está ativo e o computador ligado. Entretanto, as proteções fornecidas pelo sistema operacional são irrelevantes se um atacante tem acesso físico ao computador, podendo, simplesmente, mover os dispositivos para um outro sistema para copiar e analisar os dados significantes.

Não nos preocupando de como o atacante adquiriu acesso e apossou-se do disco rígido ou de como ele desligou o computador, a **Encriptação de Disco Baseada em Geometria** (`gbde`, GEOM Based Disk Encryption) pode proteger os dados nos sistemas de arquivos contra o mais motivado dos atacantes com recursos significantes. Em contrapartida aos desajeitados métodos de encriptação utilizado em arquivos individuais, o **gbde** encripta, de maneira transparente, todo o sistema de arquivos. Nenhum texto sem encriptação chega a ser armazenado no disco rígido.

12.13.1. Ativando o gbde no kernel

1. Torne-se o usuário root

Configurar o **gbde** requer privilégios de superusuário.

```
% su -  
Password:
```

2. Verifique a versão do sistema operacional

O [gbde\(4\)](#) requer a utilização do FreeBSD 5.0 ou versão superior.

```
# uname -r  
5.0-RELEASE
```

3. Adicione o suporte ao gbde(4) no arquivo de configuração do kernel

Utilizando seu editor de textos favorito, adicione a seguinte linha no seu arquivo de configuração de kernel:

```
options GEOM_BDE
```

Após configurar, recompile e instale kernel do FreeBSD. Este processo está descrito em [Capítulo 9](#).

Reinicie utilizando o novo kernel.

12.13.2. Preparando a encriptação do disco rígido

O exemplo que segue assume que você adicionou um novo disco rígido no seu sistema e este terá uma partição encriptada. Esta partição será montada como `/private`. O **gbde** também pode ser utilizado para encriptar o `/home` e o `/var/mail`, mas isso requer instruções mais complexas que excedem o escopo desta introdução.

1. Adicionando um novo disco rígido

Instale o novo disco no sistema como explicado em [Seção 12.3](#). Para o propósito deste exemplo, uma nova partição foi adicionada a este disco como `/dev/ad4s1c`. Os dispositivos `/dev/ad0s1*` representam partições padrões do FreeBSD no sistema do exemplo.

```
# ls /dev/ad*  
/dev/ad0          /dev/ad0s1b      /dev/ad0s1e      /dev/ad4s1  
/dev/ad0s1       /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c  
/dev/ad0s1a     /dev/ad0s1d     /dev/ad4
```

2. Crie um diretório para o gbde armazenar seus arquivos de trava

```
3. # mkdir /etc/gbde
```


Os arquivos de trava do **gbde** contém informações que o **gbde** precisa para acessar partições encriptadas. Sem ter acesso aos arquivos de trava o **gbde** não será capaz de descriptar os dados contidos na partição encriptada sem intervenção manual significanete, o que não é suportado pelo programa. Cada partição encriptada usa arquivos de travas separados.

4. Inicialize a partição gbde

Uma partição **gbde** deve ser iniciada antes de poder ser utilizada. Este processo precisa ser realizado pelo menos uma vez:

```
# gbde init /dev/ad4s1c -i -L /etc/gbde/ad4s1c
```

O [gbde\(8\)](#) abrirá seu editor, permitindo que você configure diversas opções fornecidas num modelo. Para utilizar com UFS1 ou UFS2, configure o `sector_size` para 2048:

```
$FreeBSD: src/sbin/gbde/template.txt,v 1.1 2002/10/20 11:16:13
phk Exp $
#
# Sector size is the smallest unit of data which can be read or
written.
# Making it too small decreases performance and decreases
available space.
# Making it too large may prevent filesystems from working. 512
is the
# minimum and always safe. For UFS, use the fragment size
#
sector_size      =      2048
[...]
```

O [gbde\(8\)](#) perguntará duas vezes a frase de encriptação utilizada para assegurar os dados. A frase de encriptação deve ser a mesma em ambas as vezes. As habilidades do **gbde** para proteger seus dados depende inteiramente da qualidade da frase de encriptação que você escolhe. [\[1\]](#)

O comando `gbde init` cria um arquivo de trava para a sua partição **gbde** que neste exemplo é armazenado como `/etc/gbde/ad4s1c`.

CuidadoOs arquivos de trava do **gbde** *devem* ser copiados junto ao backup do conteúdo das partições encriptadas. Apagando um arquivo de trava sozinho pode não prevenir que um atacante determinado consiga descriptar a partição **gbde**, sem o arquivo de trava, mesmo o proprietário dos dados não conseguirá acessar os dados na partição encriptada sem uma quantidade significativa de trabalho que é totalmente isento suporte, tanto do [gbde\(8\)](#) quanto de seu desenvolvedor.

5. Anexando a partição encriptada ao kernel

```
6. # gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c
```

Você será solicitado a fornecer a frase de encriptação que você selecionou durante a inicialização da partição encriptada. O novo dispositivo encriptado será mostrado em `/dev` como `/dev/device_name.bde`:

```
# ls /dev/ad*
/dev/ad0          /dev/ad0s1b      /dev/ad0s1e      /dev/ad4s1
/dev/ad0s1        /dev/ad0s1c      /dev/ad0s1f      /dev/ad4s1c
/dev/ad0s1a       /dev/ad0s1d      /dev/ad4          /dev/ad4s1c.bde
```

7. Criar um sistema de arquivos encriptado no dispositivo

Uma vez que o dispositivo encriptado foi anexado ao kernel, você pode criar um sistema de arquivos nele. Para criar um sistema de arquivos no dispositivo encriptado use o [newfs\(8\)](#). Considerando ainda que é muito mais rápido de iniciar um novo sistema de arquivos UFS2 que um velho UFS1, utilizando o [newfs\(8\)](#) com a opção `-O2` é uma recomendação.

Nota: A opção `-O2` é o padrão para o FreeBSD 5.1-RELEASE e versões mais novas.

```
# newfs -U -O2 /dev/ad4s1c.bde
```

Nota: O comando [newfs\(8\)](#) deve ser executado na partição **gbde** anexada que é identificada pela extensão `*.bde` no nome do dispositivo.

8. Montar a partição encriptada

Crie um ponto de montagem para o sistema de arquivos encriptado.

```
# mkdir /private
```

Monte o sistema de arquivos encriptado.

```
# mount /dev/ad4s1c.bde /private
```

9. Verifique se o sistema de arquivos encriptado está utilizável

O sistema de arquivos encriptado agora deve poder ser visível na listagem do comando [df\(1\)](#) e deve estar disponível para uso.

```
% df -H
Filesystem      Size    Used    Avail Capacity  Mounted on
/dev/ad0s1a     1037M    72M    883M      8%      /
/devfs           1.0K    1.0K      0B    100%    /dev
/dev/ad0s1f      8.1G    55K    7.5G      0%    /home
/dev/ad0s1e     1037M    1.1M    953M      0%    /tmp
/dev/ad0s1d      6.1G    1.9G    3.7G     35%    /usr
/dev/ad4s1c.bde  150G    4.1K    138G      0%    /private
```

12.13.3. Montando um sistema de arquivos encriptado existente

Após cada início do sistema, qualquer sistema de arquivos encriptado deve ser reanexado ao kernel, verificado se possui erros, e montado, isso tudo antes do sistema de arquivos poder ser utilizado. Os comandos necessário devem ser executados pelo usuário `root`.

1. Anexar uma partição gbde ao kernel

```
2. # gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c
```

Você será solicitado a fornecer a frase de encriptação que você selecionou durante a inicialização da partição gbde encriptada.

3. Checar erros no sistema de arquivos

Considerando que os sistemas de arquivos encriptados ainda não podem ser listados no `/etc/fstab` para a montagem automática, o sistema de arquivos deve ser checado através da execução manual do [fsck\(8\)](#) antes de ser montado.

```
# fsck -p -t ffs /dev/ad4s1c.bde
```

4. Montar um sistema de arquivos encriptado

```
5. # mount /dev/ad4s1c.bde /private
```

O sistema de arquivos encriptado agora está disponível para o uso.

12.13.3.1. Montagem automática de sistemas de arquivos encriptados

É possível criar um script que automaticamente, anexe, verifique, e monte a partição encriptada, mas por motivos de segurança o script não deve conter a frase de encriptação do [gbde\(8\)](#). Ao invés de fazer isso, é recomendado que este script seja executado manualmente enquanto se provê a frase de encriptação via console ou [ssh\(1\)](#).

12.13.4. Proteções de encriptação empregadas pelo gbde

O [gbde\(8\)](#) encripta um setor utilizando AES de 128-bit em modo CBC. Cada setor do disco é encriptado com uma chave AES diferente. Para mais informações sobre o projeto de encriptação do **gbde**, incluindo como as chaves para os setores são derivadas da frase de encriptação fornecida pelo usuário, veja a página de manual do [gbde\(4\)](#).

12.13.5. Informações de compatibilidade

O [sysinstall\(8\)](#) é incompatível com dispositivos encriptados do **gbde**. Todos os dispositivos `*.bde` devem ser desconectados do kernel antes da execução do [sysinstall\(8\)](#) ou haverá uma parada de execução durante a fase inicial do programa onde é feita uma pesquisa sobre os dispositivos disponíveis. Para desconectar o dispositivo encriptado utilizado no exemplo dado, use o seguinte comando:

```
# gbde detach /dev/ad4s1c
```

Note também que como o [vinum\(4\)](#) não utiliza o subsistema [geom\(4\)](#), ele também não possibilita o uso do **gbde** com volumes **vinum**.

Notas

[1] Para dicas em como selecionar uma frase de encriptação segura e fácil de lembrar, veja o website [Diceware Passphrase](#).

Capítulo 13. O Gerenciador de Volumes Vinum

Índice

- 13.1. [Sinótese](#)
- 13.2. [Discos são muito pequenos](#)
- 13.3. [Gargalo de acesso](#)
- 13.4. [Integridade de dados](#)
- 13.5. [Objetos do Vinum](#)
- 13.6. [Alguns exemplos](#)
- 13.7. [Nomenclatura de objetos](#)
- 13.8. [Configurando Vinum](#)
- 13.9. [Usando Vinum para o Sistema de Arquivos Raiz](#)

13.1. Sinótese

Não importa quais discos você tem, eles sempre terão limitações:

- Eles podem ser pequenos demais.
- Eles podem ser lentos demais.
- Eles podem ser pouco seguros.

13.2. Discos são muito pequenos

Originally written by Greg Lehey.

Vinum o chamado *Gerenciador de Volumes* é o disco virtual que trata destes três problemas, vamos analisa-los com mais detalhes. Soluções para estes problemas tem sido propostas e implementadas:

os discos estão ficando maiores, mas os requisitos para armazenamento de dados também. Frequentemente você vai querer um sistema de arquivos maior do que o disco que você tem disponível. Reconhecidamente, este problema já não é tão agudo com era a dez anos atrás, mas ele ainda existe. Alguns sistemas resolveram isso criando um dispositivo abstrato onde armazenam seus dados em um número de discos.

13.3. Gargalo de acesso

Sistemas modernos frequentemente precisam acessar dados de uma maneira incrivelmente rápida. Por exemplo, grandes servidores de FTP e HTTP podem manter milhares de sessões ao mesmo tempo e podem ter múltiplas conexões de 100 Mbit/s com o mundo externo, bem além das taxas de transmissão da maioria dos discos.

Os discos atuais podem transferir dados sequencialmente em até 70 MB/s, mas este valor é de pouca importância em um ambiente onde muitos processos independentes acessam um disco, onde eles talvez atinjam apenas uma fração desse valor. Nesses casos é mais importante observar o problema do ponto de vista do subsystem: o parâmetro importante é a carga que uma transferência coloca no subsystem, em outras palavras, o tempo em que a transferência ocupa os discos envolvidos.

Em qualquer operação de transferência, o disco deve primeiramente posicionar as cabeças de leitura e esperar até que o primeiro setor passe em baixo delas e depois simplesmente executar a transferência. Esses atos podem ser considerados como se fossem atômicos: não faz o menor sentido interromper eles.

Considere uma transferência típica de aproximadamente 10 kB: a geração atual de discos de alta performance podem posicionar as cabeças em uma média de 3.5 ms. Os discos mais rápidos giram em 15.000 rpm, então, a rotação latente média (metade de uma revolução) é de 2 ms. Em 70 MB/s, a transferência propriamente dita, leva em torno de 150 μ s, o que é quase nada comparado com o tempo de posicionamento. Em tal caso, a taxa efetiva de transferência fica um pouco maior que 1 MB/s e é altamente dependente no tamanho do que vai ser transferido, é claro.

A solução óbvia e tradicional para esse gargalo é "more spindles": ele usa vários discos pequenos com a mesma capacidade de armazenamento em vez de usar um outro disco grande. Cada disco é capaz de se posicionar e transferir dados de maneira independente, então o ritmo de transferência efetivo, cresce a uma taxa próxima ao número de discos usados.

O ritmo de transferência exato é, claramente, menor que o número de discos envolvidos: apesar de cada disco ser capaz de realizar transferências em paralelo, não existe nenhuma forma de se assegurar que as solicitações estão distribuídas de maneira uniforme entre os discos. Inevitavelmente a carga no disco será maior do que no outro.

A distribuição equilibrada da carga nos discos é fortemente dependente do modo em que os dados são compartilhados entre eles. No próximo debate, seria conveniente visualizar um disco como se fosse um grande número de setores de dados mapeados por números, como as páginas de um livro. O método mais simples seria dividir o disco virtual em pequenos grupos de setores consecutivos do tamanho do disco físico individual e depois armazená-los desse jeito, semelhante a pegar um livro grande e dividi-lo em pequenas sessões. Esse método é chamado de *concatenação* e tem a vantagem que os discos não precisam ter nenhum tipo de relacionamento quanto a tamanho. Funciona bem quando o acesso ao disco virtual é dividido de maneira igual entre os seus espaços mapeados. Quando o acesso é concentrado uma área menor, o

aumento no desempenho é menos acentuado. A [Figura 13-1](#) ilustra a seqüência na qual as unidades de armazenamento são alocadas em uma organização concatenada.

Figura 13-1. Organização Concatenada

Disk 1	Disk 2	Disk 3	Disk 4
0	6	10	12
1	7	11	13
2	8		14
3	9		15
4			16
5			17

Um mapeamento alternativo é dividir o espaço de endereçamento em componentes menores, de tamanhos iguais e armazená-los seqüencialmente em discos diferentes. Por exemplo, os primeiros 256 setores poderiam ser armazenados no primeiro disco, os 256 setores seguintes no próximo disco e assim por diante. Depois de preencher o último disco o processo se repete até que os discos estejam cheios. Esse mapeamento é chamado de *striping* ou RAID-0. [1]. O *striping* exige um esforço grande para localizar os dados e isso causa carga adicional de I/O onde a transferência é distribuída para diversos discos, mas pode também fornecer uma carga constante entre os discos. [Figura 13-2](#) ilustra uma seqüência em que as unidades de armazenamento são alocadas em uma organização *striped*.

Figura 13-2. Organização Striped

Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23

Notas

[1] RAID (Redundant Array of Inexpensive Disks) significa *Vetor redundante de discos baratos* e oferece várias formas de tolerância de erros, apesar de que um termo um pouco equivocado, pois não fornece redundância.

13.4. Integridade de dados

O outro problema com os discos atuais é que eles não são confiáveis. Apesar do crescimento significativo na credibilidade dos discos nos últimos anos, esses ainda são um dos componentes do núcleo de servidor com a maior possibilidade de falhar. E quando eles falham, os resultados podem ser catastróficos: substituir um disco e recuperar seus dados pode levar dias.

O método tradicional de abordar esse problema tem sido o *espelhamento*, mantendo duas cópias dos dados em dois lugares físicos distintos. Desde da aparição dos níveis de RAID, essa técnica de espelhamento também é chamada de RAID nível 1 ou RAID-1. Qualquer escrita feita no volume é realizada nas duas localidades, a leitura pode ser feita a partir de uma delas, então se o disco falhar, os dados ainda estarão disponíveis no outro.

O espelhamento tem dois problemas:

- O preço. Exige duas vezes mais armazenamento de disco do que uma solução sem redundância.
- O impacto de performance. A escrita deve ser feita para os dois discos, devido a isso eles usam duas vezes mais largura de banda do que um volume não espelhado. Na leitura não há queda de performance: chega até a parecer que está mais rápido.

Uma solução alternativa é a *paridade*, implementada nos níveis 2, 3, 4 e 5 do RAID. Dentre esses, RAID-5 é o mais interessante. Como implementado no Vinum, é uma variante na organização *striped* onde dedica um bloco de cada *slice* à paridade dos outros blocos. Conforme implementado no Vinum, um plex RAID-5 é parecido com um plex *striped*, exceto que ele implementa o RAID-5 incluindo um bloco de paridade em cada *slice*. Como exigido pelo RAID-5, o local do bloco de paridade muda de um *slice* para outro. Os números nos blocos de dados indicam o número relativo do bloco.

Figura 13-3. Organização RAID-5

Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	Parity
3	4	Parity	5
6	Parity	7	8
Parity	9	10	11
12	13	14	Parity
15	16	Parity	17

Comparado com espelhamento, RAID-5 tem a vantagem de requerer significativamente menos espaço de armazenameto. O acesso de leitura é parecido com o utilizado nas organizações *striped*, mas o acesso à escrita é bem reduzido, chega a até 25% da performance de leitura aproximadamente. Se uma partição falha, o vetor pode continuar a operar em modo degradado: a leitura das partições acessíveis restantes continua normalmente, mas a leitura da partição danificada é recalculada pelo bloco correspondente de todos os discos restantes.

13.5. Objetos do Vinum

Para que o Vinum possa solucionar esse problema, ele implementa uma hierarquia de objetos em quatro níveis:

- O objeto mais visível é o disco virtual, chamado de *volume*. Volumes tem essencialmente as mesmas propriedades de uma partição UNIX®, contudo tem algumas pequenas diferenças. Eles não tem nenhuma limitação de tamanho.
- Volumes são compostos de *plexes*, no qual representa o espaço total de endressamento de um volume. Dessa forma, esse nível na hierarquia fornece redundância. Pense em plexes como uma partição individual em um vetor clonado, cada um contendo os mesmos dados.
- Desde que o Vinum exista no sistema de armazenamento de discos UNIX, seria possível usar partições UNIX como um bloco de construção para plexes de múltiplos discos, mas no final isso acaba sendo muito inflexível: o número de partições de um disco UNIX é limitado. Ao contrário, Vinum subdivide uma única partição UNIX (os *discos do vinum*) em áreas contínuas chamadas *subdiscos*, os quais são usados como blocos de construção para plexes.
- Subdiscos residem em discos do Vinum. Atualmente partições UNIX. Com a exceção de uma pequena área no começo do disco, a qual é usada para armazenamento de informações de configuração e estado, o disco todo fica disponível para armazenamento de dados.

As sessões seguintes descrevem o modo com que esses objetos fornecem a funcionalidade requerida pelo Vinum.

13.5.1. Considerações de um tamanho de volume

Plexes podem ter muitos subdiscos espalhados por todas as partições nas configurações do vinum. Como resultado, o tamanho de uma prtição original não limta o tamanho nem de um plex e nem de um volume.

13.5.2. Armazenamento de Dados Redundante

Vinum implementa espelhamento anexando múltiplos plexes a um volume. Cada plex é a representação dos dados de um volume. Um volume pode conter de um a oito plexes.

Apesar de um plex representar os dados completos de um volume, é possível que partes da representação estejam fisicamente ausentes, seja por planejamento (por não definir um subdisco como parte de um plex), ou por acidente (como resultado de uma falha no disco). Contanto que, pelo menos um plex possa fornecer dados para a extensão completa de endereço do volume, o sistema é completamente funcional.

13.5.3. Questões de performance

Vinum implementa no nível do plex tanto concatenação quanto striping:

- Um *plex concatenado* usa um espaço de endereçamento de cada subdisco por vez.
- Um *plex striped* distribui os dados entre cada subdisco. Todos os subdiscos têm que ter o mesmo tamanho, e é preciso que existam pelo menos dois subdiscos para que eles possam ser distinguidos a partir de um plex concatenado.

13.5.4. Qual organização plex?

A versão do Vinum suprida com o FreeBSD 5.3 implementa dois tipos de plexes:

- Plexes concatenados são os mais flexíveis: eles podem conter infinitos números de subdiscos e os subdiscos podem ser de tamanhos diferentes. Os plexes podem ser extendidos adicionando subdiscos. Eles requerem menos tempo de CPU do que plexes *striped*, mas a diferença em sobrecarga de CPU não dá para ser medida. Por outro lado, eles são mais suscetíveis a *hot spots*, onde um disco é muito ativo e outros estão ociosos.
- A grande vantagem de plexes *striped* (RAID-0) é que eles reduzem os *hot spots*: escolhendo o melhor tamanho de *stripe* (mais ou menos 256 kB), você pode igualar a carga nos discos componentes. As desvantagens dessa escolha são códigos (fracionalmente) mais complexos e restrições em subdiscos: eles tem que ser todos do mesmo tamanho, e estender um plex através da adição de um novo subdisco é tão complicado que Vinum atualmente nem o implementa. Vinum impõe uma restrição trivial adicional: Um plex *striped* precisa ter pelo menos dois subdiscos, pois caso contrário seria indistinguível de um plex concatenado.

[Tabela 13-1](#) resume as vantagens e desvantagens de cada organização plex.

Tabela 13-1. Organizações do Plex no Vinum

Tipo de Plex	Mínimo de subdiscos	Pode adicionar subdiscos	Deve ser de igual tamanho	Aplicação
concatenado	1	sim	não	Grande armazenamento de dados com máxima flexibilidade de posicionamento e performance

Tipo de Plex	Mínimo de subdiscos	Pode adicionar subdiscos	Deve ser de igual tamanho	Aplicação
				moderada
<i>striped</i>	2	não	sim	Alta performance em combinação com um acesso altamente concorrido

13.6. Alguns exemplos

Vinum mantém uma *base de dados de configuração* onde descreve os objetos conhecidos por um sistema individual. Inicialmente, o usuário cria o banco de dados de configuração a partir de um ou mais arquivos de configuração com a ajuda do utilitário [vinum\(8\)](#). Vinum guarda uma cópia das configurações de seu banco de dados em cada *slice* de disco (a qual Vinum chama de *dispositivo*) sobre seu controle. Essa base de dados é atualizada em cada mudança de estado, então, precisamente recupera o estado de cada objeto do Vinum.

13.6.1. O Arquivo de Configuração

O arquivo de configuração descreve os objetos individuais do Vinum. A definição de um simples volume poderia ser:

```
drive a device /dev/da3h
volume myvol
plex org concat
sd length 512m drive a
```

Esse arquivo descreve quatro objetos do Vinum:

- A linha *drive* descreve uma partição de disco (*drive*) e sua localidade relativa ao *hardware* subjacente. É dado um nome simbólico *a*. Essa separação entre nomes simbólicos e nomes de dispositivos permite que os discos sejam movidos de um lado a outro sem confusão.
- A linha *volume* descreve um volume. O único atributo requerido é o nome, nesse caso *myvol*.
- A linha *plex* define um plex. O único parâmetro requerido é a organização, nesse caso *concat*. Nenhum nome é necessário: o sistema irá automaticamente gerar um nome a partir do volume adicionando o sufixo *.px*, onde *x* é o número de plexes no volume. Então, esse plex terá o nome *myvol.p0*.
- A linha *sd* descreve um subdisco. As especificações mínimas são o nome do disco no qual será armazenado, e o tamanho do subdisco. Como nos plexes, nenhum nome é necessário: o sistema delega automaticamente nomes provenientes dos plexes adicionando o sufixo *.sx*, onde *x* é o número de subdiscos do plex. Então Vinum delega a esse subdisco o nome de *myvol.p0.s0*.

Depois de processar esse arquivo, [vinum\(8\)](#) gera a seguinte saída:

```
# vinum -> create config1
Configuration summary
Drives:          1 (4 configured)
Volumes:         1 (4 configured)
Plexes:          1 (8 configured)
Subdisks:        1 (16 configured)

D a                               State: up           Device /dev/da3h
Avail: 2061/2573 MB (80%)

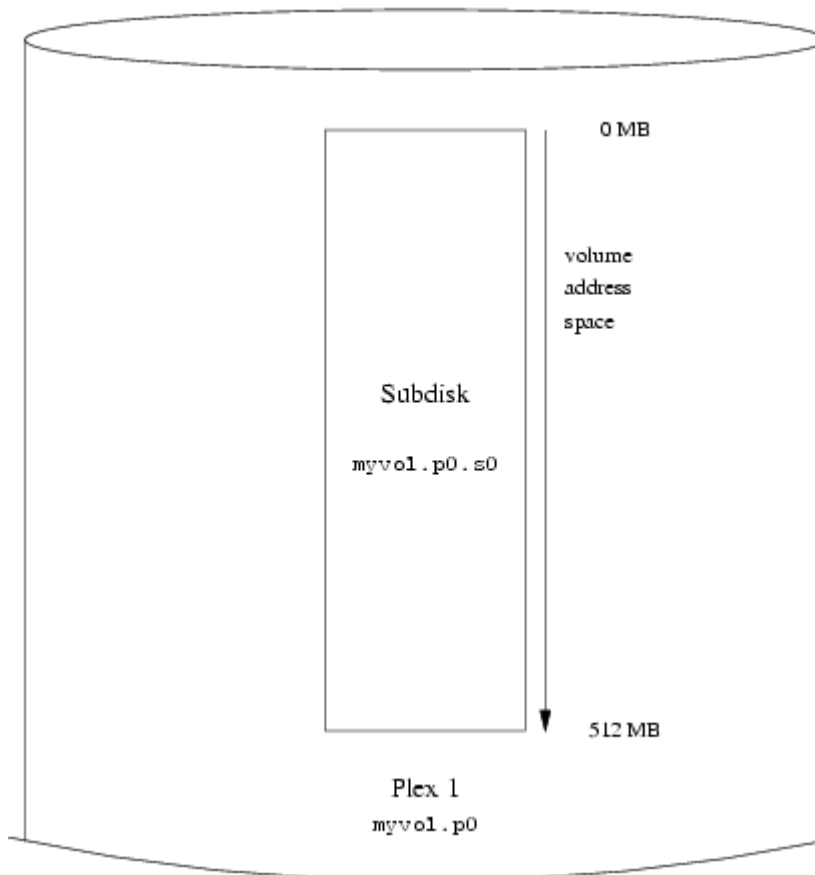
V myvol                           State: up           Plexes:          1 Size:
512 MB

P myvol.p0                         C State: up         Subdisks:        1 Size:
512 MB

S myvol.p0.s0                      State: up           PO:             0 B Size:
512 MB
```

Essa saída mostra um pequeno formato de listagem do [vinum\(8\)](#). É representado graficamente na [Figura 13-4](#).

Figura 13-4. Um Volume Simples de Vinum



Essa figura, e as outras que seguem, representam um volume, que contém plexes, e estes, por sua vez possuem subdiscos. Nesse exemplo trivial, o volume contém um plex, e o plex contém um subdisco.

Esse volume em particular não tem nenhuma vantagem específica sobre uma partição de um disco convencional. Contém um único plex, então, não é redundante. O plex contém um único subdisco, então a alocação de dados não se difere de uma partição de disco convencional. As sessões seguintes exemplificam diversos métodos de configuração mais interessantes.

13.6.2. Crescimento na Confiança: Espelhamento

A confiança em um volume pode ser aumentada com espelhamento. Quando planejando um espelhamento de volume, é importante se assegurar que o subdisco de cada plex esta em partições diferentes, para que uma falha de disco não derrube os dois plexes. A seguinte configuração espelha um volume:

```
drive b device /dev/da4h
volume mirror
plex org concat
sd length 512m drive a
plex org concat
sd length 512m drive b
```

Nesse exemplo, não foi necessário especificar uma definição de uma partição *a* novamente, a partir do momento que o vinum armazena todos os objetos em sua base de dados de configuração. Depois de processar essa definição, a configuração se parece com:

```
Drives:          2 (4 configured)
Volumes:         2 (4 configured)
Plexes:          3 (8 configured)
Subdisks:        3 (16 configured)

D a              State: up          Device /dev/da3h
Avail: 1549/2573 MB (60%)
D b              State: up          Device /dev/da4h
Avail: 2061/2573 MB (80%)

V myvol          State: up          Plexes:          1 Size:
512 MB
V mirror         State: up          Plexes:          2 Size:
512 MB

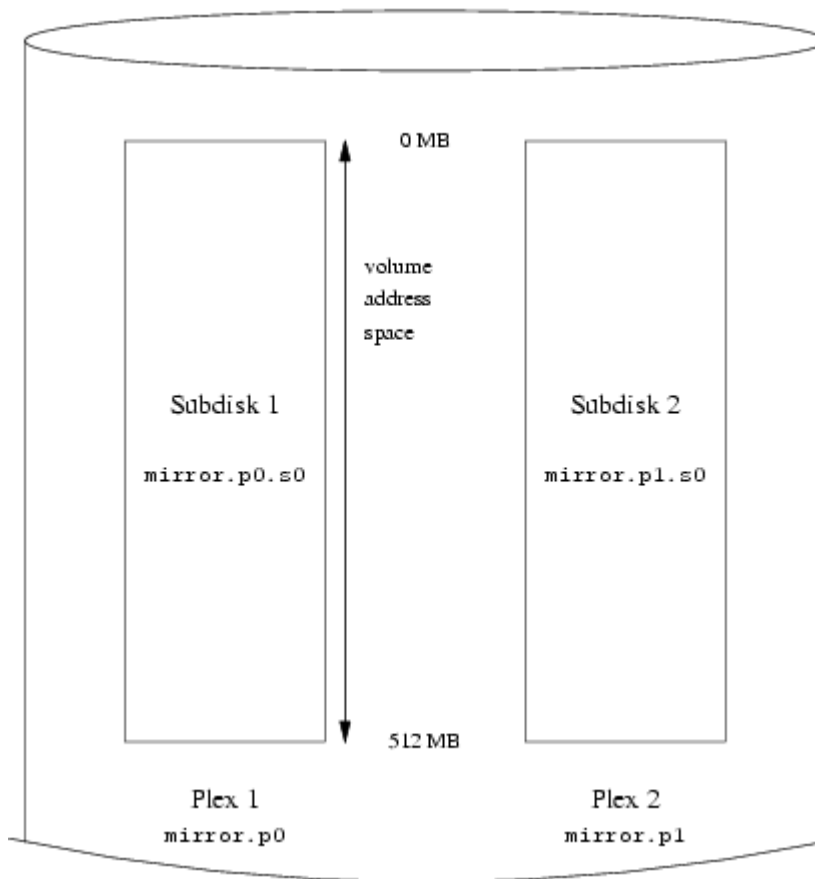
P myvol.p0       C State: up          Subdisks:        1 Size:
512 MB
P mirror.p0      C State: up          Subdisks:        1 Size:
512 MB
P mirror.p1      C State: initializing Subdisks:        1
Size:           512 MB

S myvol.p0.s0    State: up          PO:             0 B Size:
512 MB
S mirror.p0.s0   State: up          PO:             0 B Size:
512 MB
```

```
S mirror.pl.s0          State: empty    PO:          0 B Size:
512 MB
```

[Figura 13-5](#) mostra a estrutura graficamente.

Figura 13-5. Um Volume de Vinum Espelhado



Nesse exemplo, cada plex contém os 512 MB de espaço de endereço. E como no exemplo anterior, cada plex contém um único subdisco.

13.6.3. Otimizando a Performance

O espelhamento do volume no exemplo anterior é mais resistente a falhas do que o volume não espelhado, mas sua performance é menor: cada escrita no volume requer a escrita em dois discos, usando assim mais largura de banda de disco. Quando se quer considerar a performance é indicado uma abordagem diferente: ao invés de espelhar, os dados são segmentados dentro o maior numero de partições possível. A configuração seguinte mostra um volume com um plex *striped* entre quatro partições:

```
drive c device /dev/da5h
drive d device /dev/da6h
volume stripe
plex org striped 512k
sd length 128m drive a
```

```
sd length 128m drive b
sd length 128m drive c
sd length 128m drive d
```

Como antes, não é necessário definir as partições que já são conhecidas pelo vinum.
Depois de processar essa definição, a configuração se parece com:

```
Drives:          4 (4 configured)
Volumes:         3 (4 configured)
Plexes:          4 (8 configured)
Subdisks:        7 (16 configured)

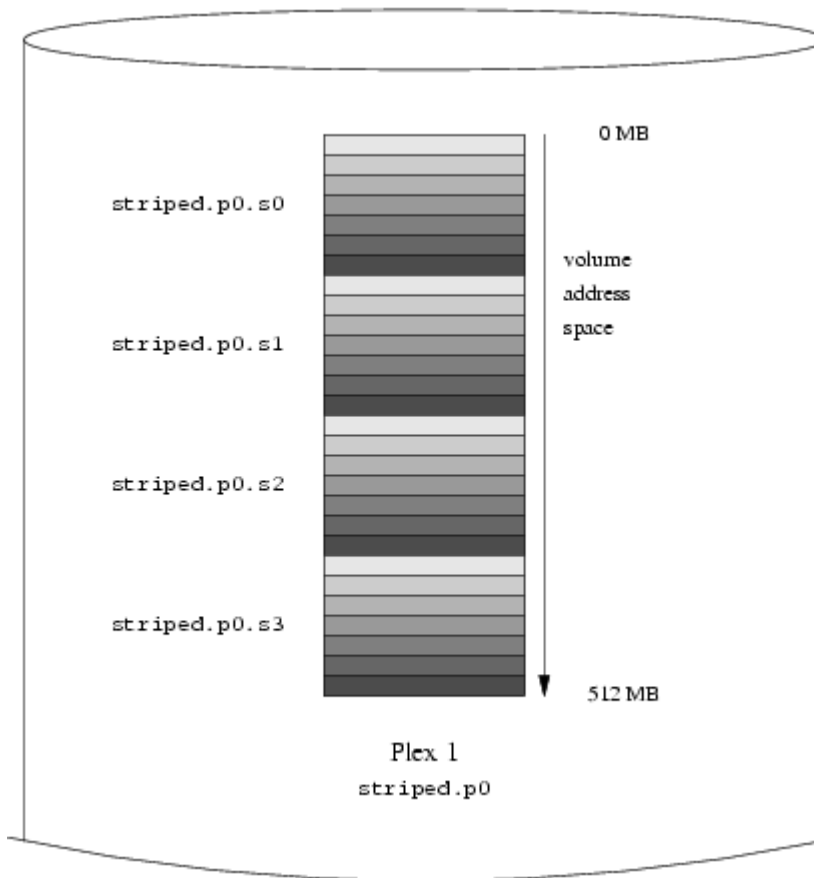
  D a                State: up          Device /dev/da3h
Avail: 1421/2573 MB (55%)
  D b                State: up          Device /dev/da4h
Avail: 1933/2573 MB (75%)
  D c                State: up          Device /dev/da5h
Avail: 2445/2573 MB (95%)
  D d                State: up          Device /dev/da6h
Avail: 2445/2573 MB (95%)

  V myvol            State: up          Plexes:      1 Size:
512 MB
  V mirror           State: up          Plexes:      2 Size:
512 MB
  V striped          State: up          Plexes:      1 Size:
512 MB

  P myvol.p0         C State: up          Subdisks:    1 Size:
512 MB
  P mirror.p0        C State: up          Subdisks:    1 Size:
512 MB
  P mirror.p1        C State: initializing Subdisks:    1
Size:          512 MB
  P striped.p1       State: up          Subdisks:    1 Size:
512 MB

  S myvol.p0.s0      State: up          PO:          0 B Size:
512 MB
  S mirror.p0.s0     State: up          PO:          0 B Size:
512 MB
  S mirror.p1.s0     State: empty       PO:          0 B Size:
512 MB
  S striped.p0.s0    State: up          PO:          0 B Size:
128 MB
  S striped.p0.s1    State: up          PO:          512 kB Size:
128 MB
  S striped.p0.s2    State: up          PO:          1024 kB Size:
128 MB
  S striped.p0.s3    State: up          PO:          1536 kB Size:
128 MB
```

Figura 13-6. Um volume Vinum *Striped*



Esse volume é representado em [Figura 13-6](#). O sombreado dos *stripes* indica a porção junto ao espaço endereçado do plex: os stripes mais claros vem primeiro, os mais escuros por último.

13.6.4. Estabilidade e Performance

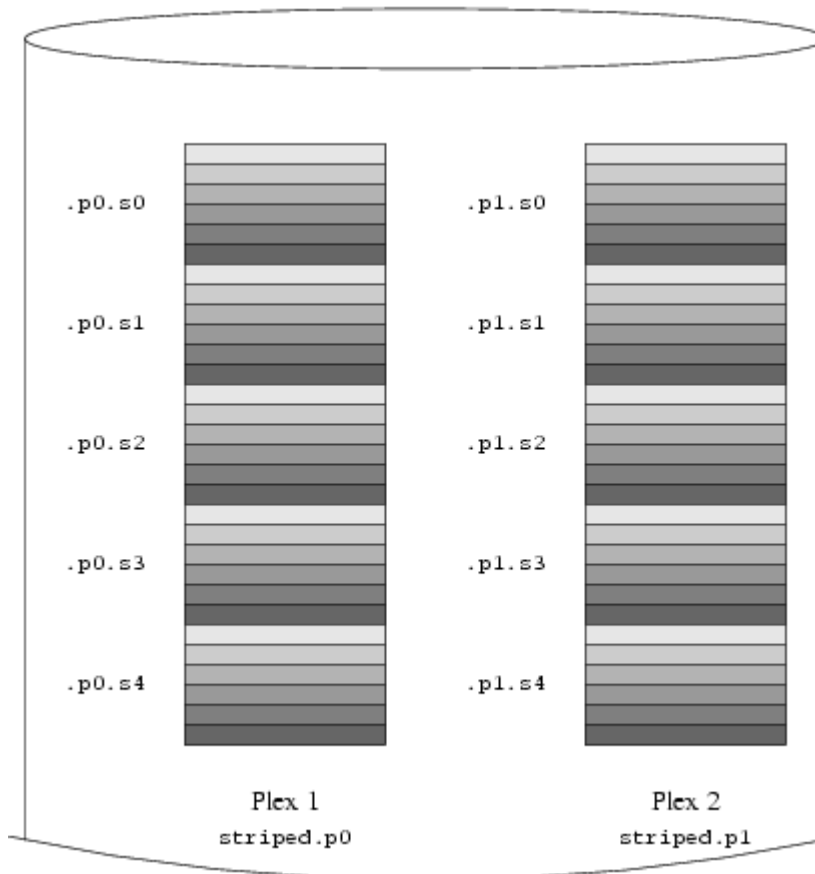
Com o hardware necessário, é possível construir volumes que apresentam tanto acréscimo de performance quanto de estabilidade quando comparados às partições UNIX® comuns. Um arquivo de configuração típico pode ser:

```
volume raid10
  plex org striped 512k
    sd length 102480k drive a
    sd length 102480k drive b
    sd length 102480k drive c
    sd length 102480k drive d
    sd length 102480k drive e
  plex org striped 512k
    sd length 102480k drive c
    sd length 102480k drive d
    sd length 102480k drive e
    sd length 102480k drive a
    sd length 102480k drive b
```

Os subdiscos do segundo plex estão deslocados por dois discos em relação ao primeiro plex: isso ajuda a garantir que não acontecerá escrita no mesmo subdisco, mesmo que a transferência aconteça sobre dois discos.

[Figura 13-7](#) representa a estrutura desse volume.

Figura 13-7. Um Volume Vinum Striped e Espelhado



13.7. Nomenclatura de objetos

Como descrito acima, Vinum designa nomes padrões para plexes e subdiscos, embora você possa sobrescrevê-los. Sobrescrever os nomes padrões, não é recomendado: experiências com o gerenciador de volumes VERITAS, que permite nomes arbitrários para objetos, têm mostrado que essa flexibilidade não trás vantagem significativa e pode causar confusão.

Nomes podem conter qualquer caracter não nulo, mas é recomendado que estes sejam restritos a letras, dígitos e caracteres de sublinhado. Os nomes dos volumes, plexes e subdiscos podem ser formados por até 64 caracteres, e o nome dos drives por até 32.

Objetos do Vinum são dispositivos designados na hierarquia `/dev/vinum`. A configuração descrita acima faria o Vinum a criar os seguintes nós de dispositivo:

- Os dispositivos de controle `/dev/vinum/control` e `/dev/vinum/controld`, os quais são usados por [vinum\(8\)](#) e pelo daemon do Vinum respectivamente.
- Entrada de dispositivo de caracter e bloco para cada volume. Esses são os principais dispositivos usados pelo Vinum. Os nomes dos dispositivos de bloco são o nome do volume, enquanto os de caracter seguem a tradição BSD de preceder a letra *r* ao nome. Por isso a configuração abaixo incluiria os dispositivos de bloco `/dev/vinum/myvol`, `/dev/vinum/mirror`, `/dev/vinum/striped`, `/dev/vinum/raid5` e `/dev/vinum/raid10`, e os dispositivos de caracter `/dev/vinum/rmyvol`, `/dev/vinum/rmirror`, `/dev/vinum/rstriped`, `/dev/vinum/rraid5` e `/dev/vinum/rraid10`. óbviamente existe um problema aqui: é possível ter dois volumes chamados *r* e *rr*, mas ocorreria um conflito criando o nó do dispositivo `/dev/vinum/rr`: seria um dispositivo de caracter para o volume *r* ou um dispositivo bloco para o volume *rr*? Atualmente Vinum não trata este conflito: o primeiro volume definido levará o nome.
- Um diretório `/dev/vinum/drive` com entradas para cada disco. Essas entradas são na verdade *links* simbólicos para os discos nós correspondentes.
- Um diretório `/dev/vinum/volume` com entradas para cada volume, que contém um subdiretório para cada plex, que por sua vez contém subdiretórios para seus subdiscos.
- Os diretórios `/dev/vinum/plex`, `/dev/vinum/sd`, e `/dev/vinum/rsd`, onde contém um nó de dispositivo de bloco para cada plex e um nó de dispositivo de bloco e caracter para cada subdisco, respectivamente.

Por exemplo, considere o seguinte arquivo de configuração:

```
drive drive1 device /dev/sd1h
drive drive2 device /dev/sd2h
drive drive3 device /dev/sd3h
drive drive4 device /dev/sd4h
volume s64 setupstate
  plex org striped 64k
    sd length 100m drive drive1
    sd length 100m drive drive2
    sd length 100m drive drive3
    sd length 100m drive drive4
```

Depois de processar esse arquivo, [vinum\(8\)](#) cria a seguinte estrutura em `/dev/vinum`:

```
brwx----- 1 root wheel 25, 0x40000001 Apr 13 16:46 Control
brwx----- 1 root wheel 25, 0x40000002 Apr 13 16:46 control
brwx----- 1 root wheel 25, 0x40000000 Apr 13 16:46 controld
drwxr-xr-x 2 root wheel 512 Apr 13 16:46 drive
drwxr-xr-x 2 root wheel 512 Apr 13 16:46 plex
crwxr-xr-- 1 root wheel 91, 2 Apr 13 16:46 rs64
drwxr-xr-x 2 root wheel 512 Apr 13 16:46 rsd
drwxr-xr-x 2 root wheel 512 Apr 13 16:46 rvol
brwxr-xr-- 1 root wheel 25, 2 Apr 13 16:46 s64
drwxr-xr-x 2 root wheel 512 Apr 13 16:46 sd
drwxr-xr-x 3 root wheel 512 Apr 13 16:46 vol
```

```

/dev/vinum/drive:
total 0
lrwxr-xr-x 1 root wheel 9 Apr 13 16:46 drive1 -> /dev/sd1h
lrwxr-xr-x 1 root wheel 9 Apr 13 16:46 drive2 -> /dev/sd2h
lrwxr-xr-x 1 root wheel 9 Apr 13 16:46 drive3 -> /dev/sd3h
lrwxr-xr-x 1 root wheel 9 Apr 13 16:46 drive4 -> /dev/sd4h

/dev/vinum/plex:
total 0
brwxr-xr-- 1 root wheel 25, 0x10000002 Apr 13 16:46 s64.p0

/dev/vinum/rsd:
total 0
crwxr-xr-- 1 root wheel 91, 0x20000002 Apr 13 16:46 s64.p0.s0
crwxr-xr-- 1 root wheel 91, 0x20100002 Apr 13 16:46 s64.p0.s1
crwxr-xr-- 1 root wheel 91, 0x20200002 Apr 13 16:46 s64.p0.s2
crwxr-xr-- 1 root wheel 91, 0x20300002 Apr 13 16:46 s64.p0.s3

/dev/vinum/rvol:
total 0
crwxr-xr-- 1 root wheel 91, 2 Apr 13 16:46 s64

/dev/vinum/sd:
total 0
brwxr-xr-- 1 root wheel 25, 0x20000002 Apr 13 16:46 s64.p0.s0
brwxr-xr-- 1 root wheel 25, 0x20100002 Apr 13 16:46 s64.p0.s1
brwxr-xr-- 1 root wheel 25, 0x20200002 Apr 13 16:46 s64.p0.s2
brwxr-xr-- 1 root wheel 25, 0x20300002 Apr 13 16:46 s64.p0.s3

/dev/vinum/vol:
total 1
brwxr-xr-- 1 root wheel 25, 2 Apr 13 16:46 s64
drwxr-xr-x 3 root wheel 512 Apr 13 16:46 s64.plex

/dev/vinum/vol/s64.plex:
total 1
brwxr-xr-- 1 root wheel 25, 0x10000002 Apr 13 16:46 s64.p0
drwxr-xr-x 2 root wheel 512 Apr 13 16:46 s64.p0.sd

/dev/vinum/vol/s64.plex/s64.p0.sd:
total 0
brwxr-xr-- 1 root wheel 25, 0x20000002 Apr 13 16:46 s64.p0.s0
brwxr-xr-- 1 root wheel 25, 0x20100002 Apr 13 16:46 s64.p0.s1
brwxr-xr-- 1 root wheel 25, 0x20200002 Apr 13 16:46 s64.p0.s2
brwxr-xr-- 1 root wheel 25, 0x20300002 Apr 13 16:46 s64.p0.s3

```

Apesar de não ser recomendado alocação de nomes específicos para plexes e subdiscos, os discos Vinum devem ser nomeados. Isso torna possível mover um disco para uma outra localidade e ainda sim reconhecê-lo automaticamente. Os nomes dos discos podem conter até 32 caracteres.

13.7.1. Criando Sistemas de Arquivos

Volumes e discos parecem idênticos para o sistema, com uma exceção. Ao contrário de discos UNIX®, Vinum não particiona volumes, assim, não possui uma tabela de partições. Isso exigiu uma mudança em alguns utilitários de discos como [newfs\(8\)](#), que previamente tentava interpretar a última letra de um nome de volume Vinum como um

identificador de partições. Por exemplo, uma partição de disco pode ter um nome como `/dev/ad0a` ou `/dev/da2h`. Esses nomes representam a primeira partição (a) no primeiro (0) disco IDE (ad) e a oitava partição (h) no terceiro disco SCSI (2) (da), respectivamente. Em contraste, um volume Vinum pode ser chamado de `/dev/vinum/concat`, um nome que não tem nenhum relacionamento com o nome da partição.

Normalmente, [newfs\(8\)](#) interpreta o nome do disco e reclama se não entendê-lo. Por exemplo:

```
# newfs /dev/vinum/concat
newfs: /dev/vinum/concat: can't figure out file system partition
```

Nota: O que segue só é válido para as versões de FreeBSD anteriores a 5.0:

Para criar um sistema de arquivos nesse volume, use a opção `-v` para [newfs\(8\)](#):

```
# newfs -v /dev/vinum/concat
```

13.8. Configurando Vinum

O *kernel* GENERIC não contém Vinum. É possível construir um *kernel* especial para incluir o Vinum, mas não é recomendado. A maneira simples de iniciar o Vinum é com o módulo de *kernel* (kld). Você nem precisa usar [kldload\(8\)](#) para Vinum: quando você inicia [vinum\(8\)](#), ele verifica se o módulo foi carregado ou não, e o carrega automaticamente.

13.8.1. Inicialização

Vinum armazena as informações de configuração das partições de disco essencialmente da mesma forma que os arquivos de configuração. Quando lê do banco de dados de configuração, Vinum reconhece algumas palavras chave que não são permitidas nos arquivos de configuração. Por exemplo, uma configuração de disco talvez contenha o seguinte texto:

```
volume myvol state up
volume bigraid state down
plex name myvol.p0 state up org concat vol myvol
plex name myvol.p1 state up org concat vol myvol
plex name myvol.p2 state init org striped 512b vol myvol
plex name bigraid.p0 state initializing org raid5 512b vol bigraid
sd name myvol.p0.s0 drive a plex myvol.p0 state up len 1048576b
driveoffset 265b plexoffset 0b
sd name myvol.p0.s1 drive b plex myvol.p0 state up len 1048576b
driveoffset 265b plexoffset 1048576b
sd name myvol.p1.s0 drive c plex myvol.p1 state up len 1048576b
driveoffset 265b plexoffset 0b
sd name myvol.p1.s1 drive d plex myvol.p1 state up len 1048576b
driveoffset 265b plexoffset 1048576b
sd name myvol.p2.s0 drive a plex myvol.p2 state init len 524288b
driveoffset 1048841b plexoffset 0b
```

```
sd name myvol.p2.s1 drive b plex myvol.p2 state init len 524288b
driveoffset 1048841b plexoffset 524288b
sd name myvol.p2.s2 drive c plex myvol.p2 state init len 524288b
driveoffset 1048841b plexoffset 1048576b
sd name myvol.p2.s3 drive d plex myvol.p2 state init len 524288b
driveoffset 1048841b plexoffset 1572864b
sd name bigraid.p0.s0 drive a plex bigraid.p0 state initializing len
4194304b driveoff set 1573129b plexoffset 0b
sd name bigraid.p0.s1 drive b plex bigraid.p0 state initializing len
4194304b driveoff set 1573129b plexoffset 4194304b
sd name bigraid.p0.s2 drive c plex bigraid.p0 state initializing len
4194304b driveoff set 1573129b plexoffset 8388608b
sd name bigraid.p0.s3 drive d plex bigraid.p0 state initializing len
4194304b driveoff set 1573129b plexoffset 12582912b
sd name bigraid.p0.s4 drive e plex bigraid.p0 state initializing len
4194304b driveoff set 1573129b plexoffset 16777216b
```

A diferença clara aqui é a presença explícita de informações sobre localidade e nomenclatura (Sendo as duas no caso permitidas, mas desencorajadas) e também informações sobre os estados (mas não são disponíveis para o usuário). Vinum não armazena informações sobre os discos nas informações de configuração: ele acha as informações pesquisando nos discos configurados por partições identificadas como Vinum. Isso habilita o Vinum a identificar discos corretamente, mesmo que eles tenham sido rotulados com uma identificação de drives UNIX® diferente.

13.8.1.1. Inicialização Automática

Para iniciar o Vinum automaticamente quando você inicializa o sistema, certifique-se de que você tem a seguinte linha no seu `/etc/rc.conf`:

```
start_vinum="YES"          # set to YES to start vinum
```

Se o seu sistema não possuir o arquivo `/etc/rc.conf`, crie um com esse conteúdo. Isso fará com que o sistema carregue o Vinum kld na inicialização, e inicie qualquer objeto citado na configuração. Isso é feito antes de montar os sistemas de arquivos, então, é possível fazer automaticamente um [fsck\(8\)](#) e montar os sistemas de arquivo nos volumes Vinum.

Quando você inicia Vinum com o comando `vinum start`, ele lê a base de dados de configuração de um dos discos Vinum. Sobre circunstâncias normais, cada um dos discos contém uma cópia idêntica da base de dados de configuração, então não importa que disco é lido. Embora, após uma pane, Vinum tenha que determinar qual o disco foi atualizado mais recentemente e ler as configurações desse disco. E então as atualiza, se necessário, progressivamente no disco mais velho.

13.9. Usando Vinum para o Sistema de Arquivos Raiz

Para uma máquina que esta espelhada por inteiro usando Vinum, é desejável espelhar também o sistema de arquivos raiz. Executar tal configuração é menos trivial do que espelhar um disco arbitrário:

- O sistema de arquivos raiz deve estar disponível bem cedo durante o processo de inicialização, então a infra-estrutura Vinum deveria estar disponível também nessa hora.
- O volume contendo os arquivos de sistema raiz também contém o bootstrap de sistema e o *kernel*, no qual tem que ser lido pela utilidade nativa de sistema (ex. A BIOS no caso dos PC's) onde frequentemente não pode ser ensinada sobre os detalhes do Vinum.

Nas sessões que seguem, o termo ``root volume" é usado para descrever o volume Vinum que contém os sistemas de arquivo raiz. É provavelmente uma boa idéia usar o nome "*root*" para esse volume, mas isso não é de maneira nenhuma obrigatório. Apesar de que todos os exemplos de comandos nas sessões seguintes assumem esse nome.

13.9.1. Iniciando o Vinum a Tempo Para o Sistema de Arquivos Raiz

Existem algumas medidas a se tomar para que isso aconteça:

- Vinum precisa estar disponível no *kernel* na hora do boot. Então, o método para iniciar automaticamente o Vinum descrito em [Seção 13.8.1.1](#) não é aplicável nesse caso, e o parâmetro *start_vinum* não pode estar ativado quando a seguinte configuração estiver sendo arrumada. A primeira opção seria compilar o Vinum staticamente no *kernel*, então estará disponível o tempo todo, mas isso também não é sempre desejável. Existe outra opção no entanto, que seria fazer com que o */boot/loader* ([Seção 7.3.3](#)) carregue o módulo de kernel do vinum cedo, antes de iniciar o *kernel*. Isso pode ser feito adicionando a seguinte linha

```
vinum_load="YES"
```

no arquivo */boot/loader.conf*.

- Vinum deve ser iniciado cedo pois ele precisa suprir o volume para os arquivos de sistemas raiz. Por padrão, a parte do *kernel* do vinum não esta procurando por discos que possam conter informações de volume Vinum até que o administrador (ou um dos scripts de inicialização) lance o comando `vinum start`.

Nota: Os parágrafos seguintes descrevem os passos necessários para o FreeBSD 5.x e superiores. Os passos obrigatórios para o 4.x, são diferentes e estão descritos a baixo em [Seção 13.9.5](#).

Colocando a linha:

```
vinum.autostart="YES"
```

em `/boot/loader.conf`, Vinum é instruído para procurar automaticamente em todos os discos por informações úteis a ele como parte da inicialização do *kernel*.

Note que não é necessário ensinar ao *kernel* aonde procurar pelos arquivos de sistema raiz. `/boot/loader` procura o nome do dispositivo raiz em `/etc/fstab`, e passa essa informação para o *kernel*. Quando chega a hora de montar os arquivos de sistema raiz, o *kernel* descobre a partir do nome do dispositivo fornecido qual é o disco que se deve pedir para passar o nome do dispositivo raiz para um ID interno de dispositivo (número maior/menor).

13.9.2. Tornando um Volume Raiz Baseado em Vinum Acessível Para o Bootstrap

Visto que o atual *Bootstrap* do FreeBSD contém apenas 7.5 KB de código, e já tem o trabalho de ler arquivos no UFS (como `/boot/loader`), é completamente impossível ensiná-lo também sobre as estruturas internas do Vinum para que ele possa analisar seus dados de configuração, e descobrir sozinho os elementos de um volume de inicialização. Dessa forma, alguns truques são necessários para fornecer ao código do *bootstrap* a ilusão de que ele é uma partição "a" padrão que contém o arquivo de sistema raiz.

Para que isso se torne possível, os seguintes requisitos para o volume raiz devem ser cumpridos:

- O volume raiz não pode estar partido ou em RAID-5.
- Um volume raiz não pode conter mais de um subdisco concatenado por plex.

Note que é desejado e possível que existam múltiplos plexes, cada um contendo uma réplica do sistema de arquivos raiz. O processo de *bootstrap* irá, entretanto, usar somente uma delas para achar o *bootstrap* e todos os arquivos, até que o próprio *kernel* monte os sistemas de arquivos. Cada subdisco dentro desse plex irá então precisar da sua própria partição imaginária "a", para que o dispositivo correspondente se torne bootável. Não é estritamente necessário que cada uma dessas falsas partições "a" se encontrem no mesmo ponto junto ao seu dispositivo, comparado com outros plexes do volume raiz. Entretanto, provavelmente seria uma boa idéia criar os volumes do Vinum desse jeito para que os dispositivos espelhados sejam simétricos, evitando, assim, confusão.

Para definir essas partições "a", para cada dispositivo contendo parte do volume raiz, as etapas seguintes precisam ser realizadas:

1. A localização (*offset* do início do dispositivo) e tamanho dos subdiscos dos dispositivos que fazem parte do volume raiz precisam ser examinados usando o comando

```
vinum l -rv root
```

Note que no Vinum, os *offsets* e tamanhos são medidos em bytes. Eles precisam ser divididos por 512 para obter o número de blocos que irão, por sua vez, ser usados no comando `disklabel`.

2. Execute o comando

```
disklabel -e devname
```

Para cada dispositivo que participa no volume raiz. *devname* deve ser ou o nome do disco (exemplo `da0`) para discos sem tabela de partições (ex. `fdisk`), ou o nome da partição (exemplo `ad0s1`).

Se já existir uma partição "*a*" no dispositivo (provavelmente, contendo um sistema de arquivos raiz pré-Vinum), essa deve ser renomeada para um nome qualquer, permanecendo acessível (por segurança), mas não será mais usado para *bootstrap* como padrão. Note que partições ativas (como um sistema de arquivo raiz atualmente montado) não podem ser renomeadas, então isso deve ser feito quando a máquina é inicializada através de uma mídia ``Fixit'', ou em um processo de duas etapas, onde (em uma situação de espelhamento) o disco que não foi usado no boot será manipulado primeiro.

Então, o *offset* da partição Vinum nesse dispositivo (se existente) deve ser adicionado ao *offset* do subdisco de volume raiz respectivo desse mesmo dispositivo. O valor resultante passará a ser o valor "*offset*" para a nova partição "*a*". O valor do "*size*" dessa partição pode ser tirado literalmente do cálculo acima. O "*fstype*" deve ser *4.2BSD*. Os valores de "*fsize*", "*bsize*", e "*cpg*" devem ser escolhidos para melhor se adaptar aos atuais do sistema, apesar de que, eles não são muito importantes dentro desse contexto.

Dessa forma, uma nova partição "*a*" será formada e sobrepõe as partições Vinum nesse dispositivo. Note que o `disklabel` só vai permitir isso se as partições do vinum tiverem sido propriamente marcadas usando o "*vinum*" *fstype*.

3. Isso é tudo! Uma partição falsa "*a*" agora existe em cada dispositivo que tenha uma réplica do volume raiz. É altamente recomendável verificar o resultado novamente, usando um comando como

```
fsck -n /dev/devnamea
```

Deve ser lembrado que todos os arquivos que contém informações de controle, precisam ser relativos ao sistema de arquivos raiz no volume Vinum onde, quando definido um novo volume raiz Vinum, não deve coincidir com o sistema raiz atualmente ativo. Então, em particular, os arquivos `/etc/fstab` e `/boot/loader.conf` precisarão ser observados.

Na próxima inicialização, o *bootstrap* deve descobrir a informação de controle apropriada a partir do novo sistema de arquivos raiz Vinum, e agir adequadamente. No final do processo de inicialização do *kernel*, depois que todos os dispositivos forem

anunciados, a notícia importante que mostra o sucesso dessa configuração é uma mensagem como:

```
Mounting root from ufs:/dev/vinum/root
```

13.9.3. Exemplo Configuração Vinum Para o Sistemas de Arquivos Raiz

Após o volume raiz do Vinum ser ativado, a saída de `vinum l -rv root` deve parecer com:

```
...
Subdisk root.p0.s0:
  Size:      125829120 bytes (120 MB)
  State: up
  Plex root.p0 at offset 0 (0 B)
  Drive disk0 (/dev/da0h) at offset 135680 (132 kB)

Subdisk root.p1.s0:
  Size:      125829120 bytes (120 MB)
  State: up
  Plex root.p1 at offset 0 (0 B)
  Drive disk1 (/dev/dalh) at offset 135680 (132 kB)
```

Os valores a serem notados são *135680* para o *offset* (relativo à partição `/dev/da0h`). Isso quer dizer, em termos de `disklabel`, 265 blocos de disco com 512 bytes cada. Da mesma forma, o tamanho desse volume raiz é de 245760 blocos de 512 bytes. `/dev/dalh`, contendo a segunda réplica desse volume raiz, tem uma configuração simétrica.

O *disklabel* para esses dispositivos devem se parecer com:

```
...
8 partitions:
#      size  offset  fstype  [fsize bsize bps/cpg]
#  a:  245760    281   4.2BSD   2048 16384    0  # (Cyl.  0*-
15*)
#  c:  71771688    0  unused      0    0    # (Cyl.  0 -
4467*)
#  h:  71771672    16   vinum      # (Cyl.  0*-
4467*)
```

Pode ser observado que o parâmetro *"size"* para a partição *"a"* falsa bate com o valor esboçado acima, enquanto o parâmetro *"offset"* é a soma do *offset* dentro da partição Vinum *"h"*, e o *offset* dessa partição dentro do dispositivo (ou *slice*). Essa é a configuração típica necessária para evitar os problemas descritos em [Seção 13.9.4.3](#). Pode também ser visto que toda partição *"a"* está completamente dentro da partição *"h"* contendo todos os dados do Vinum para esse dispositivo.

Note que no exemplo acima, o dispositivo inteiro está dedicado ao Vinum, e não há nenhum resto de partição raiz pré-Vinum, partindo de que esse foi um disco com configurações novas e foi dedicado exclusivamente a ser parte de uma configuração Vinum.

13.9.4. Resolução de Problemas

Caso algo saia errado, é preciso ter uma saída para recuperar a situação. A seguinte lista contém algumas armadilhas e suas soluções.

13.9.4.1. O *Bootstrap* do sistema entra, mas o sistema não inicializa

Caso, por alguma razão, o sistema não continue inicializando, o *bootstrap* pode ser interrompido pressionando-se a tecla **espaço** no alerta de 10 segundos. As variáveis do *loader* (como *vinum.autostart*) podem ser examinadas usando o `show`, e manipuladas com os comandos `set` ou `unset`.

Caso o único problema seja que o módulo Vinum do *kernel* não esteja na lista de módulos para serem carregados automaticamente, um simples `load vinum` irá ajudar.

Quando pronto, o processo de *boot* pode ser retomado com o comando `boot -as`. As opções *-as* irão requerer ao *kernel* que consulte qual sistema de arquivos raiz deve montar (*-a*), e faça o processo em modo *single-user* (*-s*), onde o sistema raiz é montado apenas para leitura. Desta forma, mesmo que apenas um plex de um volume multi-plex tenha sido montado, nenhum dado entre os plexes corre risco de inconsistência.

Quando for solicitado o sistema de arquivos raiz a ser montado, qualquer dispositivo que contenha um sistema de arquivos raiz válido pode ser informado. Caso `/etc/fstab` tenha sido definido corretamente, o esperado é algo como `ufs:/dev/vinum/root`. Uma escolha alternativa típica pode ser algo como `ufs:da0d` que pode ser uma hipotética partição contendo o sistema de arquivos pré-Vinum. Deve-se tomar cuidado caso uma das partições apelido "a" forem informadas onde há na verdade uma referência aos subdiscos Vinum para o sistema de arquivos raiz, porque em uma configuração de espelhamento, deve-se apenas montar uma peça de um dispositivo raiz. Caso o sistema de arquivos for montado para leitura e escrita posteriormente, será necessário remover o(s) outro(s) plex(es) do volume Vinum raiz, uma vez que estes plexes provavelmente trarão dados inconsistentes.

13.9.4.2. Apenas o *Bootstrap* Primário Inicia

Caso o `/boot/loader` não seja carregado, mas o *bootstrap* primário continue carregando (vível por uma barra a esquerda da tela imediatamente após o processo de *boot* iniciar), uma tentativa de interromper o *bootstrap* primário é possível, usando a tecla **espaço**. Isto vai o *bootstrap* parar no estágio dois, ver [Seção 7.3.2](#). Neste momento é possível realizar o *boot* de uma partição alternativa, como a partição contendo o sistema de arquivos raiz antigo que foi removido de "a" acima.

13.9.4.3. Nada inicializa, o *Bootstrap* trava

Esta situação vai ocorrer quando o *bootstrap* foi destruído pela instalação do Vinum. Infelizmente, atualmente o Vinum deixa livres apenas 4 KB no início de sua partição antes de iniciar a escrita do seu cabeçalho de informações. Contudo, os estágios um e dois do *bootstrap* mais o *disklabel* incorporado entre eles atualmente requerem 8 KB. Assim, se a partição Vinum for iniciada no *offset* 0 em um *slice* ou disco que pretende ser alvo de *boot*, a configuração do Vinum vai corromper o *bootstrap*.

Similarmente, caso a situação acima tenha sido sanada, por exemplo, reiniciando a partir de uma mídia ``Fixit'', e o *bootstrap* tenha sido reinstalado usando o comando `disklabel -B` como descrito em [Seção 7.3.2](#), o *bootstrap* irá corromper o cabeçalho Vinum, e assim, não encontrará mais seu(s) disco(s). Contudo nenhuma informação de configuração ou dados reais nos volumes Vinum serão perdidos, e é possível recuperar todos os dados informando-se exatamente a mesma configuração novamente, a situação é complexa mesmo. Pode ser necessário afastar toda a partição Vinum pelo menos 4 KB, de forma a ter o cabeçalho Vinum e o *bootstrap* do sistema sem colisões.

13.9.5. Diferenças para o FreeBSD 4.x

Com o FreeBSD 4.x, algumas funções internas são obrigatórias para fazer com que o Vinum procure por todos os discos, e o código que descobre o identificador interno do dispositivo raiz não é inteligente o suficiente para tratar algo como `/dev/vinum/root` automaticamente. Então, as coisas são um pouco diferentes aqui.

Vinum precisa ser explicitamente informado dos discos que ele tem que procurar, use uma linha parecida com a seguinte em `/boot/loader.conf`:

```
vinum.drives="/dev/da0 /dev/da1"
```

É importante que todos os discos que contém dados do Vinum sejam mencionados. Não será prejudicial se discos a *mais* forem listados, e nem é necessário adicionar cada *slice* ou cada partição explicitamente, porque o Vinum irá procurar por cabeçalhos válidos em todas as partições dos discos citados.

A partir do momento em que as rotinas usadas para passar o nome do sistema de arquivos raiz e derivar o ID do dispositivo (número maior/menor) estão preparadas apenas para suportar nomes de dispositivos ``clássicos" como `/dev/ad0s1a`, eles não tem a menor idéia do que fazer com um nome de volume raiz do tipo `/dev/vinum/root`. Por esta razão, o próprio Vinum precisa definir antecipadamente o parâmetro interno de *kernel* que armazena o número de identidade do dispositivo raiz durante a sua inicialização. Isso é feito definindo a variável `vinum.root` com o valor do nome do volume raiz. A entrada em `/boot/loader.conf` para efetuar essa tarefa se parece com:

```
vinum.root="root"
```

Agora, quando a inicialização do *kernel* tenta encontrar o dispositivo raiz a ser montado, vê se algum módulo do *kernel* já pré-iniciou o parâmetro para isso. Caso seja esta a situação, e o dispositivo requisitando o sistema raiz coincida com o número maior do driver, como indicado pelo nome do dispositivo raiz sendo passado (isto é, "vinum" no nosso caso), ela vai usar o ID do dispositivo pré-alocado, ao contrário de tentar eleger

um por si mesma. Desta maneira, no decorrer da inicialização automática, pode continuar a montar o volume Vinum raiz como sistema de arquivos raiz.

Contudo, quando `boot -a` requisitar a entrada do nome do dispositivo raiz manualmente, deve-se considerar que esta rotina permanece não tratando o valor informado caso refira-se a um volume Vinum. Caso algum nome de dispositivo não referente aos dispositivos Vinum seja informado, a distinção entre os números maiores do parâmetro raiz pré-alocado e o driver como descrito no nome informado vai fazer esta rotina entrar em seu interpretador normal, assim entrando uma seqüência como: `ufs:da0d` irá funcionar como o esperado. Observe que se isto falhar, não será mais possível informar um novo valor como `ufs:vinum/root`, e iniciar novamente, uma vez que não pode mais ser tratado pela rotina. Ao informar dados no ``askroot'', o `/dev/` inicial sempre poderá ser omitido.)

Capítulo 14. Localização - I18N/L10N

Uso e Configuração

Índice

- 14.1. [Sinótese](#)
- 14.2. [O Básico](#)
- 14.3. [Usando a localização](#)
- 14.4. [Compilando programas I18N](#)
- 14.5. [Localizando o FreeBSD para línguas específicas](#)

Contribuído por Andrey A. Chernov. Reescrito por Michael C. Wu.

14.1. Sinótese

O FreeBSD é um projeto bastante distribuído com usuários e contribuidores localizados por todo o mundo. Este capítulo discute características da internacionalização e localização do FreeBSD que permitem usuários que não falam inglês conheçam o real trabalho feito. Há muitos aspectos da implementação do i18n nos níveis de aplicação e sistema, então onde aplicável nos recomendamos o leitor as fontes de documentação mais específica.

Depois de ler esse capítulo, você irá saber:

- Como linguagens diferentes e locais são codificados em sistemas operacionais modernos.
- Como configurar o locale para a shell do seu login.
- Como configurar seu console para linguagens que não seja o inglês.
- Como usar efetivamente X Windows com uma linguagem diferente.
- Onde encontrar mais informações sobre escrita de aplicações i18n-compliant applications.

Antes de ler este capítulo, você deve:

- Saber como instalar aplicativos adicionais de terceiros ([Capítulo 4](#)).
-

14.2. O Básico

14.2.1. O que é I18N/L10N?

Desenvolvedores encurtaram a palavra "internationalization" no termo I18N, contando o número de letras entre a primeira e a última letra de "internationalization". L10N usa o mesmo esquema, vindo de "localization". Combinados juntos, os métodos I18N/L10N, protocolos, e aplicações permitem os usuários usar linguagens de sua escolha.

As aplicações I18N são programadas usando kits I18N sob bibliotecas. Isto permite aos desenvolvedores escrever um arquivo simples e traduzir menus e textos expostos para cada linguagem. Nós incentivamos fortemente programadores a seguir essa convenção.

14.2.2. Por que eu devo usar I18N/L10N?

I18N/L10N é usado sempre quando você quiser visualizar a entrada, ou dados do processo em línguas não inglesas.

14.2.3. Quais linguagens são suportadas no esforço do I18N?

O I18N e o L10N não são específicos do FreeBSD. Atualmente, pode ser escolhido uma das principais línguas do mundo, incluindo mas não limitada a: Chinês, Alemão, Japonês, Coreano, Francês, Russo, Vietnamita, Português do Brasil e outras.

14.3. Usando a localização

Em todo seu esplendor, I18N não é específico ao FreeBSD e é uma convenção. Nós incentivamos você a ajudar a seguir esta convenção.

Configurações de localização são baseadas em três termos principais: Código de língua, Código de país, e Codificação. Nomes de locale são construídos a partir dessas partes que seguem:

LanguageCode_CountryCode.Encoding

14.3.1. Códigos de língua e país

A fim do sistema FreeBSD localizar línguas específicas (ou qualquer outra suportada no I18N em sistemas UNIX® like), o usuário precisa procurar pelos códigos para país e língua específicos (códigos de país fala para a aplicação que variação de língua deve ser usada). E também, navegadores web, servidores SMTP/POP, servidores web, etc. Faça decisões baseada nelas. Os seguintes exemplos são de códigos de língua/país:

Código de Língua/País	Descrição
en_US	Inglês - Estados Unidos
ru_RU	Russo para Rússia
zh_TW	Tradicional Chinês para Taiwan

14.3.2. Codificações

Algumas línguas usam codificações não-ASCII que são 8-bit, caracteres wide ou multibyte, veja [multibyte\(3\)](#) para mais detalhes. Aplicações mais velhas não os reconhecem e os confundem por caracteres de controle. Aplicações mais novas geralmente reconhecem caracteres 8-bit. Dependendo da implementação, usuários podem necessitar compilar a aplicação com suporte a caracteres wide ou multibyte, ou configura-lo corretamente. Para poder entrar e processar caracteres wide ou multibyte, a [Coleção de Ports do FreeBSD](#) fornece cada língua com programas diferentes. Consulte a documentação do I18N no respectivo port do FreeBSD.

Especialmente, o usuário deve olhar a documentação da aplicação para decidir como configura-la corretamente ou para passar valores corretos ao configure/Makefile/compilador.

Algumas coisas que você deve ter sempre em mente são:

- Conjunto de caracteres únicos da linguagem C (veja [multibyte\(3\)](#)), ex., ISO-8859-1, ISO-8859-15, KOI8-R, CP437.
- Codificações Wide ou multibyte, ex. EUC, Big5.

Você pode verificar a lista ativa de caracteres configurados em [IANA Registry](#).

Nota: As versões do FreeBSD 4.5 e acima usam preferivelmente codificações do locale X11-compatível.

14.3.3. Aplicações I18N

No sistema de Ports e Pacotes do FreeBSD, aplicações I18N são nomeadas com *I18N* em seus nomes para fácil identificação. Entretanto, elas nem sempre suportam a língua necessitada.

14.3.4. Configurando Locale

Geralmente é suficiente exportar o valor do nome do locale como LANG no login da shell. Isto deve ser feito no arquivo `~/.login_conf` dos usuários ou no arquivo de inicialização dos usuários da shell em (`~/.profile`, `~/.bashrc`, `~/.cshrc`). Não é necessário configurar subconjuntos do locale subsets como o LC_CTYPE, LC_CTIME. Por favor consulte a documentação do FreeBSD sobre línguas-específicas para mais informações.

Você deve configurar as duas seguintes variáveis de ambiente em seus arquivos de configuração:

- LANG para funções da família POSIX® [setlocale\(3\)](#)
- MM_CHARSET para configurar caracteres MIME em aplicações.

Isto inclui a configuração shell do usuário, a configuração específica de aplicação, e configuração do X11.

14.3.4.1. Configurando métodos de Locale

Há dois métodos para configurar o locale, e ambos estão descritos abaixo. O primeiro (recomendado) é atribuindo as variáveis de ambiente em [classe de login](#), e o segundo é adicionando as atribuições da variável de ambiente ao [arquivo de inicialização](#) do sistema shell.

14.3.4.1.1. Método de classes de login

Este método permite as variáveis de ambientes necessárias para o nome do locale e o caracter do MIME ser atribuído uma vez para cada shell em vez de adicionar atribuições específicas da shell para cada arquivo de inicialização. [Configuração a nível de usuário](#) pode ser feita pelo próprio usuário e [configuração a nível de administrador](#) requer privilégios de super usuário.

14.3.4.1.1.1. Configuração a nível de usuário

Aqui está um pequeno exemplo do arquivo `.login_conf` no diretório home do usuário que tem ambas as variáveis de codificação para Latin-1:

```
me:\
:charset=ISO-8859-1:\
:lang=de_DE.ISO8859-1:
```

Aqui está um exemplo do `.login_conf` com variáveis configuradas para o chinês tradicional em codificação BIG-5. Observe que há muito mais variáveis configuradas porque alguns programas não respeitam as variáveis do locale corretamente para Chinês, Japonês, e Coreano.

```
#Usuários que não desejam usar unidades monetárias ou formato de hora
#de Taiwan podem manualmente mudar cada variável
me:\
lang=zh_TW.Big5:\
```

```
lc_all=zh_TW.Big5:\
lc_collate=zh_TW.Big5:\
lc_ctype=zh_TW.Big5:\
lc_messages=zh_TW.Big5:\
lc_monetary=zh_TW.Big5:\
lc_numeric=zh_TW.Big5:\
lc_time=zh_TW.Big5:\
charset=big5:\
xmodifiers="@im=xcin": #Setting the XIM Input Server
```

Veja [Configuração a nível de administrador](#) e [login.conf\(5\)](#) para mais detalhes.

14.3.4.1.1.2. Configuração a nível de administrador

Verifique que a classe do login do usuário em `/etc/login.conf` configura a língua correta. Certifique-se que estas configurações aparecem no `/etc/login.conf`:

```
nome__da_língua:título_da_conta:\
:charset=MIME_charset:\
:lang=nome_do_locale:\
:tc=default:
```

Então baseando em nosso exemplo anterior usando Latin-1, isto deveria ficar assim:

```
german:Conta de usuários alemães:\
:charset=ISO-8859-1:\
:lang=de_DE.ISO8859-1:\
:tc=default:
```

Modificando classes de login com o [vipw\(8\)](#)

Use o `vipw` para adicionar novos usuários, e veja que foi criado uma entrada como esta:

```
usuário:senha:1111:11:língua:0:0:Nome do usuário:/home/usuário:/bin/sh
```

Modificando classes de login com o [adduser\(8\)](#)

Use o `adduser` para adicionar novos usuários, e faça o seguinte:

- Configure `defaultclass = língua` em `/etc/adduser.conf`. Mantenha em mente que você deve colocar uma classe *padrão* para todos usuários de outras línguas neste caso.
- Um variante alternativo é respondendo a língua específica cada vez
- Enter login class: default []:

aparece no [adduser\(8\)](#).

- Outra alternativa é usar o seguinte para cada usuário de língua diferente que você deseja adicionar:
- `# adduser -class língua`

Modificando classes de login com o [pw\(8\)](#)

Se você usa o [pw\(8\)](#) para adicionar novos usuários, rode-o dessa forma:

```
# pw useradd nome_do_usuario -L língua
```

14.3.4.1.2. Método do arquivo de inicialização da shell

Nota: Este método não é recomendado porque ele requer configurações diferentes para cada possível programa shell escolhido. Use o [Método de classes de Login](#) em vez disso.

Para adicionar o nome do locale e a configuração do caracter MIME, basta configurar duas variáveis de ambiente mostradas abaixo nos arquivos de inicialização de shell `/etc/profile` e/ou `/etc/csh.login`. Nós iremos usar a língua alemã como exemplo abaixo:

In `/etc/profile`:

```
LANG=de_DE.ISO8859-1; export LANG
MM_CHARSET=ISO-8859-1; export MM_CHARSET
```

Ou em `/etc/csh.login`:

```
setenv LANG de_DE.ISO8859-1
setenv MM_CHARSET ISO-8859-1
```

Alternativamente, você pode adicionar as instruções acima ao `/usr/share/skel/dot.profile` (similar ao que foi usado acima no `/etc/profile`), ou `/usr/share/skel/dot.login` (similar ao que foi usado acima no `/etc/csh.login`).

Para o X11:

No `$HOME/.xinitrc`:

```
LANG=de_DE.ISO8859-1; export LANG
```

Or:

```
setenv LANG de_DE.ISO8859-1
```

Dependendo de sua shell (veja acima).

14.3.5. Configuração do Console

Para todos os caracteres C chars configurados, configure as fontes corretas do console no `/etc/rc.conf` para a língua em questão com:

```
font8x16=nome_da_fonte
font8x14=nome_da_fonte
font8x8=nome_da_fonte
```

O `nome_da_fonte` aqui é pego do diretório `/usr/share/syscons/fonts`, sem o sufixo `.fnt`.

Também tenha certeza de ter configurado corretamente o keymap e screenmap para o seu caracter C char único configurado através do `/stand/sysinstall`. Dentro do **sysinstall**, escolha Configure, depois Console. Alternativamente, você pode adicionar o seguinte ao `/etc/rc.conf`:

```
scrnmap=nome_da_screenmap
keymap=nome_da_keymap
keychange="sequência de número_fkey"
```

O `screenmap_name` aqui é pego do diretório `/usr/share/syscons/scrnmaps`, sem o sufixo `.scm`. O screenmap com a correspondente fonte mapeada é necessária como correção para expandir 8 bit para 9 bit em fontes de adaptadores VGA com matrix de caracteres em área de gráficos pseudos, ex. para mover letras fora da área se a fonte da tela usar uma coluna de 8 bit.

Se você tem o daemon **moused** habilitado setando o seguinte em seu `/etc/rc.conf`:

```
moused_enable="YES"
```

então examina a informação do cursor do mouse no próximo parágrafo.

Por padrão o cursor do mouse do driver [syscons\(4\)](#) ocupa uma escala 0xd0-0xd3 na configuração de caracter. Se sua língua usa esta escala, você precisa mover a escala do cursor para fora dela. Para habilitar a correção disso para versões do FreeBSD antes da 5.0, insira a seguinte linha em sua configuração do kernel:

```
options          SC_MOUSE_CHAR=0x03
```

Para versões do FreeBSD 4.4 e acima, insira a seguinte linha no `/etc/rc.conf`:

```
mousechar_start=3
```

O `nome_da_keymap` aqui é pego do diretório `/usr/share/syscons/keymaps`, sem o sufixo `.kbd`. Se você não tem certeza que keymap usar, você pode usar o [kbdmap\(1\)](#) para testar keymaps sem reboatar.

O `keychange` é normalmente necessário para teclas de função de programas para achar o tipo de terminal selecionado porque as seqüências de teclas de funções não podem ser definidas no mapa de teclado.

Esteja certo também de ter configurado corretamente o tipo de terminal de console em `/etc/ttys` para todas entradas `ttv*`. As correspondências pré definidas atuais são:

Configuração de caracter Tipo de terminal

ISO-8859-1 ou ISO-8859-15 `cons2511`

ISO-8859-2 `cons2512`

ISO-8859-7 `cons2517`

Configuração de caracter Tipo de terminal

KOI8-R	<i>cons25r</i>
KOI8-U	<i>cons25u</i>
CP437 (padrão VGA)	<i>cons25</i>
US-ASCII	<i>cons25w</i>

Para línguas de caracteres wide ou multibyte, use o port do FreeBSD correto em seu diretório `/usr/ports/língua`. Alguns ports aparecem como console enquanto o sistema os enxerga como uma vty serial, daqui você deve reservar bastante vty's para o X11 e o console pseudo-serial. Aqui está a lista parcial de aplicações para usar outras línguas no console:

Língua	Localização
Chinês tradicional(BIG-5)	chinese/big5con
Japonês	japanese/ja-kon2-* or japanese/Mule_Wnn
Coreano	korean/ko-han

14.3.6. Configuração do X11

Embora o X11 não faça parte do projeto FreeBSD, nós incluímos alguma informação aqui para os usuários do FreeBSD. Para mais detalhes, consulte o [XFree86™ site web](#) ou qualquer servidor X11 que você usa.

No `~/ .Xresources`, você pode adicionalmente ajustar aplicações específicas a configuração I18N (ex., fontes, menus, etc.).

14.3.6.1. Mostrando fontes

Instale o servidor X11 TrueType® Common ([x11-servers/XttXF86srv-common](#)) e instale as fontes de língua TrueType. Configurando o locale correto, deve permitir você visualizar sua língua selecionada nos menus e semelhantes.

14.3.6.2. Colocando caracteres não-inglês

O protocolo de método de entrada do X11 (XIM) é um padrão novo para todos clientes X11. Todas aplicações do X11 devem ser escritas como clientes XIM que fazem entrada do servidor XIM. Há diversos servidores XIM disponíveis para diferentes línguas.

14.3.7. Configuração de impressora

Alguns caracteres C chars únicos configurados são geralmente códigos de hardware nas impressoras. Caracter wide ou multibyte configurados requerem configuração especial e nós recomendamos usar o **apsfilter**. Você pode também converter o documento para PostScript® ou formato PDF usando convertor de língua específica.

14.3.8. Kernel e Sistema de arquivos

O FreeBSD fast filesystem (FFS) é limpo 8-bit, então ele pode ser usado com qualquer caracter único C chars configurado (veja [multibyte\(3\)](#)), mas não há nenhum nome de caracter configurado armazenado no sistema de arquivos; ex., é 8-bitcrú e não sabe nada sobre a ordem de codificação. Oficialmente, o FFS não suporta nenhuma forma de caracteres wide ou multibyte configurados ainda. Porém, alguns caracteres wide ou multibyte tem configurados patches independentes para o FFS habilitando o suporte. São somente soluções importáveis provisórias ou hacks e nós decidimos não incluí-las na árvore de código fonte. Consulte web sites de línguas respectivas para mais informações e arquivos de patch.

O sistema de arquivos MS-DOS® do FreeBSD tem a habilidade configurável de converter-se entre o MS-DOS, o caracter Unicode configurado e escolher a configuração de caracter no sistema de arquivos do FreeBSD. Veja [mount_msdos\(8\)](#) para detalhes.

14.4. Compilando programas I18N

Muitos dos ports do FreeBSD foram portados com o suporte a I18N. Alguns deles são marcados com -I18N no nome do port. Estes e muitos outros programas são compilados com suporte a I18N e não precisam de nenhuma consideração especial.

No entanto, algumas aplicações como o **MySQL** precisam que o `Makefile` seja configurado com um específico charset. Isto normalmente é feito no `Makefile` ou é feito passando o valor para o **configure** no fonte.

14.5. Localizando o FreeBSD para línguas específicas

14.5.1. Língua Russa (Codificaçãotilde;o KOI8-R)

Originalmente contribuído por Andrey A. Chernov.

Para mais informações sobre codificação KOI8-R , veja [Referências do KOI8-R \(Russian Net Character Set\)](#).

14.5.1.1. Configuração do Locale

Coloque as seguintes linhas em seu arquivo `~/.login_conf`:

```
me:Minha conta:\
  :charset=KOI8-R:\
  :lang=ru_RU.KOI8-R:
```

Veja anteriormente nesse capítulo para exemplos de configuração do [locale](#).

14.5.1.2. Configuração do console

- Para versões do FreeBSD antes da 5.0 adicione a seguinte linha em seu arquivo de configuração do kernel:
- `options SC_MOUSE_CHAR=0x03`

Para versões do FreeBSD 4.4 e acima, insira a seguinte linha no `/etc/rc.conf`:

```
mousechar_start=3
```

- Use a seguinte configuração no `/etc/rc.conf`:
- `keymap="ru.koi8-r"`
- `scrnmap="koi8-r2cp866"`
- `font8x16="cp866b-8x16"`
- `font8x14="cp866-8x14"`
- `font8x8="cp866-8x8"`
- Para cada entrada `ttv*` no `/etc/ttys`, use o `cons25r` como tipo de terminal.

Veja anteriormente neste capítulo exemplos de configuração do [console](#).

14.5.1.3. Configuração de impressora

Desde que a maioria das impressoras com caracteres russo vem com a página CP866 no código de hardware, um filtro especial de saída é necessário para converter de KOI8-R para CP866. Tal filtro é instalado por padrão em `/usr/libexec/lpr/ru/koi2alt`. Uma entrada da impressora Russa no `/etc/printcap` deve ser assim:

```
lp|Linha local de impressora Russa:\
  :sh:of=/usr/libexec/lpr/ru/koi2alt:\
  :lp=/dev/lpt0:sd=/var/spool/output/lpd:lf=/var/log/lpd-errs:
```

Veja o [printcap\(5\)](#) para uma descrição detalhada.

14.5.1.4. MS-DOS® FS e nome de arquivos Russos

O seguinte exemplo de entrada do [fstab\(5\)](#) habilita o suporte para nomes de arquivos Russos montados em um sistema de arquivos do MS-DOS®:

```
/dev/ad0s2          /dos/c  msdos  rw,-Wkoi2dos,-Lru_RU.KOI8-R 0 0
```

A opção *-L* seleciona o nome local usado, e o *-W* configura a conversão da tabela de caracter. Para usar a opção *-W*, certifique-se de montar o `/usr` antes da partição MS-DOS porque a conversão de tabelas estão localizadas em `/usr/libdata/msdosfs`. Para mais informações, veja a página de manual do [mount_msdos\(8\)](#).

14.5.1.5. Configuração do X11

1. Faça primeiro [a configuração do X não locale](#) como descrito.

Nota: O locale Russo KOI8-R pode não funcionar com versões antigas do **XFree86™** (abaixo da 3.3). **XFree86 4.X** é agora a versão padrão do sistema de janelas X no FreeBSD. Esta não deve ser a questão a menos que você esteja usando uma versão antiga do FreeBSD.

2. Vá para o diretório [russian/X.language](#) e dê o seguinte comando:
3. `# make install`

O port acima instalará a última versão das fontes do KOI8-R. O **XFree86 3.3** já tem algumas fontes KOI8-R, mas estas são melhor escaladas.

Cheque a seção "*Files*" em seu arquivo `/etc/XF86Config`. As seguintes linhas devem ser adicionadas *antes* de qualquer outra entrada *FontPath*:

```
FontPath "/usr/X11R6/lib/X11/fonts/cyrillic/misc"
FontPath "/usr/X11R6/lib/X11/fonts/cyrillic/75dpi"
FontPath "/usr/X11R6/lib/X11/fonts/cyrillic/100dpi"
```

Se você usa um modo de resolução alta de video, troque as linhas 75 dpi e 100 dpi.

4. Para habilitar o teclado Russo, adicione o seguinte na seção "*Keyboard*" do seu arquivo `XF86Config`.

Para versões 3.X do **XFree86**:

```
XkbLayout "ru"
XkbOptions "grp:caps_toggle"
```

Para versões 4.X do **XFree86**:

```
Option "XkbLayout" "ru"
Option "XkbOptions" "grp:caps_toggle"
```

Também certifique-se que o *XkbDisable* está desabilitado (comentando) lá.

O switch do RUS/LAT será o **CapsLock**. As funções antigas do **CapsLock** ainda estarão disponíveis através do **Shift+CapsLock** (no modo LAT somente).

Se você tem teclas ``Windows®" em seu teclado, e observar que algumas teclas não alfabéticas estão sendo mapeadas incorretamente no modo RUS, adicione a seguinte linha em seu arquivo `XF86Config`.

Para versões 3.X **XFree86**:

```
XkbVariant "winkeys"
```

Para versões 4.X do **XFree86**:

```
Option "XkbVariant" "winkeys"
```

Nota: O teclado Russo XKB pode não funcionar com versões antigas do **XFree86**, veja a nota [acima](#) para mais informações. O teclado Russo XKB pode também não funcionar bem com aplicações não localizadas. Aplicações minimamente localizadas devem chamar a função `XtSetLanguageProc (NULL, NULL, NULL)`; no programa. Veja [KOI8-R para X Window](#) para mais instruções de aplicações localizadas do X11.

14.5.2. Tradicional localização Chinês para Taiwan

O projeto FreeBSD-Taiwan tem um tutorial I18N/L10N para o FreeBSD em <http://freebsd.sinica.edu.tw/~ncvs/zh-l10n-tut/> usando muitos ports Chineses. O editor para o *zh-L10N-tut* é o Clive Lin <Clive@CirX.org>. Você pode também fazer o cvsup das seguintes coleções no endereço `freebsd.sinica.edu.tw`:

Coleção	Descrição
outta-port tag=.	qualidade Beta de coleção de ports para chinês
zh-L10N-tut tag=.	Tutorial de localização do tradicional BIG-5 Chinês
zh-doc tag=.	Tradução de documentação do FreeBSD para o tradicional BIG-5 Chinês

Chuan-Hsing Shen <s874070@mail.yzu.edu.tw> criou o [Coleção Chinês do FreeBSD \(CFC\)](#) usando o FreeBSD-Taiwan *zh-L10N-tut*. Os pacotes e os scripts de arquivos estão disponíveis em <ftp://ftp.csie.ncu.edu.tw/OS/FreeBSD/taiwan/CFC/>.

14.5.3. Localização de língua Alemã (para todas línguas ISO 8859-1)

Slaven Rezic <eserte@cs.tu-berlin.de> escreveu um tutorial de como usar umlauts em uma máquina FreeBSD. O tutorial está escrito em alemão e está disponível em <http://www.de.FreeBSD.org/de/umlaute/>.

14.5.4. Localização de língua Japonesa e Coreana

Para Japonês, consulte <http://www.jp.FreeBSD.org/>, e para Coreano, consulte <http://www.kr.FreeBSD.org/>.

14.5.5. Documentação não inglesa do FreeBSD

Alguns contribuidores do FreeBSD tem traduzido partes do FreeBSD para outras línguas. Eles estão disponíveis através dos links no [site principal](#) ou em `/usr/share/doc`.

Capítulo 15. Aplicações de *Desktop*

Índice

- 15.1. [Sinópse](#)
- 15.2. [Navegadores](#)
- 15.3. [Ferramentas de produtividade](#)
- 15.4. [Visualizadores de Documentos](#)
- 15.5. [Aplicações Financeiras](#)
- 15.6. [Sumário](#)

Contribuição de Christophe Juniet.

15.1. Sinópse

O FreeBSD pode executar uma grande variedade de aplicações de *Desktop*, como navegadores a processadores de texto. Muitos destes estão disponíveis como pacotes ou podem ser automaticamente compilados da coleção de *ports*. Muitos usuários novos esperam encontrar este tipo de aplicação em seu *desktop*. Este capítulo irá lhe apresentar como dispor destas aplicações sem esforço, seja através de pacotes ou pela coleção de *ports*.

Note que quando programas são instalados pelos *ports*, eles são compilados a partir do código fonte. Isso pode tomar muito tempo, dependendo do que você está compilando e da capacidade de processamento de sua(s) máquina(s). Caso a compilação tome tempo em demasia para você, ainda é possível instalar a maioria dos programas da coleção de *ports* dos pacotes pré-compilados.

Devido ao FreeBSD ter como característica a compatibilidade com binários de Linux, muitas das aplicações originalmente desenvolvidas para Linux estão disponíveis para o seu *desktop*. É extremamente recomendado que você leia o [Capítulo 22](#) antes de instalar qualquer aplicação para Linux. Muitos dos *ports* que usam a compatibilidade com binários de Linux começam com `linux-`. Lembre-se disso quando procurar por algum *port*, a propósito use [whereis\(1\)](#). No texto adiante, assume-se que você ativou a compatibilidade com binários de Linux antes de instalar qualquer das aplicações para Linux.

Estas são as categorias cobertas neste capítulo:

- Navegadores (como **Mozilla**, **Netscape®**, **Opera**)
- Ferramentas de produtividade (como **KOffice**, **AbiWord**, **The GIMP**, **OpenOffice.org**)

- Visualizadores (como **Acrobat Reader®**, **gv**, **Xpdf**, **GQview**)
- Aplicações Financeiras (como **GnuCash**, **Gnumeric**, **Abacus**)

Depois de ler este capítulo, você deve:

- Aprender a instalar *software* de terceiros ([Capítulo 4](#)).
- Aprender a instalar *software* para Linux ([Capítulo 22](#)).

Para mais informações sobre como definir um ambiente multimídia, leia [Capítulo 16](#). Caso pretenda utilizar o correio eletrônico, consulte [Capítulo 20](#).

15.2. Navegadores

O FreeBSD não vem com nenhum navegador em particular pré-instalado. Porém, o diretório [www](#) da coleção de *ports* contém um grande número de navegadores prontos para serem instalados. Caso não tenha tempo para compilar tudo (isto pode tomar um tempo bem longo em alguns casos) muitos deles estão disponíveis como pacotes.

KDE e **GNOME** também dispõem de navegadores HTML. Por favor, consulte a [Seção 5.7](#) para mais informação sobre como configurar estes *desktops* completos.

Caso esteja procurando por navegadores mais simples, você deve investigar a coleção de *ports* a procura de: [www/dillo](#), [www/links](#), or [www/w3m](#).

Esta seção cobre estas aplicações

Nome da Aplicação	Recursos Necessários	Instalação via <i>ports</i>	Principais Dependências
Mozilla	pesado	pesado	Gtk+
Netscape®	pesado	leve	Compatibilidade de binários Linux
Opera	leve	leve	Versão FreeBSD: não. Versão Linux: Compatibilidade de binários Linux e linux-openmotif

15.2.1. Mozilla

Mozilla é o navegador provavelmente mais adaptado ao *Desktop* FreeBSD. Ele é moderno, estável, e totalmente convertido para FreeBSD. Seu mecanismo de renderização atende a uma ampla gama de padrões HTML. Provê correio eletrônico e leitor de *news*. Também dispõe de um editor de HTML caso você planeje escrever algumas páginas. Os usuários do **Netscape** vão reconhecer as semelhanças com o **Communicator**, já que ambos os navegadores compartilham da mesma base.

Em máquinas lentas com velocidades de CPU abaixo de 233MHz ou com menos de 64MB de RAM, **Mozilla** pode consumir muitos recursos para ser plenamente utilizável. Você deve alternativamente buscar utilizar o navegador **Opera**, descrito um pouco mais adiante neste capítulo.

Caso você não queira ou não possa compilar o **Mozilla** por alguma razão, o grupo FreeBSD GNOME já fez isso para você. Basta instalar o pacote pela rede com:

```
# pkg_add -r mozilla
```

Havendo indisponibilidade do pacote, e falta de tempo e espaço em seus discos, você pode obter o código fonte do **Mozilla**, compilá-lo e instalá-lo. Isto pode ser realizado com:

```
# cd /usr/ports/www/mozilla
# make install clean
```

O *port Mozilla* certifica a correta inicialização executando a configuração do registro chrome com privilégios de `root`. Contudo, se você desejar obter alguns aditivos como animações no *mouse*, você deve executar **Mozilla** como `root` para tê-lo devidamente instalado.

Uma vez que seja concluída a instalação do **Mozilla**, você não necessitará mais do status de `root`. Você pode iniciar o **Mozilla** como navegador digitando:

```
% mozilla
```

Você pode inicializá-lo diretamente como cliente de email ou *news* conforme abaixo:

```
% mozilla -mail
```

15.2.2. Mozilla, Java™, e Macromedia® Flash™

Contribuição de Tom Rhodes.

Instalar o **Mozilla** é simples, mas instalar **Mozilla** com suporte para aditivos como Java™ e Macromedia® Flash™, infelizmente, consome simultaneamente tempo e espaço em disco.

A primeira coisa a fazer é baixar da internet os arquivos que vão ser usados com **Mozilla**. Aponte seu navegador web para <http://www.sun.com/software/java2/download.html> e crie uma conta neste sítio www. Lembre-se de guardar seu nome de usuário e senha neste sítio pois serão necessários no futuro. Baixe uma cópia do arquivo `j2sdk-1_3_1-src.tar.gz` e coloque-o em `/usr/ports/distfiles/` visto que o *port* não irá pegá-lo automaticamente. Este procedimento deve-se a limitações impostas pela licença do programa. Enquanto isso, baixe o "ambiente java" de <http://java.sun.com/webapps/download/Display?BundleId=7905>. O arquivo chama-se `j2sdk-1_3_1_08-linux-i586.bin` e é grande (aproximadamente 25 megabytes!). Como antes, este arquivo deve ser colocado em `/usr/ports/distfiles/`. Finalmente baixe uma cópia do "java patchkit" de <http://www.eyesbeyond.com/freebsd/om/java/> e coloque em `/usr/ports/distfiles/`.

Instale o *port* [java/jdk13](#) como de costume com `make install clean` e então instale o *port* do [www/flashpluginwrapper](#). Este *port* requer [emulators/linux_base](#) que é um *software* grande. É fato que existam outros *plugins* **Flash**, contudo, não funcionaram comigo.

Instale o *port* [www/mozilla](#), caso ainda não esteja instalado.

Agora copie os arquivos do plug-in **Flash** com:

```
# cp /usr/local/lib/flash/libflashplayer.so \  
    /usr/X11R6/lib/browser_plugins/libflashplayer_linux.so  
# cp /usr/local/lib/flash/ShockwaveFlash.class \  
    /usr/X11R6/lib/browser_plugins/
```

Nota: Caso esteja usando o [www/mozilla-devel](#), os diretórios de destino serão diferentes.

Agora adicione as seguintes linhas no topo do *script* de inicialização do **Mozilla** (imediatamente após o `#!/bin/sh`): `/usr/X11R6/bin/mozilla`.

```
LD_PRELOAD=/usr/local/lib/libflashplayer.so.1  
export LD_PRELOAD
```

Isto vai habilitar o **Flash** plug-in.

Agora basta iniciar o **Mozilla** com:

```
% mozilla &
```

E acessar a opção About Plug-ins do *menu* Help. Uma lista com os *plugins* disponíveis deve aparecer. Ambos, **Java** e **Shockwave® Flash**, devem estar listados.

15.2.3. Netscape®

A coleção de *ports* contém diversas versões do navegador Netscape. Desde que as versões nativas do FreeBSD contém sérias falhas de segurança, instalá-las é algo fortemente desencorajado. Alternativamente, opte por uma versão mais recente para Linux ou DIGITAL UNIX

Netscape 7 é a última versão do navegador Netscape. Ela pode ser instalada a partir da coleção de *ports*:

```
# cd /usr/ports/www/netscape7  
# make install clean
```

Estão disponíveis também versões localizadas nas categorias Francês, Alemão e Japonês.

Cuidado A série **Netscape 4.x** não é recomendada porque não está de acordo com os padrões atuais. Contudo, **Netscape 7.x** e versões mais novas são as únicas disponíveis para a plataforma i386™

15.2.4. Opera

Opera um navegador muito rápido, amplamente funcional e adequado a padrões. Está disponível em dois sabores: uma versão "nativa" do FreeBSD e outra que executa sob emulação Linux. Para cada sistema operacional há uma versão gratuita do navegador que apresenta propagandas e uma livre de propagandas que pode ser adquirida em [Opera web site](#).

Para navegar pela rede mundial de computadores com uma versão do **Opera** para FreeBSD instale o pacote:

```
# pkg_add -r opera
```

Alguns sítios de FTP não têm todos os pacotes, mas o mesmo resultado pode ser obtido com a coleção de *ports* digitando-se:

```
# cd /usr/ports/www/opera
# make install clean
```

Para instalar a versão Linux do **Opera**, no exemplo acima substitua *linux-opera* por *opera*. A versão Linux é bastante útil em situações onde sejam necessários *plugins* apenas disponíveis para Linux, como **Adobe Acrobat Reader®**. Nos demais quisitos a versão para FreeBSD é funcionalmente idêntica a do Linux.

15.3. Ferramentas de produtividade

Quando chegamos a produtividades, novos usuários geralmente procuram por bons pacotes de ferramentas de escritório ou um processador de textos amigável. Ao passo que alguns [ambientes de desktop](#) como o **KDE** já fornecem um pacote de ferramentas de escritório, não há uma aplicação padrão. FreeBSD fornece tudo que for necessário, seja qual for seu *desktop*.

Esta seção cobre as seguintes aplicações

Nome da Aplicação	Recursos Necessários	Instalação via <i>ports</i>	Principais dependências
KOffice	leve	pesado	KDE
AbiWord	leve	leve	Gtk+ ou GNOME
The Gimp	leve	pesado	Gtk+
OpenOffice.org	pesado	excessivamente pesado	GCC 3.1, JDK™ 1.3, Mozilla

15.3.1. KOffice

A comunidade KDE oferece este ambiente de *desktop* com uma suíte de ferramentas de escritório que pode ser usada fora do **KDE**. Ela inclui os quatro componentes padrão que podem ser encontrados nas outras suítes de ferramentas de escritório. **KWord** é o processador de textos, **KSpread** é o programa de planilhas eletrônicas, **KPresenter** organiza apresentações de slides, e **Kontour** que permite desenhar documentos gráficos.

Antes de instalar o **KOffice** mais recente, certifique-se de ter uma versão atualizada do **KDE**.

Para instalar o **KOffice** a partir de um pacote, proceda com o seguinte comando:

```
# pkg_add -r koffice
```

Caso o pacote não esteja disponível, você pode usar a coleção de *ports*. Assim, para instalar o **KOffice** para **KDE3**, faça:

```
# cd /usr/ports/editors/koffice-kde3
# make install clean
```

15.3.2. AbiWord

AbiWord é um processador de textos livre, similar ao **Microsoft® Word** em aparência e funcionamento. Ele é adequado a digitação de textos, cartas, relatórios, memorandos, etc. É bem rápido, contém muitas funcionalidades, e é de utilização bastante amigável.

AbiWord pode importar e exportar diversos formatos de arquivos, incluindo alguns proprietários como Microsoft *.doc*.

AbiWord está disponível como pacote. Você pode instalá-lo com:

```
# pkg_add -r AbiWord-gnome
```

Não havendo disponibilidade do pacote, ele pode ser compilado a partir da coleção de *ports*. Na coleção de *ports* ele deve estar mais atualizado. Isto pode ser feito com:

```
# cd /usr/ports/editors/AbiWord
# make install clean
```

15.3.3. The GIMP

Para a criação ou retoques em imagens **The GIMP** é uma ferramenta bastante sofisticada. Pode ser usado como um simples programa de pintura ou como uma avançada suíte de edição de imagens. Fornece uma grande variedade de plug-ins e possui uma interface para *scripts*. **The GIMP** pode ler e escrever em um grande universo de formatos de arquivo. Suporta interfaces para *scanners* e *tablets*.

Você pode instalar o pacote informando o comando:

```
# pkg_add -r gimp
```

Na eventualidade de seu sítio FTP não possuir o pacote você pode instalá-lo a partir da coleção de *ports*. O diretório [graphics](#) da coleção de *ports* também dispõe do **The Gimp Manual**. Instale-o assim:

```
# cd /usr/ports/graphics/gimpl
# make install clean
# cd /usr/ports/graphics/gimp-manual-pdf
# make install clean
```

Nota: O diretório [graphics](#) da coleção de *ports* armazena a versão de desenvolvimento do **The GIMP** em [graphics/gimp-devel](#). Versões em HTML e PostScript® do **The Gimp Manual** estão em [graphics/gimp-manual-html](#) e [graphics/gimp-manual-ps](#).

15.3.4. OpenOffice.org

OpenOffice.org inclui todas as aplicações necessárias em uma suíte de ferramentas de escritório: um processador de textos, uma planilha eletrônica, um gerenciador de apresentações, e um programa de desenho. A interface do usuário é bem similar às das demais suítes de escritório, e pode importar e exportar arquivos em vários formatos populares. Está disponível em diversas linguagens incluindo as interfaces, verificadores ortográficos e dicionários.

O processador de textos do **OpenOffice.org** utiliza em seus arquivos XML nativo para uma maior portabilidade e flexibilidade. O programa de planilha eletrônica dispõe de linguagem de macro e pode comunicar-se com bancos de dados externos.

OpenOffice.org é estável; atualmente é executado nativamente em Windows®, Solaris™, Linux, FreeBSD, and Mac OS® X. Mais informações sobre o **OpenOffice.org** podem ser encontradas no [sítio www do OpenOffice](#). Para informação específica sobre FreeBSD, e para baixar da internet o pacote refira-se ao sítio [www do Grupo de Conversão do OpenOffice para FreeBSD](#).

Para instalar **OpenOffice.org**:

```
# pkg_add -r openoffice
```

Uma vez que o pacote esteja instalado você precisa executar o programa de configuração e escolher uma *instalação de estação de trabalho padrão*. Execute este comando como o usuário que irá utilizar **OpenOffice.org**:

```
% openoffice-setup
```

Caso o pacote **OpenOffice.org** não esteja disponível, você ainda pode contar com a compilação do port. Contudo, você deve estar ciente que isso requer bastante espaço em disco e muito tempo de compilação.

```
# cd /usr/ports/editors/openoffice
# make install clean
```

Uma vez que isto foi feito, execute o programa de configuração como o usuário que vai utilizar **OpenOffice.org** e escolha a *instalação de estação de trabalho padrão* com:

```
% cd /usr/ports/editors/openoffice
% make install-user
```

Caso prefira usar uma versão localizada, esta é a variável do *ports*:

Linguagem	<i>Port</i>
Arabe	editors/openoffice-ar
Dinamarquês	editors/openoffice-dk
Espanhol	editors/openoffice-es
Grego	editors/openoffice-gr
Italiano	editors/openoffice-it
Holandês	editors/openoffice-nl
Sueco	editors/openoffice-se
Turco	editors/openoffice-tr
Francês	french/openoffice
Alemão	german/openoffice
Japonês	japanese/openoffice
Coreano	korean/openoffice
Polonês	polish/openoffice
Português	portuguese/openoffice
Russo	russian/openoffice

15.4. Visualizadores de Documentos

Diversos formatos novos de documentos ganharam popularidade recentemente. Os visualizadores padrão exigidos podem não estar disponíveis na base do sistema. Nesta seção vamos ver como instalá-los.

Esta seção cobre as seguintes aplicações:

Nome da Aplicação	Recursos necessários	Instalação via <i>ports</i>	Principais dependências
Acrobat Reader®	leve	leve	Compatibilidade de binários Linux
gv	leve	leve	Xaw3d
Xpdf	leve	leve	FreeType
GQview	leve	leve	Gtk+ ou GNOME

15.4.1. Acrobat Reader®

Muitos documentos são agora distribuídos como arquivos PDF, que significa "Formato de Documento Portátil". Um dos visualizadores recomendados para este tipo de arquivos é **Acrobat Reader**, lançado pela Adobe para Linux. Como FreeBSD pode executar binários de Linux, ele também está disponível para FreeBSD.

Para instalar o pacote do **Acrobat Reader 5** execute:

```
# pkg_add -r acroread5
```

Como de costume, se o pacote não estiver disponível ou, caso você deseje a última versão, pode também usar a coleção de *ports*:

```
# cd /usr/ports/print/acroread5
# make install clean
```

Nota: **Acrobat Reader** está disponível em diversas versões. Durante a escrita deste documento havia: [print/acroread](#) (versão 3.0.2), [print/acroread4](#) (versão 4.0.5), e [print/acroread5](#) (versão 5.0.6). Elas podem não ter sido todas empacotadas para a sua versão de FreeBSD. A coleção de *ports* sempre irá conter a versão mais atual.

15.4.2. gv

gv é um visualizador de PostScript® e PDF. Foi baseado originalmente no **ghostview** mas tem uma aparência melhor graças a biblioteca **Xaw3d**. Ele é rápido e sua interface

é limpa. **gv** tem muitas características como orientação, tamanho do papel, escala e antialias. A maior parte das operações pode ser realizada a partir do teclado ou *mouse*.

Para instalar **gv** como pacote, execute:

```
# pkg_add -r gv
```

Caso não consiga o pacote pode usar a coleção de *ports*:

```
# cd /usr/ports/print/gv
# make install clean
```

15.4.3. Xpdf

Desejando um simples visualizador de PDF para FreeBSD, **Xpdf** é uma ferramenta leve e eficiente. Ele requer muito poucos recursos e é bastante estável. Usa fontes padrão do X e não requer **Motif**® ou qualquer outro *X toolkit*.

Para instalar o pacote **Xpdf**, proceda este comando:

```
# pkg_add -r xpdf
```

Não havendo disponibilidade do pacote, ou caso prefira, use a coleção de *ports*:

```
# cd /usr/ports/graphics/xpdf
# make install clean
```

Assim que a instalação se completar você pode disparar **Xpdf** e usar o botão direito do *mouse* para ativar o *menu*.

15.4.4. GQview

GQview é um visualizador de imagens. Você pode ver um arquivo com um simples clique, disparar um editor externo, visualizar miniaturas, e muito mais. Ele também dispõe de exibição de *slides* e algumas operações básicas com arquivos. Você pode manipular coleções e facilmente identificar duplicatas. **GQview** pode fazer visualizações em tela cheia e suporta internacionalização.

Querendo instalar **GQview** a partir do pacote, execute:

```
# pkg_add -r gqview
```

Não havendo disponibilidade do pacote, ou caso prefira, use a coleção de *ports*:

```
# cd /usr/ports/graphics/gqview
# make install clean
```


15.5. Aplicações Financeiras

Caso, por alguma razão, você pretenda administrar suas finanças pessoais no seu *Desktop* FreeBSD, saiba que existem algumas ferramentas simples e poderosas prontas para serem instaladas. Algumas delas compatíveis com formatos populares de arquivos como aqueles dos documentos **Quicken**® ou **Excel**.

Esta seção cobre as seguintes aplicações:

Nome da Aplicação	Recursos Necessários	Instalação a partir do <i>ports</i>	Principais Dependências
GnuCash	leve	pesado	GNOME
Gnumeric	leve	pesado	GNOME
Abacus	leve	leve	Tcl/Tk

15.5.1. GnuCash

GnuCash é parte dos esforços **GNOME** em prover aplicações amigáveis e para usuários finais. Com **GnuCash**, você pode registrar seus proventos e despesas, suas contas bancárias, seus investimentos na bolsa de valores. Dispõe de uma interface intuitiva ao mesmo tempo que mantém o profissionalismo.

GnuCash fornece registros inteligentes, um sistema de apontamentos hierárquicos, muitas teclas de atalho e métodos de *auto-completion*. Pode segmentar uma transação simples em diversos elementos mais detalhados. **GnuCash** pode importar e concatenar arquivos QIF do **Quicken**. Também manipula a maioria dos formatos internacionais de data e moeda.

Para instalar **GnuCash** em seu sistema, execute:

```
# pkg_add -r gnuCash
```

Caso o pacote não esteja disponível, você pode usar a coleção de *ports*:

```
# cd /usr/ports/finance/gnuCash  
# make install clean
```

15.5.2. Gnumeric

Gnumeric é uma planilha eletrônica, parte do ambiente de *desktop* **GNOME**. Dispõe de um conveniente "convite" para que o usuário entre com dados de acordo com a formatação da célula e de um sistema auto-preenchimento para muitas seqüências. Pode importar arquivos de diversos outros formatos populares como **Excel**, **Lotus 1-2-3**, ou **Quattro Pro**. **Gnumeric** suporta gráficos através do programa gráfico [math/guppi](#).

Tem um grande número de funções pré-programadas e permite todos os formatos usuais de células como números, moedas, datas, tempo e muito mais.

Para instalar **Gnumeric** em seu sistema, digite:

```
# pkg_add -r gnumeric
```

Caso o pacote não esteja disponível, você pode usar a coleção de *ports* executando:

```
# cd /usr/ports/math/gnumeric
# make install clean
```

15.5.3. Abacus

Abacus é uma planilha eletrônica bem simples e muito fácil de usar. Inclui muitas funções internas úteis em diversos domínios como: estatística, finanças e matemática. Pode importar e exportar o formato de arquivo do **Excel**. **Abacus** pode produzir saídas em PostScript®.

Para instalar **Abacus** a partir de um pacote, execute:

```
# pkg_add -r abacus
```

Caso o pacote não esteja disponível, você pode usar a coleção de *ports* executando:

```
# cd /usr/ports/deskutils/abacus
# make install clean
```

15.6. Sumário

Enquanto o FreeBSD é popular dentre os ISPs por sua performance e estabilidade, está prontinho para o uso do dia-a-dia como *desktop*. Com milhares de aplicações disponíveis como [coleção de pacotes](#) ou [ports](#), você pode montar um ótimo *desktop* que vai atender todas suas necessidades.

Uma vez que você conclua a instalação de seu *desktop*, deverá querer ir um passo a frente com [misc/instant-workstation](#). Este "meta-port" permite que você monte um conjunto típico de *ports* para uma estação de trabalho. Você pode customizá-lo editando `/usr/ports/misc/instant-workstation/Makefile`. Siga a sintaxe usada para o conjunto padrão para adicionar ou remover *ports*, e monte-o da maneira usual. No final das contas, você estará apto a criar um grande pacote que corresponda ao seu próprio padrão de *desktop* e instalá-lo nas suas outras estações de trabalho!

Segue uma breve visão de todas as aplicações de *desktop* vistas neste capítulo:

Nome da Aplicação	Nome do Pacote	Nome do Ports
Mozilla	<i>mozilla</i>	www/mozilla
Netscape®	<i>linux-netscape7</i>	www/netscape7

Nome da Aplicação	Nome do Pacote	Nome do Ports
Opera	<i>linux-opera</i>	www/linux-opera
KOffice	<i>koffice-kde3</i>	editors/koffice-kde3
AbiWord	<i>AbiWord-gnome</i>	editors/AbiWord
The GIMP	<i>gimp</i>	graphics/gimpl
OpenOffice.org	<i>openoffice</i>	editors/openoffice
Acrobat Reader®	<i>acroread5</i>	print/acroread5
gv	<i>gv</i>	print/gv
Xpdf	<i>xpdf</i>	graphics/xpdf
GQview	<i>gqview</i>	graphics/gqview
GnuCash	<i>gnucash</i>	finance/gnucash
Gnumeric	<i>gnumeric</i>	math/gnumeric
Abacus	<i>abacus</i>	deskutils/abacus

Capítulo 16. Multimedia

Índice

- 16.1. [Sinótese](#)
- 16.2. [Configurando a placa de som](#)
- 16.3. [Audio MP3](#)
- 16.4. [Reprodução de Video](#)

Edited by Ross Lippert.

16.1. Sinótese

O FreeBSD suporta uma ampla variedade de placas de som, permitindo que você aprecie sons de alta fidelidade a partir do seu computador. Isto inclui a habilidade de gravar e reproduzir audio no formato MP3 (MPEG Audio Layer 3), WAV e Ogg Vorbis assim como muitos outros formatos. A coleção de ports do FreeBSD também possui aplicativos que permitem a edição de arquivos de audio, a adição de efeitos sonoros, e o controle de dispositivos MIDI conectados ao seu computador.

Com alguma disposição para experimentar, o FreeBSD pode reproduzir arquivos de video e DVD's. O número de aplicações para codificar, converter, e reproduzir as várias mídias de video é mais limitada do que o número de aplicações para som. Por exemplo, enquanto este capítulo está sendo escrito, não existe uma boa aplicação de *re-encoding* na coleção de ports do FreeBSD, a qual possa ser utilizada para converter arquivos de um formato para outro, da mesma forma que podemos fazer com arquivos de audio utilizando o [audio/sox](#). Entretanto, o cenário nesta área de software está mudando rapidamente.

Este capítulo irá descrever os passos necessários para configurar a sua placa de som. A instalação e configuração do XFree86™ ([Capítulo 5](#)) já foi abordada, e na ocasião

foram tomados todos os cuidados necessários em relação a correta configuração da sua placa de vídeo, embora possam existir alguns ajustes a serem aplicados para uma melhor reprodução de vídeos.

Após a leitura deste capítulo, você saberá:

- Como configurar o seu sistema de forma que a sua placa de som seja reconhecida.
- Métodos para testar se sua placa de som está funcionando, utilizando aplicativos de exemplo.
- Como realizar uma pesquisa de defeitos em sua configuração de som.
- Como reproduzir e encodar arquivos de áudio MP3 e em outros formatos.
- Como o vídeo é suportado pelo **XFree86**.
- Alguns ports de reprodutores/codificadores os quais apresentam bons resultados.
- Como reproduzir DVD's, e arquivos .mpg e .avi.
- Como reparar informações de CDs e DVDs para arquivos.

Antes de ler este capítulo, você deve:

- Conhecer como configurar e instalar um novo kernel ([Capítulo 9](#)).

Para as sessões de vídeo, assumimos que o **XFree86 4.X** ([x11/XFree86-4](#)) já está instalado. O **XFree86 3.X** pode funcionar, mas ele não foi testado com o que é descrito neste capítulo. Se você constatar que algo descrito aqui, funciona com o **XFree86 3.X**, por favor nos informe.

Atenção Tentar montar um CD de áudio ou um DVD de vídeo com o comando [mount\(8\)](#) irá resultar ao menos em um erro, e no pior dos casos em um *kernel panic*. Estas mídias possuem uma codificação especializada, a qual difere do sistema de arquivos ISO usual.

16.2. Configurando a placa de som

Contribuído por *Moses Moore*.

16.2.1. Localizando o dispositivo correto

Antes de você começar, você deve saber qual o modelo da placa que você possui, qual chipset ela utiliza, e se ela é uma placa PCI ou ISA. O FreeBSD suporta uma larga variedade de placas PCI e ISA. Se você não localizar a sua placa na lista seguinte, verifique a página de manual [pcm\(4\)](#). Esta não é uma lista completa; entretanto ela lista algumas das placas mais comuns.

- Crystal 4237, 4236, 4232, 4231
- Yamaha OPL-SAx
- OPTi931
- Ensoniq AudioPCI 1370/1371

- ESS Solo-1/1E
- NeoMagic 256AV/ZX
- SoundBlaster® Pro, 16, 32, AWE64, AWE128, Live
- Creative ViBRA16
- Advanced Asound 100, 110, e Logic ALS120
- ES 1868, 1869, 1879, 1888
- Gravis UltraSound
- Aureal Vortex 1 ou 2

Para utilizar o seu dispositivo de som, você irá precisar carregar o driver de dispositivo apropriado. Isto pode ser efetuado de duas formas. A forma mais simples consiste-se em carregar o módulo de kernel apropriado para sua placa de som com o comando [kldload\(8\)](#) o que pode ser feito a partir da linha de comando:

```
# kldload snd_emul0k1.ko
```

ou através da adição de uma linha apropriada no arquivo `/boot/loader.conf`, tal como o exemplo abaixo:

```
snd_emul0k1_load="YES"
```

Estes exemplos são para uma placa de som Creative SoundBlaster Live!. Outros módulos de som carregáveis disponíveis são listados no `/boot/defaults/loader.conf`.

Alternativamente, você pode compilar de forma estática o suporte à sua placa de som no seu kernel. A sessão abaixo fornece as informações que você precisa para adicionar o suporte ao seu hardware desta maneira. Para maiores informações sobre a recompilação do seu kernel, por favor consulte [Capítulo 9](#).

16.2.1.1. Placas de som Creative, Advance, e ESS

Se você possui uma das placas abaixo, você irá precisar adicionar:

```
device pcm
```

Ao seu arquivo de configuração do kernel. Se você possui uma placa ISA PnP, você também terá que adicionar:

```
device sbc
```

Para uma placa não-PnP, adicione:

```
device pcm
device sbc0 at isa? port 0x220 irq 5 drq 1 flags 0x15
```

Ao seu arquivo de configuração do kernel. A configuração mostrada acima é a padrão. Você pode precisar alterar o IRQ ou outras configurações para adaptar-se a sua placa. Consulte a página de manual [sbc\(4\)](#) para maiores informações.

Nota: A placa de som Sound Blaster Live não é suportada no FreeBSD FreeBSD 4.0 sem um *patch*, o qual esta sessão não irá abranger. É recomendado que você atualize seu sistema para a última versão -STABLE antes de tentar utilizar esta placa.

16.2.1.2. Placas Gravis UltraSound

Para um placa ISA PnP, você irá precisar adicionar:

```
device pcm
device gusc
```

Ao seu arquivo de configuração do kernel. se você possui uma placa ISA não-PnP, você irá precisar adicionar:

```
device pcm
device gus0 at isa? port 0x220 irq 5 drq 1 flags 0x13
```

Ao seu arquivo de configuração do kernel. A configuração mostrada acima é a padrão. Você pode precisar alterar o IRQ ou outras configurações para adaptar-se a sua placa. Consulte a página de manual [gusc\(4\)](#) para maiores informações.

16.2.1.3. Placas de som Crystal

Para placas Crystal, você irá precisar adicionar:

```
device pcm
device csa
```

Ao seu arquivo de configuração do kernel.

16.2.1.4. Suporte Genérico

Para placas PCI ou ISA PnP, você irá precisar adicionar:

```
device pcm
```

Ao seu arquivo de configuração do kernel. Se você possui uma placa de som ISA não-PnP a qual não possui um *bridge driver*, você irá precisar adicionar:

```
device pcm0 at isa? irq 10 drq 1 flags 0x0
```

Ao seu arquivo de configuração do kernel. A configuração mostrada acima é a padrão. Você pode precisar alterar o IRQ ou outras configurações para adaptar-se a sua placa.

16.2.1.5. Som Onboard

Alguns sistemas com dispositivos de som embutidos na placa mãe podem precisar que a seguinte opção esteja presente em seu arquivo de configuração do kernel:

```
options PNPBIOS
```

16.2.2. Criando e testando os nós de dispositivos (Device Nodes)

Depois que você reinicializar, logue no sistema e verifique se o seu dispositivo está sendo listado no arquivo `/var/run/dmesg.boot`, como mostrado abaixo:

```
# grep pcm /var/run/dmesg.boot
pcm0: <SB16 DSP 4.11> on sbc0
```

A saída para o seu sistema pode parecer diferente. Se o dispositivo `pcm` não for exibido, algo de errado ocorreu anteriormente. Se isto acontecer, verifique novamente o arquivo de configuração do seu kernel e certifique-se de que você escolheu o dispositivo correto. Os problemas mais comuns estão listados em [Seção 16.2.2.1](#).

Nota: Se você estiver executando o FreeBSD 5.0 ou posterior, você pode pular tranquilamente o resto desta sessão. Estas versões utilizam um [devfs\(5\)](#) para criar automaticamente os nós de dispositivos.

Se o comando prévio retornou `pcm0`, você terá que executar os seguintes comandos, logado como `root`:

```
# cd /dev
# sh MAKEDEV snd0
```

Se o comando retornou `pcm1`, siga os mesmos passos mostrados acima, substituindo `snd0` por `snd1`.

Nota: Os comandos acima *não* irão criar o dispositivo `/dev/snd!`

O `MAKEDEV` irá criar um grupo de nós de dispositivo, incluindo:

Dispositivo	Descrição
<code>/dev/audio</code>	Dispositivo de audio compatível com padrão Sparc®
<code>/dev/dsp</code>	Dispositivo de voz digitalizada
<code>/dev/dspW</code>	Semelhante ao <code>/dev/dsp</code> , mas com uma amostragem de 16 bits
<code>/dev/midi</code>	Acesso direto ao dispositivo midi
<code>/dev/mixer</code>	Porta de controle do dispositivo de mixagem
<code>/dev/music</code>	Interface nível 2 do sequenciador
<code>/dev/sequencer</code>	Dispositivo sequenciador
<code>/dev/pss</code>	Interface de dispositivo programável

Se tudo correu bem, agora você deve ter uma placa de som que funciona. Se o seu drive de CD-ROM ou de DVD-ROM estiver devidamente conectado a sua placa de som, você pode colocar um CD no drive e reproduzi-lo com o uso do [cdcontrol\(1\)](#):

```
% cdcontrol -f /dev/acd0c play 1
```

Várias aplicações, tal como o [audio/workman](#) oferece uma interface melhor. Você pode desejar instalar uma aplicação tal como o [audio/mpg123](#) para ouvir seus arquivos de audio no formato MP3.

16.2.2.1. Problemas Comuns

Erro	Solução
<code>``unsupported subdevice XX``</code>	Um ou mais dos nós de dispositivo não foram criados corretamente. Repita os passos acima.
<code>``sb_dspwr(XX) timed out``</code>	A porta de I/O não esta ajustada corretamente.
<code>``bad irq XX``</code>	O IRQ está ajustado de forma incorreta. Certifique-se de que o IRQ setado é o mesmo IRQ utilizado pela sua placa.
<code>``xxx: gus pcm not attached, out of memory``</code>	Não existe memória suficiente para utilizar o dispositivo.
<code>``xxx: can't open /dev/dsp!``</code>	Verifique com o comando <code>fstat grep dsp</code> se alguma outra aplicação esta mantendo o dispositivo aberto. Criadores de problema dignos de atenção são o esound e o suporte a som do KDE .

16.2.3. Utilizando múltiplas fontes de som

Contribuido por Munish Chopra.

É freqüentemente desejável termos múltiplas fontes de som, as quais possam ser reproduzidas simultaneamente, como por exemplo quando o **esound** ou o **artsd** não suportam o compartilhamento do dispositivo de som com alguma aplicação.

O FreeBSD permite que você faça isso através dos *Canais virtuais de som*, os quais podem ser configurados através do [sysctl\(8\)](#). Os canais virtuais permitem que você multiplexe os canais de reprodução da sua placa de som através de uma mixagem do som pelo kernel.

Para configurar os canais virtuais, existem 2 variáveis do `sysctl` que precisam ser ajustadas, se você é o usuário `root`, isto pode ser feito da seguinte forma:

```
# sysctl hw.snd.pcm0.vchans=4
# sysctl hw.snd.maxautovchans=4
```


O exemplo acima define 4 canais virtuais, o qual é um número razoável para uso no dia a dia. A variável `hw.snd.pcm0.vchans` define o número de canais virtuais que o `pcm0` possui, e ela é configurável uma vez que o dispositivo tenha sido atachado. Já a variável `hw.snd.maxautovchans` define o número de canais virtuais atribuído a um dispositivo atachado via [kldload\(8\)](#). Uma vez que o módulo `pcm` pode ser carregado independentemente dos drivers de hardware, a variável `hw.snd.maxautovchans` pode armazenar quantos canais virtuais serão atribuídos a qualquer dispositivo de som atachado após a inicialização do sistema.

Se você não está utilizando o [devfs\(5\)](#), você terá que apontar a sua aplicação para `/dev/dsp0.x`, onde `x` será um valor de 0 a 3 se a variável `hw.snd.pcm.0.vchans` estiver definida para 4 como no exemplo anterior. Em um sistema que utilize o [devfs\(5\)](#), a atribuição ocorre de forma automática e transparente ao usuário.

16.3. Audio MP3

Contribuído por Chern Lee.

Os arquivos de audio MP3 (MPEG Layer 3) oferecem uma qualidade de som próxima a qualidade de um CD de audio convencional, de forma que não existe nenhuma razão para que você deixe a sua estação de trabalho FreeBSD sem suporte aos mesmos.

16.3.1. Reprodutores de MP3

De longe, o reprodutor MP3 mais popular no ambiente **XFree86™** é o **XMMS** (X Multimedia System). Os skins do **Winamp** podem ser utilizados com o **XMMS** uma vez que a GUI é quase idêntica a do Nullsoft's **Winamp**. O **XMMS** também possui um suporte nativo a plugins.

O **XMMS** pode ser instalado a partir do port [multimedia/xmms](#) ou então através de um pacote binário.

A interface do **XMMS'** é intuitiva, com uma playlist, equalizador gráfico, etc. Aqueles que estão familiarizados com o **Winamp** irão achar o **XMMS** simples de se utilizar.

O port [audio/mpg123](#) é uma alternativa, visto que é um reprodutor de mp3 de linha de comando.

O **mpg123** pode ser executado, especificando como parametros na linha de comando, o dispositivo de som e o arquivo MP3, como ilustrado abaixo:

```
# mpg123 -a /dev/dsp1.0 Foobar-GreatestHits.mp3
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layer 1, 2 and 3.
Version 0.59r (1999/Jun/15). Written and copyrights by Michael Hipp.
Uses code from various people. See 'README' for more!
THIS SOFTWARE COMES WITH ABSOLUTELY NO WARRANTY! USE AT YOUR OWN RISK!
```

```
Playing MPEG stream from Foobar-GreatestHits.mp3 ...
MPEG 1.0 layer III, 128 kbit/s, 44100 Hz joint-stereo
```

O `/dev/dsp1.0` deve ser substituído com o dispositivo `dsp` correspondente ao do seu sistema.

16.3.2. Ripando trilhas de um CD de áudio

Antes de encodar um CD ou uma trilha de CD para MP3, os dados de áudio do CD devem ser ripados para o seu disco rígido. Isto é feito através da cópia dos dados CDDA brutos (CD Digital Audio) para arquivos `.WAV`.

A ferramenta `cdda2wav` a qual faz parte do pacote [sysutils/cdrtools](#), é utilizada para ripar informações de áudio dos CDs, bem como as informações associadas a ele.

Com o CD de áudio no drive, o seguinte comando pode ser executado (como `root`) para ripar um CD inteiro para arquivos `.WAV` individuais (um por trilha):

```
# cdda2wav -D 0,1,0 -B
```

O `cdda2wav` também suporta drives de CDROM padrão ATAPI (IDE). Para ripar a partir de um drive IDE, especifique o nome do dispositivo no lugar do número de unidade SCSI. Por exemplo, para ripar a trilha 7 a partir de um drive IDE:

```
# cdda2wav -D /dev/acd0a -t 7
```

A opção `-D 0,1,0` indica o dispositivo SCSI `0,1,0`, o qual corresponde a saída do comando `cdrecord -scanbus`.

Para ripar trilhas individuais, faça uso da opção `-t` como mostrado:

```
# cdda2wav -D 0,1,0 -t 7
```

Este exemplo ripa a trilha 7 do CDROM de áudio. Para ripar um range de trilhas, por exemplo, da trilha 1 a 7, especifique o range:

```
# cdda2wav -D 0,1,0 -t 1+7
```

O utilitário [dd\(1\)](#) também pode ser utilizado para extrair uma trilha de áudio de um drive ATAPI, leia a [Seção 12.5.5](#) para maiores informações sobre esta possibilidade.

16.3.3. Codificando arquivos MP3

Hoje em dia, o codificador mp3 escolhido é o **lame**. O **Lame** pode ser encontrado no caminho [audio/lame](#) da árvore do ports.

Utilizando os arquivos WAV ripados, o comando seguinte irá converter o arquivo `audio01.wav` para `audio01.mp3`:

```
# lame -h -b 128 \
--tt "Foo Song Title" \
--ta "FooBar Artist" \
--tl "FooBar Album" \
--ty "2001" \
--tc "Ripped and encoded by Foo" \
--tg "Genre" \
audio01.wav audio01.mp3
```

Aparentemente 128 kbits é o bitrate padrão para uso com arquivos MP3. Muitos apreciam a qualidade superior oferecida por bitrates maiores como 160 ou 192kbps. Quanto maior o bitrate, maior o espaço em disco ocupado pelo arquivo mp3 resultante - mas a qualidade também será maior. A opção *-h* ativa o modo "Maior qualidade mas um pouco mais lento". As opções iniciadas com *-t* indicam as tags ID3, as quais contém usualmente informações da música, para serem embutidas no arquivo MP3. Opções adicionais de codificação podem ser encontradas consultando a página de manual do lame.

16.3.4. Decodificando arquivos MP3

A fim de gravar um CD de audio a partir de arquivos MP3, os mesmos devem ser convertidos para um formato WAV não-comprimido. Tanto o **XMMS** quanto o **mpg123** suportam redirecionar a saída de um MP3 para um formato de arquivo descomprimido.

Gravando para disco com o **XMMS**:

1. Execute o **XMMS**.
2. Clique com o botão direito do mouse na janela para exibir o menu do **XMMS**.
3. Selecione a opção *Preference* abaixo do *Options*.
4. Altere o plugin de saída para "Disk Writer Plugin".
5. Pressione *Configure*.
6. Entre com o diretório (ou escolha a opção browse) no qual os arquivos descomprimidos serão gravados.
7. Carregue normalmente o arquivo MP3 no **XMMS**, com o volume ajustado para 100% e com as configurações do equalizador desligadas.
8. Pressione *Play* -- O **XMMS** irá se comportar como se estivesse reproduzindo o arquivo MP3, mas nenhuma música será ouvida. Ele está na verdade reproduzindo o arquivo MP3 para um arquivo.
9. Certifique-se de refazer a configuração do plugin de saída, para poder voltar a ouvir seus MP3 de novo.

Escrevendo para a saída padrão no **mpg123**:

1. Execute `mpg123 -s audio01.mp3 > audio01.pcm`

O **XMMS** gera um arquivo no formato WAV, enquanto o **mpg123** converte o MP3 em dados brutos de audio no formato PCM. Ambos os formatos podem ser utilizados com o **cdrecord** para criar CDs de audio. Já com o [burncd\(8\)](#), você poderá utilizar somente arquivos de audio em formato PCM. Se você utilizar arquivos WAV, você observará um pequeno ruído no início de cada trilha, este som é o cabeçalho do arquivo WAV.

Você pode simplesmente remover o cabeçalho de um arquivo WAV com o utilitário **SoX** (ele pode ser instalado a partir do port [audio/sox](#) ou de um pacote binário):

```
% sox -t wav -r 44100 -s -w -c 2 track.wav track.raw
```

Leia a sessão [Seção 12.5](#) para maiores informações sobre o uso de um gravador de CDs no FreeBSD.

16.4. Reprodução de Vídeo

Contribuído por Ross Lippert.

A reprodução de vídeo é uma área de aplicação muito nova e de rápido desenvolvimento. Seja paciente. Nem tudo está funcionando tão suavemente como com o som.

Antes de você iniciar, você deve saber qual o modelo de placa de vídeo que você possui e qual o chipset que ela utiliza. Apesar do **XFree86**TM suportar uma ampla variedade de placas de vídeo, poucas possuem uma boa performance de reprodução. Para obter a lista de extensões suportadas pelo servidor X que você está utilizando, utilize o comando [xdpyinfo\(1\)](#) enquanto o X11 estiver sendo executado.

É uma ideia possuir um arquivo MPEG curto, o qual possa ser tratado como um arquivo de teste para avaliar os diversos reprodutores e suas opções. Uma vez que alguns reprodutores de DVD irão procurar por padrão uma mídia DVD no `/dev/dvd`, ou então possuem este nome dispositivo definido estaticamente no código do programa, você pode achar útil criar links simbólicos apontando para os dispositivos apropriados:

```
# ln -sf /dev/acd0c /dev/dvd
# ln -sf /dev/racd0c /dev/rdvd
```

No FreeBSD 5.X, o qual utiliza [devfs\(5\)](#) existe um conjunto ligeiramente diferente de links recomendados:

```
# ln -sf /dev/acd0c /dev/dvd
# ln -sf /dev/acd0c /dev/rdvd
```

Adicionalmente, a decodificação de um DVD, a qual necessita da execução de funções especiais do DVD-ROM, exige permissão de escrita no dispositivo de DVD.

Alguns dos ports discutidos contam com as seguintes opções no kernel para compilarem corretamente. Antes de tentar compilá-las, adicione estas opções ao arquivo de configuração do kernel, compile um novo kernel, e reinicialize:

```
option CPU_ENABLE_SSE
option USER_LDT
```

Para aumentar a memória compartilhada disponível para a interface X11, é recomendado que os valores de algumas variáveis do [sysctl\(8\)](#) sejam aumentadas:

```
kern.ipc.shmmax=67108864
kern.ipc.shmall=32768
```

16.4.1. Determinando as Capacidades de Video

Há diversas maneiras possíveis de se exibir um vídeo sob X11. Qual realmente irá funcionar depende largamente do hardware utilizado. Cada um dos métodos descritos abaixo terá sua qualidade variando em função dos diferentes hardwares. Em segundo lugar, a renderização de video em ambiente X11 é um tópico que recebe muita atenção ultimamente, e cada nova versão do **XFree86** pode trazer melhorias significativas.

Uma lista das interfaces de video mais comuns:

1. X11: Saída normal do X11 utilizando memória compartilhada.
2. XVideo: Uma extensão para a interface X11 a qual suporta video em qualquer X11 drawable.
3. SDL: Camada simples Directmedia.
4. DGA: Acesso gráfico direto.
5. SVGAlib: Camada gráfica para console.

16.4.1.1. XVideo

O **XFree86 4.X** possui uma extensão chamada *XVideo* (também conhecida como *Xvideo*, *Xv*, ou simplesmente *xv*) a qual permite que o video seja exibido diretamente em objetos drawable através de uma aceleração especial. Esta extensão prova uma qualidade de reprodução muito boa, mesmo em equipamentos de poucos recursos (por exemplo um um laptop PIII 400 Mhz). Infelizmente, a lista de placas nas quais esta característica é atualmente suportada de forma nativa se limita a:

1. 3DFX Voodoo 3
2. Intel® i810 e i815
3. Alguns chips S3 (tais como Savage/IX e Savage/MX)

Se a sua placa não é uma destas, não fique desapontado ainda. O **XFree86 4.X** adiciona novas habilidades *xv* a cada release [\[1\]](#). Para verificar se a extensão esta executando, utilize o comando `xvinfo`:

```
% xvinfo
```

XVideo is supported for your card if the result looks like:

```
X-Video Extension version 2.2
screen #0
  Adaptor #0: "Savage Streams Engine"
    number of ports: 1
    port base: 43
    operations supported: PutImage
    supported visuals:
      depth 16, visualID 0x22
```

```
depth 16, visualID 0x23
number of attributes: 5
  "XV_COLORKEY" (range 0 to 16777215)
    client settable attribute
    client gettable attribute (current value is 2110)
  "XV_BRIGHTNESS" (range -128 to 127)
    client settable attribute
    client gettable attribute (current value is 0)
  "XV_CONTRAST" (range 0 to 255)
    client settable attribute
    client gettable attribute (current value is 128)
  "XV_SATURATION" (range 0 to 255)
    client settable attribute
    client gettable attribute (current value is 128)
  "XV_HUE" (range -180 to 180)
    client settable attribute
    client gettable attribute (current value is 0)
maximum XvImage size: 1024 x 1024
Number of image formats: 7
  id: 0x32595559 (YUY2)
    guid: 59555932-0000-0010-8000-00aa00389b71
    bits per pixel: 16
    number of planes: 1
    type: YUV (packed)
  id: 0x32315659 (YV12)
    guid: 59563132-0000-0010-8000-00aa00389b71
    bits per pixel: 12
    number of planes: 3
    type: YUV (planar)
  id: 0x30323449 (I420)
    guid: 49343230-0000-0010-8000-00aa00389b71
    bits per pixel: 12
    number of planes: 3
    type: YUV (planar)
  id: 0x36315652 (RV16)
    guid: 52563135-0000-0000-0000-000000000000
    bits per pixel: 16
    number of planes: 1
    type: RGB (packed)
    depth: 0
    red, green, blue masks: 0x1f, 0x3e0, 0x7c00
  id: 0x35315652 (RV15)
    guid: 52563136-0000-0000-0000-000000000000
    bits per pixel: 16
    number of planes: 1
    type: RGB (packed)
    depth: 0
    red, green, blue masks: 0x1f, 0x7e0, 0xf800
  id: 0x31313259 (Y211)
    guid: 59323131-0000-0010-8000-00aa00389b71
    bits per pixel: 6
    number of planes: 3
    type: YUV (packed)
  id: 0x0
    guid: 00000000-0000-0000-0000-000000000000
    bits per pixel: 0
    number of planes: 0
    type: RGB (packed)
    depth: 1
    red, green, blue masks: 0x0, 0x0, 0x0
```

Também observe que os formatos listados (YUV2, YUV12, etc) não estão presentes em todas as implementações do XVideo, e a sua ausência pode atrapalhar alguns reprodutores.

se o resultado se paracer com:

```
X-Video Extension version 2.2
screen #0
no adaptors present
```

Então provavelmente o XVideo não é suportado na sua placa.

Se o XVideo não é suportado para a sua placa, isto significa somente que será mais difícil para o seu monitor lidar com a demanda computacional envolvida no processo de renderização de vídeo. Dependendo da sua placa de vídeo e do seu processador, você pode continuar sendo capaz de ter uma experiência satisfatória. Recomenda-se que você consulte a sessão [Seção 16.4.3](#) para obter maiores informações sobre como melhorar a sua performance.

16.4.1.2. Camada simples directmedia (Simple Directmedia Layer)

O SDL (Simple Directmedia Layer), está tentando se tornar uma camada portátil entre Microsoft® Windows®, BeOS, e UNIX®, permitindo o desenvolvimento de aplicativos cross-platform as quais fariam uso eficiente do som e dos gráficos. A camada SDL disponibiliza uma abstração de baixo nível para o hardware o que em algumas vezes pode ser mais eficiente do que a interface X11.

O SDL pode ser encontrado em [devel/sdl12](#)

16.4.1.3. Acesso gráfico direto

O acesso gráfico direto é uma extensão do **XFree86** a qual permite a um programa contornar o servidor X e alterar diretamente o framebuffer da placa de vídeo. Como ele necessita de acesso ao mapeamento de memória em baixo nível para que este compartilhamento seja possível, os programas que o utilizam devem ser executados como `root`.

A extensão DGA pode ser testada e avaliada pelo [dga\(1\)](#). Quando o `dga` está executando, ele altera as cores do monitor sempre que uma tecla é pressionada. Para sair, utilize a tecla **q**.

16.4.2. Ports e Pacotes para lidar com vídeo

Esta sessão trata dos programas disponíveis na coleção de ports do FreeBSD, os quais podem ser utilizados para reproduzir vídeo. A reprodução de vídeo é uma área de desenvolvimento de software muito ativa, e as características de várias aplicações podem divergir um pouco das descrições fornecidas aqui.

Inicialmente, é importante que você saiba que muitas das aplicações de vídeo que rodam no FreeBSD foram desenvolvidas como uma aplicação linux. Muitas destas aplicações

permanecem com o status de versões beta (experimentais). Alguns dos problemas que você pode encontrar com os pacotes de vídeo no FreeBSD incluem:

1. Um aplicativo não pode reproduzir um arquivo o qual tenha sido produzido por outro aplicativo.
2. Um aplicativo não pode reproduzir um arquivo o qual foi produzido pelo próprio aplicativo.
3. O mesmo aplicativo em duas máquinas diferentes, reconstruído em cada máquina para aquela máquina específica, reproduz o mesmo arquivo de forma diferente.
4. O uso de filtro aparentemente trivial tal como reescalar o tamanho de uma imagem, gera um resultado muito ruim devido a um bug na rotina de reescalonamento.
5. Um aplicativo freqüentemente apresenta um coredump.
6. A documentação não é instalada com o port e pode ser encontrada tanto na web quanto sob o diretório `work` dentro do diretório do port.

Muitas destas aplicações também pode exibir "Linux-isms". Isto é, podem existir algumas edições resultantes da forma como algumas das bibliotecas padrões estão implementadas nas distribuições Linux, ou em algumas características do kernel do Linux a qual tenha sido assumida pelos autores da aplicação. Estas edições nem sempre são reparadas ou trabalhadas pelo mantenedor do port, o que pode conduzir a problemas tais como pode:

1. O uso do `/proc/cpuinfo` para detectar as características do processador.
2. O uso inapropriado de threads o que causa a suspensão do programa ao invés de verdadeiramente encerra-lo.
3. O software não esta ainda na coleção de ports do FreeBSD o qual é utilizado geralmente em conjunto com a aplicação.

Até agora, estes desenvolvedores de aplicativos tem sido cooperativos com os mantenedores de ports para minimizar as necessidades de work-arounds no port da aplicação.

16.4.2.1. MPlayer

O **MPlayer** é um reprodutor de vídeo desenvolvido recentemente, o qual vem sendo desenvolvido rapidamente. Os objetivos do time de desenvolvedores do **MPlayer** são a velocidade e a flexibilidade no linux e em outros sabores do unix. O projeto foi iniciado quando o fundador do projeto se encheu da má performance de reprodução de vídeo dos reprodutores que estavam disponíveis. Alguns diriam que a interface gráfica foi sacrificada em favor de um design simplificado. Entretanto, uma vez que você tenha começado a utilizar as opções de linha de comando e os controles de teclado, ele funciona muito bem.

16.4.2.1.1. Compilando o MPlayer

O **MPlayer** se localiza no [multimedia/mplayer](#). O **MPlayer** realiza diversas verificações de hardware durante o processo de compilação, resultando em um binário o qual não é portátil de um sistema para outro. Por este motivo é importante compilá-lo a

partir do ports e evitando utilizar um pacote binário. Adicionalmente, um grande número de opções podem ser especificadas na linha de comando do `make`, como descrito no início do processo de compilação.

```
# cd /usr/ports/multimedia/mplayer
# make
You can enable additional compilation optimizations
by defining WITH_OPTIMIZED_CFLAGS
You can enable GTK GUI by defining WITH_GUI.
You can enable DVD support by defining WITH_DVD.
You can enable SVGALIB support by defining WITH_SVGALIB.
You can enable VORBIS sound support by defining WITH_VORBIS.
You can enable XAnim DLL support by defining WITH_XANIM.
```

Se você possui o [x11-toolkits/gtk12](#) instalado, então você também poderá habilitar a interface gráfica. Se não, não vale o esforço. Se você planeja reproduzir (possivelmente encodados com CSS) DVD's com o **MPlayer** você precisa habilitar a opção de suporte a DVD [2]. Algumas opções razoáveis são:

```
# make WITH_DVD=yes WITH_SVGALIB=yes
```

Neste momento em que este capítulo está sendo escrito, o port do **MPlayer** irá construir a documentação HTML dele e um executável, `mplayer`. Ele também pode ser compilado de forma a gerar um encoder, `mencoder`, o qual é uma ferramenta para re-encodar video. Uma modificação no `Makefile` pode habilitar a sua geração. Ele pode ser habilitado por padrão nas versões futuras do port.

A documentação HTML do **MPlayer** é muito informativa. Se o leitor sentir falta de alguma informação sobre hardware e interface de video, a documentação do **MPlayer** é um suplemento perfeito. Você definitivamente deveria dedicar um tempo a leitura da documentação do **MPlayer** se você esta procurando por informações sobre o suporte a video no ambiente UNIX.

16.4.2.1.2. Utilizando o MPlayer

Qualquer usuário do **MPlayer** deve estabelecer um subdiretório `.mplayer` no seu diretório home. Para criar este subdiretório necessário ao funcionamento do aplicativo, você pode executar o seguinte:

```
% cd /usr/ports/multimedia/mplayer
% make install-user
```

As opções de comando para o `mplayer` estão listadas na sua página de manual. Para maiores detalhes consulte a documentação HTML. Nesta sessão, nós descreveremos apenas alguns usos comuns.

Para reproduzir um arquivo, como por exemplo `testfile.avi`, através de uma das várias interfaces de video, especifique a opção `-vo`:

```
% mplayer -vo xv testfile.avi
% mplayer -vo sdl testfile.avi
% mplayer -vo x11 testfile.avi
# mplayer -vo dga testfile.avi
```

```
# mplayer -vo 'sdl:dga' testfile.avi
```

Vale a pena tentar todas estas opções, porque seu desempenho relativo depende de muitos fatores e variará significativamente com o hardware.

Para reproduzir um DVD, substitua o `testfile.avi` com `-dvd <N> DEVICE` onde `<N>` é o número do título a ser reproduzido e `DEVICE` é o nó de dispositivo para o DVD-ROM. Por exemplo, para reproduzir o título 3 a partir do `/dev/dvd`:

```
# mplayer -vo dga -dvd 2 /dev/dvd
```

Para parar, pausar, avançar e assim por diante, consulte o mapeamento de teclas, o qual é exibido pela execução do comando `mplayer -h` ou então leia a página de manual.

Uma importante opção de reprodução é: `-fs -zoom` a qual emprega o modo de tela cheia e `-framedrop` a qual ajuda na performance.

Para que a linha de comando do `mplayer` não se torne demasiadamente grande, o usuário pode criar um arquivo `.mplayer/config` e especificar ali as opções padrões:

```
vo=xv
fs=yes
zoom=yes
```

E por último, o `mplayer` pode ser utilizado para ripar um título DVD para um arquivo `.vob`. Para ripar o segundo título de um DVD, execute isto:

```
# mplayer -dumpstream -dumpfile out.vob -dvd 2 /dev/dvd
```

O arquivo de saída, `out.vob`, será um arquivo mpeg e pode ser manipulado pelos outros pacotes descritos nesta sessão.

16.4.2.1.3. *mencoder*

Se você optou por instalar o `mencoder` quando compilou o **MPlayer**, fique avisado de antemão que ele ainda é um componente experimental. Antes de utilizar o `mencoder` é uma boa idéia se familiarizar primeiro com as suas opções a partir da leitura da documentação HTML. Existe uma página de manual, mas ela não é muito útil sem a documentação HTML. Existem inúmeras maneiras de melhorar a qualidade, diminuir a taxa de dados (bitrate), e alterar os formatos, e alguns destes truques podem fazer a diferença entre um desempenho bom e um medíocre. Aqui está um par de exemplos para iniciarmos. Primeiro uma cópia simples:

```
% mencoder input.avi -oac copy -ovc copy -o output.avi
```

Combinações inapropriadas de opções da linha de comando podem render arquivos de saída que não podem ser reproduzidos com o `mplayer`. Desta forma, se você deseja apenas ripar para um arquivo, utilize a opção `-dumpfile` no `mplayer`.

Para converter o arquivo `input.avi` para o codec de vídeo MPEG4 e com codificação de áudio no formato MPEG3 (é necessário ter o [audio/lame](#)):

```
% mencoder input.avi -oac mp3lame -lameopts br=192 \  
-ovc lavc -lavcopts vcodec=mpeg4:vhq -o output.avi
```

Isso produz um arquivo de saída que pode ser reproduzido pelo `mplayer` e pelo `xine`.

O `input.avi` pode ser substituído por `-dvd 1 /dev/dvd` e execute como `root` para re-encodar diretamente um título DVD. Uma vez que é provável que você não fique satisfeito com o resultado na sua primeira tentativa, é recomendado que você ripe o título para um arquivo e trabalhe no mesmo.

16.4.2.2. Reprodutor de video `xine`

O reprodutor de video `xine` é um projeto com objetivos amplos, tentando produzir não apenas uma solução de video tudo em uma, mas também uma biblioteca de apoio reutilizável, e um executável modular cuja funcionalidade possa ser estendido com o uso de plugins. Ele pode ser instalado tanto por um pacote quanto pelo [ports multimedia/xine](#).

O `xine` está ainda em uma fase rudimentar e avança lentamente, mais ele é claramente um bom começo. Na prática, o `xine` necessita de uma CPU rápida com uma placa de video rápida, ou então de uma placa com suporte para a extensão XVideo. A interface gráfica é utilizável, mas um pouco desajeitada.

No momento em que esta sessão está sendo escrita, não existe nenhum módulo sendo distribuído com o `xine` o qual seja capaz de reproduzir DVD's codificados com CSS. Existem configurações de terceiros as quais implementam módulos para isto, mas nenhum deles está presente na coleção de ports do FreeBSD.

Comparado com o `MPlayer`, o `xine` faz mais pelo usuário, mas ao mesmo tempo, priva o usuário do controle de ajuste fino. O `xine` apresenta a melhor performance quando sob interface XVideo.

Por padrão, ao ser executado o `xine` irá carregar a sua interface gráfica. Os menus podem ser utilizados para abrir um arquivo específico:

```
% xine
```

Alternativamente, ele pode ser executado de forma a reproduzir imediatamente um arquivo, sem exibir a interface gráfica ao usuário, com o comando:

```
% xine -g -p mymovie.avi
```

16.4.2.3. Utilitário `transcode`

O software `transcode` não é um reprodutor, mas um conjunto de ferramentas para re-encodar arquivos `.avi` e `.mpg`. Com o `transcode`, tem-se a habilidade de juntar arquivos de video, consertar arquivos quebrados, utilizando ferramentas de linha de comando com interface de fluxo de dados para `stdin/stdout`.

Assim como o **MPlayer**, o **transcode** é um software muito experimental, o qual deve ser compilado a partir do [multimedia/transcode](#). Existe uma grande variedade de opções que podem ser utilizadas com o comando `make`. As recomendadas são:

```
# make WITH_LIBMPEG2=yes
```

Se você planeja instalar o [multimedia/avifile](#), então adicione a opção `WITH_AVIFILE` a sua linha de comando `make`, como mostrado aqui:

```
# make WITH_AVIFILE=yes WITH_LIBMPEG2=yes
```

Aqui estão 2 exemplos de uso do `transcode` para para converter um video alterando o tamanho do mesmo na saída. O primeiro encoda a saída para um arquivo AVI `openDIVX`, enquanto o segundo encoda para o formato MPEGi, o qual é muito mais portátil.

```
% transcode -i input.vob -x vob -V -Z 320x240 \  
-y opendivx -N 0x55 -o output.avi  
% transcode -i input.vob -x vob -V -Z 320x240 \  
-y mpeg -N 0x55 -o output.tmp  
% tcplex -o output.mpg -i output.tmp.mlv -p output.tmp.mpa -m 1
```

Existe uma página de manual para o `transcode`, mas existe muito pouca documentação para os vários utilitários `tc*` (como por exemplo o `tcplex`) os quais também estão instalados. Entretanto, a opção de linha de comando `-h` sempre pode ser utilizada para obter uma curta instrução de uso para um comando.

Em comparação, o `transcode` roda significativamente mais devagar do que o `mencoder`, mas ele possui uma chance melhor de produzir um arquivo mais amplamente reprodutível. MPEGs criados pelo `transcode` sabidamente podem ser reproduzidos nas versões antigas do **Windows Media® Player** e do **Quicktime®** da Apple, por exemplo.

16.4.3. Referências para leitura adicional

Os vários pacotes de software de video para FreeBSD são desenvolvidos rapidamente. E é bem provável que em um futuro próximo muitos dos problemas discutidos aqui terão sido resolvidos. Neste meio tempo, aqueles que desejarem tirar o máximo das funcionalidades de A/V do FreeBSD terão que complementar seu conhecimento a partir de diversos FAQs e tutoriais e utilizar alguns aplicativos diferentes. Esta sessão existe para dar ao leitor indicações de onde obter estas informações adicionais.

A [documentação do MPlayer](#) é muito informativa tecnicamente. Esta documentação deveria ser consultada por qualquer pessoa que deseja obter um alto nível de especialização em video no ambiente UNIX. A lista de discussão do **MPlayer** é hostil para qualquer pessoa que não tenha lido a documentação, assim se você planeja enviar relatórios de problemas para ela, leia o manual (RTFM).

O [tutorial sobre xine](#) contém um capítulo sobre melhoria de performance a qual se aplica a todos os reprodutores.

Finalmente, existem algumas outras aplicações promissoras as quais o leitor pode testar:

- O [Avifile](#) também está disponível no port [multimedia/avifile](#).
- O [Ogle](#) também está disponível no port [multimedia/ogle](#).
- [Xtheater](#)
- [multimedia/dvdauthor](#), um pacote de código aberto para autoria de conteúdo de DVDs

Notas

- [1] Uma placa gráfica popular e com uma performance no **XFree86** geralmente muito boa, as placas nVidia, ainda não liberaram as especificações do seu suporte a XVideo para o grupo de desenvolvimento do **XFree86**. De forma que ainda pode levar algum tempo até que o **XFree86** suporte completamente o XVideo nestas placas.
- [2] Reprodução não autorizada de DVD's é um serio ato criminal em alguns países. Verifique as leis locais antes de habilitar esta opção.
-

Capítulo 17. Comunicações Seriais

Índice

- 17.1. [Sinópse](#)
- 17.2. [Introdução](#)
- 17.3. [Terminais](#)
- 17.4. [Serviço de Dial-in](#)
- 17.5. [Serviço de Dial-out](#)
- 17.6. [Configurando uma console serial](#)

17.1. Sinópse

O UNIX® sempre possuiu suporte a comunicações seriais. De fato, as primeiras máquinas UNIX disponibilizavam linhas seriais para as operações de entrada e de saída realizada pelos usuários. As coisas mudaram muito desde os dias em que a média dos ``terminais" consistiam-se em uma impressora serial de 10 caracteres por segundo e um teclado. Este capítulo irá abordar algumas das formas como o FreeBSD utiliza comunicações seriais.

Após ler este capítulo você irá conhecer:

- Como conectar terminais ao seu sistema FreeBSD .
- Como utilizar um modem para discar para um servidor remoto .
- Como permitir que usuários remotos se conectem ao seu sistema através de um modem.
- Como efetuar boot utilizando um console serial.

Antes de ler este capítulo, você deve:

- Conhecer como configurar e instalar um novo kernel ([Capítulo 9](#)).

- Compreender o sistema de permissões e de processos no ambiente UNIX ([Capítulo 3](#)).
 - Ter acesso ao manual técnico do hardware serial (Modem ou Placa multi-portas) que você deseja utilizar com o FreeBSD.
-

17.2. Introdução

17.2.1. Terminologia

bps

Bits por segundo -- é a taxa com que os dados são transmitidos.

DTE

Equipamento Terminal de Dados -- por exemplo, o seu computador

DCE

Equipamento de Comunicação de Dados -- seu modem

RS-232

Padrão EIA para hardwares de comunicação serial

Quando fala sobre taxa de comunicação de dados, esta sessão não usa o termo ``baud". Baud se refere ao número de estados de transição elétricas que ocorrem em um determinado periodo de tempo, enquanto ``bps" (bits por segundo) é o termo *correto* a ser utilizado (ao menos ele não parece incomodar completamente os teimosos).

17.2.2. Cabos e Portas

Para conectar um modem ou um terminal ao seu sistema FreeBSD, você irá precisar de uma porta serial em seu computador e do cabo apropriado para conectar o seu dispositivo serial. Se você já esta familiarizado com o seu dispositivo e com o cabo que ele necessita, você pode pular tranquilamente esta sessão. .

17.2.2.1. Cabos

Há diversos tipos diferentes de cabos seriais. Os dois tipos mais comuns para os nossos propósitos são os cabos *null-modem* e os cabos padrão RS-232. A documentação do seu dispositivo irá descrever o tipo de cabo necessário.

17.2.2.1.1. Cabos Null-Modem

Um cabo *null-modem* passa alguns sinais, como o ``sinal terra", de forma direta (sai e chega no mesmo pino em ambas as pontas), mas chaveia outros sinais. Por exemplo, o

pino de "envio de dados" em uma das pontas do cabo irá para o pino "recebimento de dados" na outra ponta.

Se você preferir preparar seus próprios cabos, você pode construir um cabo null-modem para utilizar com terminais. Esta tabela mostra os nomes dos sinais RS-232C e a numeração dos pinos em um conector DB-25.

Sinal	Pino #		Pino #	Sinal
SG	7	conecta-se ao	7	SG
TxD	2	conecta-se ao	3	RxD
RxD	3	conecta-se ao	2	TxD
RTS	4	conecta-se ao	5	CTS
CTS	5	conecta-se ao	4	RTS
DTR	20	conecta-se ao	6	DSR
DCD	8		6	DSR
DSR	6	conecta-se ao	20	DTR

Nota: Conecte o "Data Set Ready" (DSR) e o "Data Carrier Detect" (DCD) internamente no conector, e então conecte-os ao "Data Terminal Ready" (DTR) no conector remoto.

17.2.2.1.2. Cabos Padrão RS-232C

Um cabo serial padrão trafega todos os sinais de forma direta (sai e chega no mesmo pino em ambas as pontas). Isto é, o pino de "envio de dados" em uma ponta do cabo, irá para o pino "envio de dados" na outra ponta. Este é o tipo de cabo utilizado para conectar um modem ao seu sistema FreeBSD, e também é adequado para alguns terminais.

17.2.2.2. Portas

Portas seriais são os dispositivos através dos quais os dados são transferidos entre um host FreeBSD e um terminal. Esta sessão descreve os tipos de portas que existem e como elas são reconhecidas no FreeBSD.

17.2.2.2.1. Tipos de portas

Existem diversos tipos de portas seriais. Antes de comprar ou construir um cabo, você precisa ter certeza que ele irá se ajustar a porta existente no seu terminal e no seu sistema FreeBSD.

A maioria dos terminais irão ter portas DB25. Computadores pessoais, incluindo PCs rodando FreeBSD, terão portas DB25 ou portas DB9. Se você possui uma placa multiseriada para o seu PC, você poderá ter portas RJ-12 ou portas RJ-45.

Consulte a documentação que acompanha o seu hardware para obter a especificação do tipo de porta que ele utiliza. Uma inspeção visual da porta costuma ajudar também.

17.2.2.2. Nomes das Portas

No FreeBSD, você acessa cada porta serial através de uma entrada no diretório `/dev`. Existem 2 tipos diferentes de entradas:

- As portas utilizadas para receber ligações (*call-in*) são chamadas `/dev/ttydN`, onde *N* é o número da porta, iniciando-se em 0. Geralmente, você utiliza as portas *call-in* para terminais. As portas *call-in* necessitam que a linha serial estabeleça o sinal de detecção de portadora de dados (DCD) para funcionar corretamente.
- As portas utilizadas para efetuar ligações (*call-out*), são chamadas `/dev/cuaaN`. Você normalmente não utiliza este tipo de porta para terminais, usando-as apenas para modems. Você pode utilizar uma porta *call-out* sempre que o cabo serial ou o terminal não suportar o sinal de detecção da portadora.

Se você conectou um terminal a sua primeira porta serial (COM1 no MS-DOS®), você terá que utilizar `/dev/ttyd0` para acessar o terminal. Se o terminal estiver conectado a segunda porta serial (também conhecida como COM2), utilize `/dev/ttyd1`, e assim por diante.

17.2.3. Configuração do Kernel

O FreeBSD por default suporta 4 portas seriais. No mundo MS-DOS, elas são conhecidas como COM1, COM2, COM3, e COM4. O FreeBSD atualmente suporta diversas placas multiseriais, tais como a BocaBoard 1008 e 2016, bem como outras placas multiseriais inteligentes fabricadas com as tecnologias Digiboard e Stallion. No entanto, o kernel default suporta apenas as portas COM normais.

Para verificar se o seu kernel reconhece alguma das suas portas seriais, olhe as mensagens exibidas no console enquanto o kernel está sendo carregado, ou então utilize o comando `/sbin/dmesg` para visualizar novamente as mensagens exibidas durante o processo de boot do kernel. Em particular, procure por mensagens que iniciem com os caracteres *sio*.

Dica: Para visualizar apenas as mensagens que contenham a palavra *sio*, use o comando:

```
# /sbin/dmesg |  
grep 'sio'
```

Por exemplo, em um sistema com 4 portas seriais, as mensagens específicas do kernel para as portas seriais durante o processo de boot seriam parecidas com essas:


```
sio0 at 0x3f8-0x3ff irq 4 on isa
sio0: type 16550A
sio1 at 0x2f8-0x2ff irq 3 on isa
sio1: type 16550A
sio2 at 0x3e8-0x3ef irq 5 on isa
sio2: type 16550A
sio3 at 0x2e8-0x2ef irq 9 on isa
sio3: type 16550A
```

Se o seu kernel não reconheceu todas as suas portas seriais, você provavelmente precisa de um kernel customizado para o seu sistema FreeBSD. Para informações detalhadas de como configurar o seu kernel, por favor consulte [Capítulo 9](#).

As linhas de device relevantes em seu arquivo de configuração do kernel, irão parecer com estas, para o FreeBSD 4.X:

```
device      sio0      at isa? port IO_COM1 irq 4
device      sio1      at isa? port IO_COM2 irq 3
device      sio2      at isa? port IO_COM3 irq 5
device      sio3      at isa? port IO_COM4 irq 9
```

e parecida com esta, para o FreeBSD 5.X:

```
device      sio
```

No FreeBSD 4.X; você pode comentar ou remover por completo as linhas dos dispositivos que você não possui, no caso do FreeBSD 5.X você tem que editar o seu arquivo `/boot/device.hints` para configurar as suas portas seriais. Por favor consulte a página de manual do [sio\(4\)](#) para maiores informações sobre a configuração de portas seriais e de placas multiseriais. Tenha cuidado se você estiver utilizando um arquivo de configuração usado previamente em uma versão diferente do FreeBSD porque os *flags* de dispositivo e a sintaxe mudou entre as versões.

Nota: *port IO_COM1* é a substituição para *port 0x3f8*, *IO_COM2* é *0x2f8*, *IO_COM3* é *0x3e8*, e *IO_COM4* é *0x2e8*, os quais são os endereços mais comuns utilizados para as respectivas portas seriais. As interrupções 4, 3, 5, e 9 são as linhas de interrupção normalmente solicitadas. Observe também que as portas seriais comuns *não podem* compartilhar IRQ no barramento ISA (as placas multiseriais possuem componentes eletrônicos onboard que permitem que todos os chips 16550A da placa compartilhem 1 ou 2 IRQs).

17.2.4. Arquivos especiais de dispositivo

A maioria dos dispositivos no kernel são acessados através de "arquivos especiais de dispositivos", que estão localizados no diretório `/dev`. Os dispositivos `sio` são acessados através dos dispositivos `/dev/ttydN` (dial-in) e `/dev/cuaaN` (call-out). O FreeBSD também disponibiliza dispositivos de inicialização (`/dev/ttyidN` e `/dev/cuaiaN`) e dispositivos de "fechamento" (`/dev/ttyldN` e `/dev/cualaN`). Os dispositivos de inicialização são utilizados para inicializar os parametros das portas de comunicação cada vez que elas são abertas, como por exemplo o `crtscts` para os modems que utilizam a sinalização *RTS/CTS* para controle de fluxo. Os dispositivos de fechamento são utilizados para travar flags nas portas para prevenir que usuários ou

programas mudem certos parametros; veja as páginas de manual do [termios\(4\)](#), [sio\(4\)](#), and [stty\(1\)](#) para informações sobre a configuração de terminais, fechamento e inicialização de dispositivos, e configuração das opções de terminal, respectivamente.

17.2.4.1. Criando arquivos especiais de dispositivo

Nota: O FreeBSD 5.0 inclui o sistema de arquivos [devfs\(5\)](#), o qual cria automaticamente os *device nodes* a medida que se fazem necessários. Se você esta executando esta versão do FreeBSD com o *devfs* habilitado, você pode pular tranquilamente esta sessão.

Um script shell chamado MAKEDEV existente no diretório /dev gerência os arquivos especiais de dispositivo. Para utilizar o MAKEDEV para criar os arquivos especiais de dispositivos de dial-up para a porta COM1 (porta 0), vá para o diretório /dev e execute o comando MAKEDEV ttyd0. Da mesma forma, para criar os arquivos especiais de dispositivos dial-up para a porta COM2 (porta 1), utilize o comando MAKEDEV ttyd1.

O MAKEDEV não cria apenas os arquivos especiais de dispositivo /dev/ttydN, mas também os /dev/cuaaN, /dev/cuaiaN, /dev/cualaN, /dev/ttyldN, e /dev/ttyidN.

Depois de criar os novos arquivos especiais de dispositivo, certifique-se sobre as permissões dos arquivos (especialmente os arquivos /dev/cua*), para ter certeza de que apenas os usuários que precisam ter acesso a eles, possuem permissão de escrita e de leitura nos mesmos -- você provavelmente não deseja permitir que os seus usuários médios utilizem seus modems para dial-out. As permissões padrões nos arquivos /dev/cua* devem ser suficientes:

```
crw-rw----  1 uucp      dialer    28, 129 Feb 15 14:38 /dev/cuaa1
crw-rw----  1 uucp      dialer    28, 161 Feb 15 14:38 /dev/cuaia1
crw-rw----  1 uucp      dialer    28, 193 Feb 15 14:38 /dev/cuala1
```

Estas permissões permitem que o usuário uucp e que os usuários do grupo dialer utilizem os dispositivos de *call-out*.

17.2.5. Configuração de portas seriais

O dispositivo ttydN (ou cuaaN) é o dispositivo que normalmente você irá desejar abrir para as suas aplicações. Quando um processo abre um dispositivo, ele irá ter um conjunto padrão de configurações de I/O para o terminal.

```
# stty -a -f /dev/ttyd1
```

Quando você altera as configurações para este dispositivo, os ajustes permanecem válidos até que o dispositivo seja fechado. Quando ele é reaberto, ele voltará a utilizar as configurações padrões. Para fazer alterações na configuração padrão, você deve abrir e ajustar as configurações de "estado inicial" do dispositivo. Por exemplo, para habilitar o modo *CLOCAL*, comunicação de 8 bits e o controle de fluxo *XON/XOFF* como padrão para o dispositivo ttyd5, execute:

```
# stty -f /dev/ttyid5 clocal cs8 ixon ixoff
```

Todo o processo de inicialização dos dispositivos seriais é controlado pelo arquivo `/etc/rc.serial`. Este arquivo afeta as configurações padrões dos dispositivos seriais.

Para impedir que configurações sejam modificadas por aplicações, configure os dispositivos em "lock state". Por exemplo, para travar a velocidade do `ttyd5` em 57600 bps, digite:

```
# stty -f /dev/ttyld5 57600
```

Deste momento em diante, as aplicações que abrirem o dispositivo `ttyd5` e tentarem modificar a velocidade da porta terão sua tentativa ignorada e farão uso do dispositivo na velocidade 57600 bps.

Naturalmente, você deve definir que os dispositivos que estiverem em estado de inicialização ou travados tenham permissão de escrita somente para a conta `root`.

17.3. Terminais

Contributed by Sean Kelly.

Terminais proporcionam um meio de acesso conveniente e de baixo-custo ao seu sistema FreeBSD quando você não possui computadores para console ou uma rede para conexão. Esta seção descreve como utilizar terminais com o FreeBSD.

17.3.1. Utilidades e Tipos de Terminais

Os sistemas UNIX® primitivos não possuíam consoles. Diante disso, os usuários conectados executavam seus programas através de terminais ligados nas portas seriais dos computadores. Processo muito similar ao uso de um modem e um programa de conexão para um sistema remoto, para trabalhos em modo texto puro.

Atualmente, os PCs possuem consoles com alta definição gráfica, mas as habilidades de estabelecer uma sessão de login através de uma porta serial foi mantida nos sistemas operacionais UNIX atuais; O FreeBSD não é uma exceção. Utilizando um terminal conectado a uma porta serial você pode logar-se e rodar programas em modo texto normalmente como seria feito em um console ou numa janela de `xterm` do sistema X Window.

Os usuários corporativos tem a opção de conectar diversos terminais num sistema FreeBSD e colocá-los nas mesas de trabalho de seus funcionários. Os usuários domésticos podem utilizar de computadores antigos como IBM PCs ou Macintosh® como terminais de um computador de performance superior rodando FreeBSD. Permitindo que um computador poderoso, que na maioria das vezes seria utilizado por apenas um usuário, torne-se um sistema para múltiplos usuários.

No FreeBSD tem-se basicamente três tipos de terminais:

- [Terminais *Dumb* ou burros](#)
- [PCs atuando como terminais](#)
- [Terminais X](#)

São descritos em sequência cada um dos tipos.

17.3.1.1. Terminais *Dumb* ou burros

Terminais *Dumb* são um tipo característico de equipamento para conexão em computadores através de portas seriais. São chamados "dumb" ou "burros" devido a serem terminais que possuem habilidade única de envio e apresentação de informações em modo texto. Não é possível executar nenhum programa neles. A execução dos programas como editores de texto, compiladores, correio eletrônico, jogos, e outros, é feita pelo computador em que o terminal está conectado.

Existem diversos tipos de terminais burros, fabricados pelos mais variados fabricantes, como por exemplo o VT-100 da *Digital Equipment Corporation*, ou o WY-75 da *Wyse*. Qualquer que seja o tipo funcionará com o FreeBSD. Alguns terminais mais avançados podem até exibir gráficos, porém apenas alguns programas podem fazer uso destas funcionalidades.

Terminais burros são plataformas de trabalho bastante comuns em locais onde os usuários não necessitam de aplicações gráficas como as providas pelo sistema X Window.

17.3.1.2. PCs atuando como terminais

Se um [terminal burro](#) possui habilidades suficientes para exibir, enviar e receber texto, certamente, qualquer computador poderá ser utilizado como um terminal burro. Tudo que será preciso é um cabo apropriado e a *emulação de terminal* realizada por um programa executado no computador.

Tal configuração é comum para uso doméstico. Por exemplo, se tem uma pessoa ocupada trabalhando no seu console do FreeBSD você pode realizar trabalhos em texto ao mesmo tempo de um computador conectado como um terminal ao FreeBSD.

17.3.1.3. Terminais X

Terminais X são o tipo mais sofisticado de terminais disponíveis. Ao invés de serem conectados por portas seriais, estes são usualmente conectados via rede, como por exemplo *Ethernet*. No lugar de executarem aplicações em modo texto, estes podem exibir qualquer aplicação gráfica do X Window.

Os terminais X são apresentados aqui somente para abordagem geral do tema. Entretanto, este capítulo *não* abrange a configuração ou o uso de terminais X.

17.3.2. Configuração

Esta sessão descreve o que é necessário configurar no seu sistema FreeBSD para habilitar o *login* e utilização através de um terminal. É assumido aqui que você tenha previamente habilitado o suporte para portas seriais no seu kernel e que você tenha feito a conexão física do terminal na porta serial.

Relembrando o [Capítulo 7](#) que descreve que o processo `init` é responsável pelo controle e inicialização de todos os processos ao iniciar o sistema. Uma das atividades executadas pelo `init` é ler o arquivo `/etc/ttys` e iniciar o processo `getty` nos terminais disponibilizados. O processo `getty` é responsável pela leitura do nome de *login* e pela inicialização do programa `login`.

Assim, para configurar os terminais para o seu sistema FreeBSD os seguintes passos devem ser realizados como `root`:

1. Adicione uma linha no arquivo `/etc/ttys` para a entrada do diretório `/dev` indicando a porta serial caso esta não esteja descrita no arquivo.
2. Especifique que o `/usr/libexec/getty` deve ser executado na porta, e indique um tipo apropriado de *getty* no arquivo `/etc/gettytab`.
3. Especifique o tipo padrão do terminal.
4. Configure a porta como ``*on*."
5. Especifique se a porta deve ser ``segura."
6. Force o `init` a reler o arquivo `/etc/ttys`.

Como passo opcional, você pode desejar criar um tipo de *getty* personalizado para utilizar no passo 2 fazendo uma entrada no arquivo `/etc/gettytab`. Este capítulo não explica como fazer isso; é recomendável que você leia as páginas de manual [gettytab\(5\)](#) e a [getty\(8\)](#) para informações detalhadas deste processo.

17.3.2.1. Adicionando uma entrada no arquivo `/etc/ttys`

O arquivo `/etc/ttys` lista todas as portas do seu sistema FreeBSD em que você deseja permitir *logins*. Por exemplo, o primeiro console virtual que é definido em uma entrada deste arquivo é o `ttv0`. Você pode realizar *login* devido esta entrada. Este arquivo também contém entradas para outros consoles virtuais, portas seriais, e pseudo-ttys. Para um terminal físico, apenas liste a porta serial que está em `/dev` numa entrada sem a parte do `/dev` (por exemplo, `/dev/ttyv0` deve ser listado como `ttv0`).

A instalação padrão do FreeBSD inclui no arquivo `/etc/ttys` o suporte para as quatro primeiras portas seriais: `tyd0` à `tyd3`. Se você conectar fisicamente um terminal em uma destas portas não será necessário editar o `/etc/ttys` pois as entradas referentes a estes terminais já constarão no arquivo.

Exemplo 17-1. Adicionando entradas de terminais em `/etc/ttys`

Vamos supor que gostaríamos de conectar dois terminais ao sistema: um Wyse-50 e um velho IBM PC 286 executando o programa de terminal **Procomm**, que emula um terminal VT-100. Nós conectamos o Wyse na segunda porta serial e o 286 na sexta

porta serial (uma porta de uma placa multi-serial). As entradas correspondentes a esta configuração no arquivo `/etc/ttys` deveriam ser algo como abaixo:

```
ttyd1 ① "/usr/libexec/getty std.38400" ② wy50 ③ on ④ insecure ⑤  
ttyd5  "/usr/libexec/getty std.19200" vt100 on insecure
```

①

O primeiro campo normalmente especifica o nome de arquivo do terminal e ele é localizado em `/dev`.

②

O segundo campo é o comando executado para esta linha, que é usualmente [getty\(8\)](#). `getty` que inicializa e abre a via de comunicação, definindo a velocidade, além de perguntar por um nome de usuário e em seguida executar o programa [login\(1\)](#).

O programa `getty` aceita um (opcional) parâmetro em sua linha de comando que define o tipo do `getty`. Um tipo de `getty` determina as características de comunicação de um terminal, como taxa de bps e paridade. O programa `getty` lê estas características do arquivo `/etc/gettytab`.

O arquivo `/etc/gettytab` contém várias entradas de terminais, tanto de terminais novos quanto de antigos. Em quase todos os casos, as entradas que iniciam com o texto `std` funcionarão para os terminais conectados fisicamente. Estas entradas ignoram paridade. Há uma entrada `std` para cada taxa de bps de 110 à 115200. É claro que você pode adicionar suas próprias entradas neste arquivo. A página de manual [gettytab\(5\)](#) possui maiores informações sobre o assunto.

Quando estiver configurando o tipo de `getty` no arquivo `/etc/ttys`, tenha certeza de que as configurações de comunicação do terminal sejam equivalentes.

Para o nosso exemplo, o Wyse-50 não usa paridade e conecta à 38400 bps. O 286 PC não usa paridade e conecta à 19200 bps.

③

O terceiro campo é o tipo de terminal usualmente conectado através desta tty. Para portas *dial-up*, *unknown* ou *dialup* são tipicamente definidos neste campo desde que usuários possam fazer *dial up* com praticamente qualquer tipo de terminal ou programa. Para terminais físicos o tipo não muda, portanto, você pode colocar um tipo real de terminal neste campo, baseando-se na coleção de terminais do arquivo [termcap\(5\)](#).

Para o nosso exemplo, o Wyse-50 usa um tipo real de terminal enquanto o 286 PC executa o programa **Procomm** emulando um terminal VT-100.

④

O quarto campo especifica se a porta deve ser ativada. Colocando um *on* neste campo fará com que o processo `init` inicie o programa especificado no segundo campo, `getty`. Caso você defina *off* neste campo o `getty` não será iniciado, o que resultará na impossibilidade de *logins* nesta porta.

5

O campo final é usado para especificar se a porta será segura. Definindo a porta como segura significará que esta é confiável ao ponto de permitir que a conta `root` (ou qualquer outra conta com UID 0) possa realizar *login* nesta porta. Definindo como não segura fará com que não seja permitido *logins* com direitos de acesso como os do `root`. Numa porta não segura os usuários devem fazer acesso ao sistema através de contas sem privilégios de superusuário, e através do uso do [su\(1\)](#) ou algum método similar adquirir os privilégios.

É altamente recomendável que você defina com ```insecure"` tanto os terminais quanto as portas não habilitadas. É bastante fácil fazer o acesso e utilizar `su` caso você precise de privilégios de superusuário.

17.3.2.2. Forçando o `init` a reler o `/etc/ttys`

Após ter realizado as configurações necessárias no arquivo `/etc/ttys` você deve enviar um sinal `SIGHUP` (*hangup*) para o processo `init` para força-lo a reler as configurações e executá-las como descrito no arquivo. Por exemplo:

```
# kill -HUP 1
```

Nota: `init` é sempre o primeiro processo a ser executado no sistema, portanto, ele sempre terá o PID 1.

Se tudo foi configurado corretamente, todos os cabos devidamente conectados, e os terminais ligados, então o processo `getty` deve estar sendo executado em cada um dos terminais e você já poderá ver os *prompts* de *login*.

17.3.3. Resolução de problemas na sua conexão

Mesmo com a maior atenção aos detalhes, algo pode ainda estar errado enquanto configuramos um terminal. Aqui temos uma lista de sintomas e algumas sugestões para correção de eventuais problemas.

17.3.3.1. O *Prompt de Login* não aparece

Tenha certeza de que seu terminal está conectado e ligado. Caso seja um computador pessoal atuando como terminal tenha certeza de que está executando o programa de emulação de terminal na porta correta.

Certifique-se de que os cabos entre seu terminal e o computador FreeBSD estejam conectados firmemente. Confirme estar utilizando o tipo de cabo correto para esta conexão.

Certifique-se de que seu terminal e seu sistema FreeBSD estão configurados com a mesma velocidade em bps e com a mesma configuração de paridade. Caso seja um terminal com tela, verifique os controles de contraste e brilho. Caso seja um terminal de impressão, certifique-se que este possui papel e tinta em quantidade suficiente.

Certifique-se de que o processo `getty` está rodando e servindo um terminal. Por exemplo, para obter uma lista dos processos `getty` em execução use o `ps`, digite:

```
# ps -axww|grep getty
```

Você deverá ver uma entrada referente ao terminal. Por exemplo, a listagem seguinte mostra um `getty` sendo executado na segunda porta serial a `ttyd1` e este está usando o parâmetro `std.38400` referente a entrada no `/etc/gettytab`:

```
22189  dl  Is+    0:00.03 /usr/libexec/getty std.38400 ttyd1
```

Caso não se tenha um processo `getty` em execução, verifique se você ativou a porta no `/etc/ttys`. Lembre-se também de executar `kill -HUP 1` após modificar o arquivo `ttys`.

Caso o processo `getty` esteja rodando e o terminal não apresente o *prompt de login* ou apresente sem permitir a digitação, seu terminal ou cabo podem não suportar a verificação de conexão (*handshaking*). Tente modificar a entrada no `/etc/ttys` de `std.38400` para `3wire.38400` lembrando de executar `kill -HUP 1` após alterar o `/etc/ttys`. O `3wire` é uma entrada similar ao `std`, porém ignora a verificação de conexão. Você pode reduzir a velocidade ou ativar o controle de fluxo quando utilizar `3wire`, para prevenir estouro de áreas de troca (*buffer overflows*).

17.3.3.2. Caso apareçam caracteres estranhos ao invés do *Prompt de Login*

Certifique-se de que o terminal e o FreeBSD tenham a velocidade e a configuração de paridade compatíveis. Verifique no processo `getty` se o tipo de `getty` correto está em uso. Se não, edite o `/etc/ttys` e execute `kill -HUP 1`.

17.3.3.3. Caracteres aparecem duplicados; a senha aparece quando digitada

Altere o terminal (ou o programa de emulação de terminal) de ```half duplex``` ou ```local echo``` para ```full duplex```.

17.4. Serviço de Dial-in

Contribuição de Guy Helmer. Informações adicionais por Sean Kelly.

Configurar o seu sistema FreeBSD para o serviço de *dial-in* é similar ao processo de configurar terminais exceto que você terá de discar com ``modems''.

17.4.1. Modems Externos vs. Internos

Modems externos são mais convenientes para *dial-up*, devido a permitirem ser configurados via parâmetros armazenados em RAM não volátil e eles possuem indicadores luminosos que demonstram o estado de sinais RS-232 importantes. Luzes piscantes impressionam visitantes, e também são úteis na verificação do correto funcionamento do modem.

Modems internos usualmente não permitem configurações via RAM não volátil, limitando a serem realizadas somente através de chaves DIP. Caso seu modem possua algum indicador luminoso será, provavelmente, difícil ver as luzes enquanto seu sistema estiver no local próprio.

17.4.1.1. Modems e Cabos

Se você estiver utilizando um modem externo então será necessário utilizar um cabo apropriado. Um cabo serial do padrão RS-232 longo o suficiente com todos os fios necessários aos sinais:

- Dados transmitidos (SD)
- Dados recebidos (RD)
- Solicitar para enviar (RTS)
- Limpar para enviar (CTS)
- Dados prontos (DSR)
- Terminal de dados pronto (DTR)
- Detecção de portadora (CD)
- Sinal terra (SG)

O FreeBSD necessita dos sinais RTS e CTS para controle de fluxo sobre velocidades acima de 2400 bps, o sinal CD serve para detectar que a chamada foi respondida e a linha estabilizada, e o sinal DTR para reiniciar o modem depois de uma seção completa. Alguns cabos não possuem todos os fios necessários aos sinais, então, se você tiver problemas com a fase de login que não prosegue após a linha ter sido estabelecida, você deve ter um problema com o seu cabo.

Assim como outros sistemas operacionais UNIX®, o FreeBSD usa de sinais físicos para localizar quando uma chamada foi respondida ou a linha estabelecida e para reiniciar o modem após uma chamada. O FreeBSD evita enviar comandos para o modem ou observar por estados reportados pelo modem. Se você está acostumado a conectar modems a sistemas PC-BBS, isto deve parecer impróprio.

17.4.2. Considerações da Interface Serial

O FreeBSD tem suporte as interfaces de comunicação NS8250-, NS16450-, NS16550-, e NS16550A- baseadas na EIA RS-232C (CCITT V.24). Os dispositivos 8250 e 16450 que possuem área de troca de caractere único. O dispositivo 16550 provê áreas de troca de 16 caracteres, o que garante uma melhor performance ao sistema. (Problemas na implementação da 16550's evita o uso da área de troca de 16 caracteres, então use a 16550A's se possível). Devido a dispositivos de área de troca de caractere único requerirem maior processamento do sistema operacional que os dispositivos com área de troca de 16 caracteres, prefira utilizar interfaces seriais com sinais baseados em 16550A. Se o sistema possuir diversas portas seriais ativas ou tiver uma carga de serviços muito grande, as placas seriais baseadas em 16550A são melhores para que se tenham taxas baixas de erros de comunicação.

17.4.3. Rápida Visão Geral

Assim como os terminais, o `init` iniciará um processo `getty` para cada porta serial configurada para aceitar conexões discadas. Por exemplo, se um modem está anexado a `/dev/ttyd0` o comando `ps ax` deve apresentar algo como:

```
4850 ?? I      0:00.09 /usr/libexec/getty V19200 ttyd0
```

Quando um usuário disca para a linha em que o modem está conectado um sinal de CD (*Carrier Detect* - Detecção de Portadora) é reportado pelo modem. O kernel anuncia a detecção da portadora e completa a abertura do `getty` daquela porta. O `getty` envia um *prompt* `login:` para a linha na velocidade especificada. O `getty` verifica se os caracteres estão sendo recebidos corretamente e, numa configuração típica, se ele encontra ruídos de comunicação (provavelmente a velocidade da conexão do modem é diferente da estabelecida no `getty`), o `getty` tenta ajustar a velocidade da linha até que possa receber caracteres cabíveis.

Após o usuário informar seu nome de login, o `getty` executa `/usr/bin/login`, que se responsabiliza pelo login perguntando ao usuário sua senha e posteriormente iniciando o *shell* definido para este.

17.4.4. Arquivos de configuração

Temos três arquivos de configuração do sistema no diretório `/etc` que provavelmente será necessário editar para permitir acesso discado ao seu sistema FreeBSD. O primeiro deles é o `/etc/gettytab` que contém as informações de configuração do processo de servidor `/usr/libexec/getty`. O segundo, o `/etc/ttys`, armazena informações que descrevem ao `/sbin/init` quais dispositivos `tty` devem possuir um processo `getty` rodando sobre eles. E por último, você poderá colocar seus comandos de inicialização no *script* `/etc/rc.serial`.

Temos dois grupos distintos relativos as conexões com modems em UNIX. Um grupo gosta de configurar seus modems e sistemas para que independentemente da velocidade seja possível discar, porém a interface local RS-232 entre o computador e o modem roda numa velocidade travada. O benefício disso é que o usuário remoto sempre verá o

prompt de login imediatamente. O negativo disso é que o sistema não sabe qual a verdadeira velocidade do usuário, o que fará que programas que rodam utilizando toda a tela, como o Emacs, não ajustarem seus métodos de exibição para responderem de maneira melhor quando for executada uma conexão lenta.

O outro grupo configura as interfaces RS-232 de seus modems de maneira a diversificarem suas velocidades baseando-se na velocidade da conexão do usuário remoto. Por exemplo, conexões V.32bis (14.4 Kbps) devem fazer com que o modem execute sua interface RS-232 em 19.2 Kbps, enquanto conexões 2400 bps farão com que a interface RS-232 do modem rode em 2400 bps. Devido ao `getty` não entender qualquer particularidade da velocidade reportada do modem ele dará a mensagem `login:` na velocidade inicial e observará os caracteres que terá em resposta. Se o usuário verificar que os caracteres são impróprios ou errôneo, ele assumirá que os usuários deverão saber que devem pressionar a tecla Enter até que possa ver um prompt reconhecível. Se as velocidades não combinarem o `getty` verificará que o usuário está digitando ``lixo'', ou seja, caracteres impróprios e tentará estabelecer a próxima velocidade e enviar o prompt `login:` novamente. Este procedimento poderá pendurar por um tempo, porém normalmente após uma ou duas pressionadas de tecla o usuário verá um prompt bom. Obviamente, esta sequência de login não aparecerá tão limpa quanto a de método de velocidade travada, mas um usuário de conexão lenta deve receber uma melhor resposta interativa para programas que utilizam toda a tela para exibição.

Esta seção tentará prover informações numa configuração balanceada, mas isso será influenciado tendo em vista a velocidade do modem com a velocidade da conexão.

17.4.4.1. /etc/gettytab

O arquivo `/etc/gettytab` é um arquivo do estilo do [termcap\(5\)](#) de informações de configuração para o [getty\(8\)](#). Por favor, visite a página de manual do [gettytab\(5\)](#) para informações detalhadas sobre o formato do arquivo e a lista de habilidades.

17.4.4.1.1. Configuração de velocidade travada ou fixa

Se você está travando a velocidade de comunicação de seu modem numa específica, você provavelmente não precisará fazer qualquer alteração no arquivo `/etc/gettytab`.

17.4.4.1.2. Configuração de velocidades compatíveis

Você precisará definir uma entrada no arquivo `/etc/gettytab` para fornecer informações ao `getty` sobre velocidades que você deseja utilizar para seu modem. Se você tiver um modem de 2400 bps, você provavelmente poderá usar a entrada `D2400` existente.

```
#
# Terminais de conexões discadas rápidas,
# giram em torno de 2400/1200/300 (podendo inciar em um deles)
D2400|d2400|Fast-Dial-2400:\
      :nx=D1200:tc=2400-baud:
3|D1200|Fast-Dial-1200:\
      :nx=D300:tc=1200-baud:
```

```
5|D300|Fast-Dial-300:\
    :nx=D2400:tc=300-baud:
```

Se você possui um modem de alta velocidade, você provavelmente precisará adicionar uma entrada no arquivo `/etc/gettytab`; temos aqui uma entrada que poderia ser utilizada para um modem de 14.4 Kbps com uma interface de velocidade máxima igual a 19.2 Kbps:

```
#
# Adições para modems V.32bis
#
um|V300|High Speed Modem at 300,8-bit:\
    :nx=V19200:tc=std.300:
un|V1200|High Speed Modem at 1200,8-bit:\
    :nx=V300:tc=std.1200:
uo|V2400|High Speed Modem at 2400,8-bit:\
    :nx=V1200:tc=std.2400:
up|V9600|High Speed Modem at 9600,8-bit:\
    :nx=V2400:tc=std.9600:
uq|V19200|High Speed Modem at 19200,8-bit:\
    :nx=V9600:tc=std.19200:
```

Isto resultará em conexões de 8 bits sem paridade.

O exemplo acima inicia a velocidade de comunicação em 19.2 Kbps (para conexões V.32bis), então girando em torno de 9600 bps (para V.32), 2400 bps, 1200 bps, 300 bps, e retornando para 19.2 Kbps. Velocidades de comunicação cíclicas são implementadas com a habilidade `nx=` ("próxima tabela"). Cada uma das linhas usa uma entrada `tc=` ("continuação de tabela") para a escolha do resto das configurações "padrões" para uma velocidade em específico.

Se você tem um modem de 28.8 Kbps e você quer fazer uso das vantagens de compressão dos modems de 14.4 Kbps, você precisa utilizar velocidades de conexão maiores que 19.2 Kbps. Aqui temos um exemplo de uma entrada no `gettytab` iniciando em 57.6 Kbps:

```
#
# Adições para modems V.32bis ou V.34
# Iniciando em 57.6 Kbps
#
vm|VH300|Very High Speed Modem at 300,8-bit:\
    :nx=VH57600:tc=std.300:
vn|VH1200|Very High Speed Modem at 1200,8-bit:\
    :nx=VH300:tc=std.1200:
vo|VH2400|Very High Speed Modem at 2400,8-bit:\
    :nx=VH1200:tc=std.2400:
vp|VH9600|Very High Speed Modem at 9600,8-bit:\
    :nx=VH2400:tc=std.9600:
vq|VH57600|Very High Speed Modem at 57600,8-bit:\
    :nx=VH9600:tc=std.57600:
```

Caso você tenha uma CPU lenta ou um sistema sobre altas cargas de utilização e não possua portas seriais baseadas em 16550A, você poderá receber erros "sio" "silo" em 57.6 Kbps.

17.4.4.2. /etc/ttys

A configuração do arquivo `/etc/ttys` é abordada no [Exemplo 17-1](#). A configuração para modems é similar porém temos que passar um argumento diferente para o `getty` e especificar um tipo de terminal diferente. O formato geral para ambas as configurações, velocidade travada e velocidades compatíveis são:

```
ttyd0    "/usr/libexec/getty xxx"    dialup on
```

O primeiro item da linha acima é o arquivo especial de dispositivo para esta entrada -- `ttyd0` significando que `/dev/ttyd0` é o arquivo que este `getty` observará. O segundo item, `"/usr/libexec/getty xxx"` (`xxx` será substituído pela habilidade inicial do `gettytab`) que será executado no dispositivo pelo processo `init`. O terceiro item, `dialup`, é o tipo padrão de terminal. O quarto parâmetro, `on`, diz ao `init` que a linha é operacional. Pode haver um quinto parâmetro, `secure`, mas deve ser utilizado somente para terminais fisicamente seguros (como a console do sistema).

O tipo padrão de terminal (no exemplo acima é um `dialup`) poderá depender das configurações nativas. O tipo `dialup` é o padrão para terminais de linhas de acesso discado, portanto, os usuários podem personalizar seus scripts de login para que estes informem que o tipo do terminal é `dialup` e automaticamente sejam feitos ajustes. Entretanto, o autor deste acha mais fácil para suas configurações definir o tipo padrão como `vt102`, desde que os usuários somente usem a emulação de VT102 em seus sistemas.

Depois de realizar estas alterações no `/etc/ttys` você deve enviar um sinal HUP para o processo `init` para que este releia o arquivo e aplique as alterações. Você pode usar o comando

```
# kill -HUP 1
```

para enviar o sinal. Caso esta seja a primeira vez que você configura o sistema, você pode esperar até que seus modems estejam conectados e configurados corretamente para depois sinalizar ao `init`.

17.4.4.2.1. Configuração de velocidade travada ou fixa

Para configurações de velocidade fixa sua entrada em `ttys` precisa especificar ao `getty` a velocidade. Para um modem com a velocidade da porta fixa em 19.2 Kbps, a entrada em `ttys` deve ser algo como isso:

```
ttyd0    "/usr/libexec/getty std.19200"    dialup on
```

Caso seu modem esteja com a velocidade fixa diferente, substitua pelo valor apropriado utilizando `std.velocidade` no lugar de `std.19200`. Certifique-se de utilizar um tipo válido no arquivo `/etc/gettytab`.

17.4.4.2.2. Configuração de velocidades compatíveis

Na configuração de velocidades compatíveis, sua entrada em `ttys` precisa referenciar o início apropriado ```auto-baud"` (sic) em `/etc/gettytab`. Por exemplo, se você adicionar

acima a entrada sugerida para iniciar em 19.2 Kbps (a entrada em `gettytab` contém o ponto de início `V19200`), sua entrada em `ttys` deve parecer com isso:

```
ttyd0    "/usr/libexec/getty V19200"    dialup on
```

17.4.4.3. `/etc/rc.serial`

Modems de alta velocidade como V.32, V.32bis, e V.34 precisam usar o controle físico de fluxo (RTS/CTS). Você pode adicionar o comando `stty` no arquivo `/etc/rc.serial` para definir o controle físico de fluxo do kernel do FreeBSD na porta do modem.

Por exemplo, para definir a configuração de `crtscts`, `termios`, na primeira porta serial de dispositivos de discagem (COM2) as seguintes linhas poderiam ser adicionadas ao `/etc/rc.serial`:

```
# Configuração de inicialização de porta serial
stty -f /dev/ttyid1 crtscts
stty -f /dev/cuaia1 crtscts
```

17.4.5. Configurações do Modem

Caso você possua um modem que possibilite que seus parâmetros possam ser definidos em RAM não volátil, você precisará utilizar um programa de terminal (como o Telix sobre MS-DOS® ou o `tip` sobre o FreeBSD) para definir os parâmetros. Conecte ao modem utilizando como inicial a mesma velocidade de comunicação que o `getty` utilizará e configure o modem com RAM não volátil combinando com os seguintes requerimentos:

- CD alertou quando conectado
- DTR alertou quando em operação; apagando DTR e conectando a linha, e reinicia o modem
- CTS alertou a transmissão do controle de fluxo
- Desativa o controle de fluxo XON/XOFF
- RTS controle de fluxo de dados recebidos
- Modo silencioso (não retorna códigos)
- Sem retorno de comando

Por favor, leia a documentação do seu modem para identificar quais comandos e configurações de chaves DIP você precisará dar ao modem.

Por exemplo, para definir os parâmetros acima num modem externo U.S. Robotics® Sportster® 14,400 você poderia dar estes comandos ao modem:

```
ATZ
AT&C1&D2&H1&I0&R2&W
```

Você pode querer também usar esta oportunidade para fazer outros ajustes na configuração do modem, como por exemplo se este utilizará compressão V.42bis ou MNP5.

O modem externo U.S. Robotics Sportster 14,400 também possui algumas chaves DIP que necessitam ser configuradas; para outros modems talvez possa utilizar estas configurações, por exemplo:

- Chave 1: PARA CIMA -- DTR Normal
- Chave 2: N/D (Códigos verbais como resultado/Códigos numéricos como resultado)
- Chave 3: PARA CIMA -- Supressão de códigos de resultado
- Chave 4: PARA BAIXO -- Sem emissão de comandos quando desconectado
- Chave 5: PARA CIMA -- Resposta automática
- Chave 6: PARA CIMA -- Detecção normal de portadora
- Chave 7: PARA CIMA -- Carregar padrões da NVRAM
- Chave 8: N/D (Modo inteligente/Modo burro)

Códigos de resultados devem ser suprimidos para modems de discagem o que evitará problemas que podem ocorrer se erroneamente o `getty` dar um prompt de `login`: para o modem que estiver em modo de comando e o modem neste estado retornaria o comando ou o código de resultado. Esta sequência pode resultar numa conversa boba entre o `getty` e o modem.

17.4.5.1. Configuração de velocidade travada ou fixa

Para a configuração de velocidade fixa será necessário definir no modem que este mantenha constante a velocidade de comunicação entre ele e o computador independente da velocidade de comunicação. Num modem externo U.S. Robotics Sportster 14,400 estes comandos fixarão a velocidade de comunicação entre o modem e o computador na velocidade fornecida pelos comandos:

```
ATZ  
AT&B1&W
```

17.4.5.2. Configuração de velocidades compatíveis

Para a configuração de velocidades compatíveis ou variáveis você precisará configurar seu modem para que este ajuste sua velocidade de acordo com a velocidade que for percebida na chamada. Num modem externo U.S. Robotics Sportster 14,400 estes comandos travarão a velocidade de correção de erros na mesma que for utilizada no envio dos comandos, porém permitirá que a porta serial varie em conexões sem correção de erros:

```
ATZ  
AT&B2&W
```

17.4.5.3. Verificando a configuração do modem

A maioria dos modems de alta velocidade disponibiliza comandos para verificação dos parâmetros correntes de operação numa forma humanamente legível. Em um modem externo U.S. Robotics Sportster 14,400 o comando `ATI5` lista as configurações armazenadas em RAM não volátil. Para verificar as configurações reais de operação do

modem (como determinadas nas configurações das chaves DIP), use o comando `ATZ` e então `ATI4`.

Se você tem um tipo diferente de modem verifique o manual para saber como checar as configurações feitas.

17.4.6. Resolução de Problemas

Temos aqui alguns passos que você pode seguir para verificar o modem no seu sistema em relação a discagem e atendimento.

17.4.6.1. Checagem do sistema FreeBSD

Conecte fisicamente seu modem no seu sistema FreeBSD, inicie o sistema e se seu modem tiver luzes que indicam seu estado, observe se a luz indicadora DTR é sinalizada quando o prompt `login:` aparece no console do sistema -- se a luz acender, significa que o FreeBSD iniciou o processo `getty` na porta correta de comunicação e está esperando pelo modem para aceitar uma chamada.

Se a luz indicadora DTR não acender, execute `login` no sistema FreeBSD no console e através do comando `ps ax` verifique se o FreeBSD está tentando executar o processo `getty` na porta correta. Você deve ver uma linha como essa em meio aos processos listados:

```
114 ?? I      0:00.10 /usr/libexec/getty V19200 ttyd0
115 ?? I      0:00.10 /usr/libexec/getty V19200 ttyd1
```

Se você ver algo diferente, como isso:

```
114 d0 I      0:00.10 /usr/libexec/getty V19200 ttyd0
```

e o modem ainda não ter aceitado uma chamada, isso significa que o `getty` completou a abertura das portas de comunicação. Isto pode indicar um problema com o cabo ou uma mal configuração do modem, devido ao `getty` não estar habilitado a abrir a porta de comunicação até que o CD (detecção de portadora) seja informado pelo modem.

Caso você não veja nenhum processo `getty` indicado para abrir a porta `ttydN` designada, recheque suas entradas no arquivo `/etc/ttys` para ver se não há erros. Verifique também o arquivo de log `/var/log/messages` para ver se não há nenhuma mensagem de log do `init` ou do `getty` relativa a um problema. Se houver verifique novamente os arquivos de configuração `/etc/ttys` e o `/etc/gettytab`, assim como o arquivo especial de dispositivo `/dev/ttydN`, buscando por qualquer erro, entradas erradas ou arquivos especiais de dispositivos perdidos.

17.4.6.2. Tentando conectar-se ao sistema discando para ele

Tente discar para o sistema; tenha certeza de usar 8 bits, sem paridade, e 1 bit de parada no sistema remoto. Se você não conseguir um prompt correto, ou receber caracteres impróprios, tente pressionar `Enter` uma vez por segundo. Se ainda assim você não ver um prompt `login:` tente enviar um comando `BREAK`. Caso você esteja utilizando um

modem de alta velocidade para a conexão discada, tente discar novamente após o travamento da velocidade da interface para a linha de discagem (via o comando `AT&B1` como em um modem U.S. Robotics Sportster, por exemplo).

Se você ainda assim não conseguir um prompt `login:`, verifique novamente o `/etc/gettytab` e verifique que:

- O nome da capacidade inicial especificada no `/etc/ttys` para a linha combina com o nome da capacidade em `/etc/gettytab`
- Cada entrada `nx=` combine com outro nome de capacidade `gettytab`
- Cada entrada `tc=` combine com outro nome de capacidade `gettytab`

Se você disca mas o modem no sistema FreeBSD não responde, certifique-se de que o modem está configurado para responder a chamada telefônica quando DTR é informado. Se o modem parecer estar configurado corretamente, verifique se o DTR da linha é informado observando as luzes indicadoras do modem (se ele tiver alguma).

Se você tiver feito tudo isso diversas vezes e ainda assim não funcionar, descanse e tente mais tarde. Se ainda assim não funcionar, você pode, talvez, enviar um email para [lista de discussão FreeBSD de perguntas genéricas](#) descrevendo seu modem e seu problema e o pessoal bom da lista tentará te ajudar.

17.5. Serviço de Dial-out

A seguir, dicas para que seu sistema conecte em outro computador através de um modem. Isto é apropriado para estabilizar uma sessão de terminal com um sistema remoto.

Isto pode ser útil para logar em um BBS.

Este tipo de conexão pode ser de extremo auxílio para conseguir um arquivo na Internet caso você tenha problemas com PPP. Caso você precise de fazer FTP de algo e o PPP falhar, use a sessão de terminal para o FTP, então use `zmodem` para transferir o arquivo para a sua máquina.

17.5.1. Meu modem Stock Hayes não é suportado, o que posso fazer?

Atualmente, a página de manual para o `tip` está desatualizada. Há um discador genérico para Hayes já embutido. Só é necessário usar `at=hayes` no seu arquivo `/etc/remote`.

O driver Hayes não é inteligente o bastante para reconhecer algumas das avançadas funcionalidades dos modems--messages mais novos como *BUSY*, *NO DIALTONE*, ou *CONNECT 115200*, estas apenas irão confundí-lo. Você deve desativar estas mensagens quando utilizar o `tip` (utilizando `ATX0&W`).

Além disso, o intervalo entre discagens do `tip` é de 60 segundos. Seu modem deve utilizar um valor menor ou este achará que há um problema na comunicação. Tente `ATS7=45&W`.

Nota: O `tip` ainda não tem suporte completo a modems Hayes. A solução é editar o arquivo `tipconf.h` no diretório `/usr/src/usr.bin/tip/tip`. Obviamente, você precisará do código fonte do sistema para fazer isso.

Altere a linha `#define HAYES 0` para `#define HAYES 1`. então execute os comandos `make` e `make install`. Tudo funcionará bem após isso.

17.5.2. Como eu faço para entrar com estes comandos AT?

Faça o que é chamada de entrada "direta" em seu arquivo `/etc/remote`. Por exemplo, se seu modem está ligado fisicamente na primeira porta serial, a `/dev/cuaa0`, então insira a seguinte linha:

```
cuaa0:dv=/dev/cuaa0:br#19200:pa=none
```

Informe a maior velocidade suportada pelo seu modem na habilidade `br`. então, digite `tip cuaa0` e você será conectado ao seu modem.

Caso não exista um `/dev/cuaa0` no seu sistema, faça isso:

```
# cd /dev
# sh MAKEDEV cuaa0
```

Ou então, faça uso do comando `cu` como usuário `root` da seguinte maneira:

```
# cu -llinha -svelocidade
```

`linha` é a porta serial (`/dev/cuaa0`) e a `velocidade` é a velocidade (57600). Quando você terminar de entrar com os comandos AT tecler `~`, para sair.

17.5.3. O sinal @ para a habilidade pn não funciona!

O sinal `@` no número do telefone diz ao `tip` para que este verifique o arquivo `/etc/phones` para buscar um número telefônico. Mas o sinal `@` também é um caractere especial com habilidades em arquivos como o `/etc/remote`. Fuja disso com um sinal `/:`

```
pn=\@
```

17.5.4. Como discar para um número telefônico pela linha de comando?

Coloque o que é chamado de entrada "genérica" no seu arquivo `/etc/remote`. Por exemplo:

```
tip115200|Disca para qualquer número telefônico à 115200 bps:\
:dv=/dev/cuaa0:br#115200:at=hayes:pa=none:du:
tip57600|Disca para qualquer número telefônico à 57600 bps:\
:dv=/dev/cuaa0:br#57600:at=hayes:pa=none:du:
```

então você pode fazer coisas como:

```
# tip -115200 5551234
```

Se você preferir utilizar o comando `cu` ao invés do `tip`, use uma entrada genérica para o comando `cu`:

```
cu115200|Use o comando cu para disar à 115200bps:\
      :dv=/dev/cuaa1:br#57600:at=hayes:pa=none:du:
```

e digite:

```
# cu 5551234 -s 115200
```

17.5.5. Toda vez que eu for fazer isso tenho que fazê-lo na mesma velocidade?

Coloque uma entrada para o `tip1200` ou `cu1200`, mas vá em frente e utilize qualquer velocidade que seja condizente com a habilidade `br`. O `tip` acha um bom padrão 1200 bps que é o motivo para a entrada `tip1200`. Você não é obrigado a usar 1200 bps, no entanto.

17.5.6. Eu acesso diversos computadores através de um servidor de terminais

Particularmente para quem fica esperando até estar conectado digitando `CONNECT <host>` toda vez, utilize a habilidade `cm` do `tip`. Por exemplo, estas entradas em `/etc/remote`:

```
pain|pain.deep13.com|Computador do Forrester:\
      :cm=CONNECT pain\n:tc=deep13:
muffin|muffin.deep13.com|Computador do Frank:\
      :cm=CONNECT muffin\n:tc=deep13:
deep13:Servidor de Terminais do Instituto Gizmonics:\
      :dv=/dev/cuaa2:br#38400:at=hayes:du:pa=none:pn=5551234:
```

permitir que você digite `tip pain` ou `tip muffin` para conectar aos computadores `pain` ou `muffin`, e `tip deep13` para acessar o servidor de terminais.

17.5.7. O Tip pode tentar utilizar mais de uma linha para cada local?

Este problema ocorre com frequência em universidades que possuem diversas linhas com modems e milhares de estudantes tentando utilizá-los.

Faça uma entrada para sua universidade em `/etc/remote` e use o `@` para a habilidade `pn`:

```
big-university:\
      :pn=@:tc=dialout
dialout:\
      :dv=/dev/cuaa3:br#9600:at=courier:du:pa=none:
```

então, informe os números da universidade no `/etc/phones`:

```
big-university 5551111
big-university 5551112
big-university 5551113
big-university 5551114
```

O `tip` tentará cada um dos números em ordem e então desistirá. Se você desejar que outras tentativas sejam feitas, execute o `tip` num laço de repetição como o `while`.

17.5.8. Por que eu tenho que pressionar **Ctrl+P** duas vezes para enviar um **Ctrl+P**?

A combinação **Ctrl+P** é o padrão para "forçar" um caractere, utilizado para dizer ao `tip` que o próximo caractere é um dado literal. Você pode utilizar o escape `~s` para definir o caractere de forçar qualquer outro, o que significa "definir uma variável."

Digite `~sforce=caractere único` seguido pela nova linha. O *caractere único* é qualquer caractere. Se você não informar o *caractere único*, então será forçado o caractere nulo, que você obtém com **Ctrl+2** ou **Ctrl+Space**. Um bom valor para o *caractere único* é **Shift+Ctrl+6**, que só é utilizado em alguns servidores de terminais.

Você pode ter o caractere que quiser especificando num arquivo em seu diretório pessoal, `$HOME/.tiprc`:

```
force=<caractere único>
```

17.5.9. Repentinamente tudo que é digitado aparece em letra maiúscula??

Você tem que pressionar **Ctrl+A**, o "caractere de levantamento" do `tip` especialmente designado para pessoas com a tecla caps-lock quebradas. Use o `~s` como abaixo para definir algo cabível como `raisechar`(caractere de levantamento). De fato, você pode definir o mesmo que o caractere de forçar caso você nunca venha a fazer uso destas funcionalidades.

Temos aqui um exemplo, um simples arquivo `.tiprc` perfeito para usuários do **Emacs** que necessitam digitar **Ctrl+2** e **Ctrl+A** diversas vezes:

```
force=^^
raisechar=^^
```

O `^^` é o **Shift+Ctrl+6**.

17.5.10. Como transfiro arquivos com o `tip`?

Se você estiver falando de um outro sistema UNIX®, você pode enviar e receber arquivos com `~p` (put) e `~t` (take). Estes comandos executam `cat` e `echo` no sistema remoto para aceitar e enviar arquivos. A sintaxe é:

`~p` Arquivo-local [Arquivo-remoto]

`~t` Arquivo-remoto [Arquivo-local]

não há checagem de erro, então você deve utilizar um outro protocolo para isso, como `zmodem`.

17.5.11. Como eu executo o `zmodem` com o `tip`?

Para receber um arquivo, inicie o programa de envio no sistema remoto. então, digite `~C rz` para iniciar o recebimento localmente.

Para enviar arquivos, inicie o programa de recebimento no sistema remoto. então, digite `~C sz arquivos` para enviá-los para o sistema remoto.

17.6. Configurando uma console serial

Contribuição de Kazutaka YOKOTA. Baseado no documento de Bill Paul.

17.6.1. Introdução

O FreeBSD tem a habilidade de bootar em um sistema que possua somente um terminal burro como console, conectado na porta serial. Esta configuração pode ser útil para duas classes de pessoas: administradores de sistema que desejam instalar o FreeBSD em máquinas que não possuem teclado ou monitor conectados, e desenvolvedores que querem depurar o kernel ou drivers de dispositivos.

Como descrito em [Capítulo 7](#), o FreeBSD emprega em seu boot três estágios. Os primeiros dois estágios estão no código do bloco de boot que é armazenado no início da partição do disco de boot do FreeBSD. O bloco de boot carregará e executará o carregador de boot (`/boot/loader`) como o terceiro estágio de boot.

Visando iniciar a console serial você deve configurar o código do bloco de boot, o carregador de boot e o kernel do sistema.

17.6.2. Configuração da console serial

1. Prepare um cabo serial.

Você precisará de um cabo serial para modem ou um cabo serial padrão e um conector para serial de modem. Visite [Seção 17.2.2](#) para um debate sobre cabos seriais.

2. Desconecte seu teclado.

A maioria dos sistemas PC testa a existência de um teclado durante o POST (Power-On Self-Test) e gerará um erro se o teclado não for detectado. Algumas máquinas reclamam a falta de um teclado de maneira tal que não prosseguem para o boot até que se tenha um teclado conectado.

Se seu computador retorna este erro mas ainda assim prossegue com o boot então você não terá que fazer nada especial. (Algumas máquinas que possuem a BIOS Phoenix instalada apenas retornam ``Keyboard failed" e continuam com o boot normalmente.)

Se seu computador se recusa a continuar o boot sem ter um teclado conectado então você terá que configurar a BIOS para que este erro seja ignorado(se for possível). Consulte o manual da sua placa-mãe para verificar como fazer isso.

Dica: Configurar na BIOS que o teclado ``não foi instalado" não significa que você não poderá fazer uso do seu teclado posteriormente. Isso é para dizer para a BIOS que não cheque por um teclado quando a máquina for ligada, o que fará com que ela não reporte que o teclado não está conectado. Você pode deixar o teclado conectado mesmo com esta configuração feita e este funcionará normalmente.

Nota: Se seu sistema possui um mouse PS/2®, são grandes as chances de você ter que desconectar seu mouse assim como seu teclado. Isto devido a mouses PS/2 compartilharem dispositivos físicos com teclados e deixando o mouse conectado pode resultar na detecção de que o teclado também está conectado. Isto é reportado pelo sistema Gateway 2000 Pentium 90 MHz com a BIOS AMI, que se comporta desta maneira. No geral, isto não é um problema desde que o mouse não é muito útil sem um teclado.

3. Conecte o terminal burro na COM1 (sio0).

Se você não possui um terminal burro, é possível fazer uso de um PC/XT antigo com um programa de modem, ou uma porta serial num outro sistema UNIX®. Se você não possui uma COM1 (sio0), consiga uma. Para o que se pretende não há uma maneira de selecionar outra porta que não a COM1, para ser utilizada com o bloco de boot sem recompilá-lo. Se você já está usando a COM1 para outro dispositivo, você terá que removê-lo temporariamente para instalar um novo bloco de boot e kernel, uma vez que seu FreeBSD esteja em execução. (É assumido que a COM1 estará disponível neste servidor sem monitor e teclado; se você realmente tem a necessidade de usar a COM1 para outra coisa (e você não pode mudar a utilização da COM1 para a COM2 (sio1)), então provavelmente você não tem que se perturbar com isso em primeiro lugar.)

4. Certifique-se de que o arquivo de configuração do seu kernel tem as configurações apropriadas para a COM1 (sio0).

são configurações relevantes:

0x10

Ativa o suporte a console nesta unidade. As outras configurações serão ignoradas se esta for definida. Atualmente, pelo menos uma unidade tem que ter a configuração de suporte a console; é desejável que seja a primeira (na ordem do arquivo de configuração). Esta opção sozinha não fará da porta serial uma porta para console. Defina a configuração seguinte ou use a opção *-h* descrita abaixo, junto com a configuração de console.

0x20

Força que esta unidade seja uma console (a menos que exista uma outra com maior prioridade), sem levar em consideração a opção *-h* discutida abaixo. Esta configuração sobrepõe a opção *COMCONSOLE* na versão 2 do FreeBSD 2.X. O valor de configuração *0x20* deve ser usado junto com o valor *0x10*.

0x40

Reserve esta unidade (em conjunto com *0x10*) e fará com que a unidade fique indisponível para acesso normal. Você não deve configurar este valor para a unidade de porta serial que você quer usar como console. A única utilização deste valor é para designar a unidade para depuração remota de kernel. Visite [The Developer's Handbook](#) para maiores informações sobre depuração remota.

Nota: No FreeBSD 4.0 ou acima a semântica do valor *0x40* é ligeiramente diferente e há um outro valor para especificar que a porta serial será utilizada para depuração remota.

Exemplo:

```
device sio0 at isa? port IO_COM1 flags 0x10 irq 4
```

Veja a página de manual do [sio\(4\)](#) para maiores detalhes.

Se os valores não forem definidos, você precisará executar UserConfig (em um console diferente) ou recompilar o kernel.

5. Crie o arquivo `boot.config` no diretório root da partição *a* no disco de boot.

Este arquivo instruirá o código do bloco de boot do sistema. Visando ativar a console serial, você precisará de uma ou mais da seguinte opção `options--` se você quiser múltiplas opções inclua todas na mesma linha:

-h

Alterna entre console interna e seriais. Você pode utilizar para alternar dispositivos de console. Por exemplo, se você iniciar o sistema com a console interna (monitor de vídeo) você pode utilizar *-h* para direcionar o carregamento do boot e o kernel para a porta serial como um dispositivo de console. No contrário, com *-h* você pode direcionar da console da porta serial para a sua console do monitor de vídeo.

-D

Alterna configurações de console única ou dupla. Numa configuração de console única você terá unicamente a console interna (monitor de vídeo) ou a console da porta serial, dependendo do que foi configurado na opção *-h*. Numa configuração de console dupla, ambas, a console interna e a console serial serão utilizáveis simultaneamente, independente da configuração do *-h*. Entretanto, a configuração de console dupla terá efeito somente durante a execução do bloco de boot. Uma vez que o controlador de boot assumir o controle, a console especificada pela opção *-h* será a única console.

-P

Faz com que o bloco de boot tente detectar o teclado. Caso não encontre, a opção *-D* e *-h* são ajustadas automaticamente.

Nota: Devido as definições de espaço da versão atual dos blocos de boot a opção *-P* só é capaz de detectar teclados estendidos. Teclados com menos de 101 teclas (e sem as teclas F11 e F12) não poderão ser detectados. Alguns tipos de teclados de computadores portáteis (laptops) não poderão ser detectados corretamente devido a esta limitação. Se este é o caso do seu sistema, você terá que abandonar o uso da opção *-P*. Infelizmente, não há opções para este problema.

Utilize uma opção qualquer de *-P* para selecionar a console automaticamente, ou a opção *-h* para ativar a console serial.

Você pode ainda incluir outras opções como bem descrito em [boot\(8\)](#).

As opções, com exceção da opção *-P*, serão passadas ao carregador de boot (`/boot/loader`). O carregador de boot determinará entre o vídeo ou a porta serial qual deverá ser utilizado para a console examinando a configuração da opção *-h*. Isto significa que, se você especificar a opção *-D* mas não a opção *-h* no arquivo `/boot.config`, você poderá utilizar a porta serial como console apenas durante o bloco de boot; o carregador de boot usará o vídeo como console.

6. Boot o computador.

Quando você iniciar o seu sistema FreeBSD, o bloco de boot listará o conteúdo de `/boot.config` na console. Por exemplo:

```
/boot.config: -P  
Keyboard: no
```

A segunda linha aparece somente se você colocar a opção *-P* no `/boot.config` que indicará a presença/ausência do teclado. Estas mensagens irão tanto para a console serial, ou para a interna, ou ainda para ambas, dependerá das opções em `/boot.config`.

Opções	Mensagens irão para
none	console interna
-h	console serial
-D	console serial e interna
-Dh	console serial e interna
-P, teclado presente	console interna
-P, teclado ausente	console serial

Após as mensagens acima haverá uma pequena pausa antes que bloco de boot continue o carregamento do boot e antes que qualquer mensagem seja listada na console. Sob circunstâncias normais, você não precisará interromper o boot, mas você pode querer fazer algo para certificar que tudo foi configurado corretamente.

Pressione qualquer tecla diferente de Enter na console para interromper o processo de boot. O bloco de boot apresentará um prompt que permitirá fazer ações adicionais. Você deve ver algo como:

```
>> FreeBSD/i386 BOOT
Default: 0:wd(0,a)/boot/loader
boot:
```

Verifique se a mensagem acima aparece somente na serial ou na interna, ou ainda em ambas, de acordo como você definiu em `/boot.config`. Se a mensagem aparece na console correta, pressione Enter para continuar o processo de boot.

Se você quer usar a console serial mas você não vê o prompt no terminal serial, algo está errado em suas configurações. No momento certo, você entrou com a opção `-h` e pressionou Enter/Return (se possível) para dizer ao bloco de boot (e então ao carregador de boot e ao kernel) para escolher a porta serial para a console. Uma vez que o sistema estiver utilizável, volte e verifique o que está errado.

Após o carregamento do controlador de boot, você estará no terceiro estágio do processo de boot, você pode ainda alternar entre a console interna e a serial configurando as variáveis de ambiente de maneira apropriada no carregador de boot. Consulte a [Seção 17.6.5](#).

17.6.3. Sumário

Temos aqui um sumário das várias configurações discutidas nesta seção e eventualmente a console selecionada.

17.6.3.1. Caso 1: Você definiu o valor 0x10 para sio0

```
device sio0 at isa? port IO_COM1 flags 0x10 irq 4
```

Opções em /boot.config	Console durante o bloco de boot	Console durante o carregador de boot	Console no kernel
nenhuma	interna	interna	interna
-h	serial	serial	serial
-D	serial e interna	interna	interna
-Dh	serial e interna	serial	serial
-P, teclado presente	interna	interna	interna
-P, teclado ausente	serial e interna	serial	serial

17.6.3.2. Caso 2: Você definiu o valor 0x30 para sio0

```
device sio0 at isa? port IO_COM1 flags 0x30 irq 4
```

Opções em /boot.config	Console durante o bloco de boot	Console durante o carregador de boot	Console no kernel
nenhuma	interna	interna	serial
-h	serial	serial	serial
-D	serial e interna	interna	serial
-Dh	serial e interna	serial	serial
-P, teclado presente	interna	interna	serial
-P, teclado ausente	serial e interna	serial	serial

17.6.4. Dicas para a console serial

17.6.4.1. Configurante a mais rápida velocidade da porta serial

Por padrão, a porta serial está configurada para: 9600 baud, 8 bits, sem paridade, e 1 bit de parada. Se você quiser alterar a velocidade terá de recompilar pelo menos o bloco de boot. Adicione a seguinte linha ao `/etc/make.conf` e compile o novo bloco de boot:

```
BOOT_COMCONSOLE_SPEED=19200
```

Se a console serial está configurada de maneira diferente de estar realizando o boot com `-h`, ou se a console serial utilizada pelo kernel é diferente da utilizada pelo bloco de boot, então você deve também adicionar a seguinte opção no arquivo de configuração do seu kernel e compilar um novo kernel:

```
options CONSPEED=19200
```

17.6.4.2. Utilizando outra porta serial que não seja a `sio0` para a console

Utilizar outra porta que não seja a `sio0` como a que a console usa necessitará de recompilação. Se você deseja utilizar uma outra porta serial por qualquer que seja a razão, recompile o bloco de boot, o carregador de boot e o kernel como a seguir.

1. Obtenha o código fonte do kernel. (Visite o [Capítulo 21](#))
2. Edite o `/etc/make.conf` e configure `BOOT_COMCONSOLE_PORT` para o endereço da porta que você quer utilizar (0x3F8, 0x2F8, 0x3E8 ou 0x2E8). Somente de `sio0` a `sio3` (COM1 a COM4) podem ser utilizadas; placas de multiseriais não funcionarão. Não é necessária a configuração de interrupção.
3. Crie um arquivo de configuração de kernel personalizado e adicione as opções apropriadas para a porta serial que você deseja utilizar. Por exemplo, se você quer usar a `sio1` (COM2) como console:
4.

```
device sio1 at isa? port IO_COM2 flags 0x10 irq 3
```

ou

```
device sio1 at isa? port IO_COM2 flags 0x30 irq 3
```

As opções de console para as outras portas não devem ser definidas.

5. Recompile e instale o bloco de boot e o carregador de boot:
6.

```
# cd /sys/boot
```
7.

```
# make
```
8.

```
# make install
```
9. Reconstrua e instale o kernel.
10. Grave o bloco de boot no disco de boot com [disklabel\(8\)](#) e boot com este novo kernel.

17.6.4.3. Entrando no DDB Debugger através da linha serial

Se você deseja definir que as informações de depuração do kernel sejam emitidas na console serial (muito útil para diagnósticos remotos, porém também perigoso se houver a geração de um `BREAK` falso na porta serial!) então você deve compilar o seu kernel com as seguintes opções:

```
options BREAK_TO_DEBUGGER
options DDB
```

17.6.4.4. Obtendo um prompt de login na console serial

Enquanto isso não é requerido, você pode desejar obter um prompt de *login* através da linha serial, o que até agora permitiria ver mensagens do boot e entrar em depuração de kernel através da console serial. Veja aqui como fazer isso.

Abra o arquivo `/etc/ttys` com um editor de textos e localize as linhas:

```
ttyd0 "/usr/libexec/getty std.9600" unknown off secure
ttyd1 "/usr/libexec/getty std.9600" unknown off secure
ttyd2 "/usr/libexec/getty std.9600" unknown off secure
ttyd3 "/usr/libexec/getty std.9600" unknown off secure
```

De *ttyd0* a *ttyd3* que corresponde da COM1 a COM4. Altere onde está *off* para *on* para as portas que desejar. Se você tiver alterado a velocidade da porta serial, você precisará alterar *std.9600* para a configuração correspondente, como por exemplo *std.19200*.

Você pode também querer alterar o tipo do terminal de *unknown* para o tipo atual do seu terminal serial.

Após editar o arquivo você deve executar `kill -HUP 1` para que as alterações realizadas tenham efeito.

17.6.5. Alterando a console através do carregador de boot

Seções anteriores descreveram como configurar uma console no bloco de boot. Esta seção demonstra que você pode especificar a console entrando com alguns comandos e configurando algumas variáveis de ambiente no carregador de boot. Como o carregador de boot é executado como o terceiro estágio do processo de boot, depois do bloco de boot, as configurações do carregador de boot irão sobrepôr as configurações definidas no bloco de boot.

17.6.5.1. Configurando a console serial

Você pode facilmente especificar ao carregador de boot e ao kernel para que estes usem a console serial adicionando apenas uma linha em `/boot/loader.rc`:

```
set console=comconsole
```

Isto terá efeito apesar das configurações feitas no bloco de boot, como discutido em seções anteriores.

Prefira colocar a linha acima como a primeira linha do `/boot/loader.rc` pois assim você verá as mensagens de boot na console serial o mais rápido possível.

Da mesma maneira que você pode especificar o uso da console interna com:

```
set console=vidconsole
```

Se você não configurou a variável de ambiente `console` no carregador de boot, este, e posteriormente o kernel, utilizarão qualquer console indicada pela opção `-h` no bloco de boot.

Nas versões 3.2 ou posteriores, você pode especificar a console em `/boot/loader.conf.local` ou `/boot/loader.conf`, assim também como no `/boot/loader.rc`. Neste método, seu `/boot/loader.rc` deve apresentar algo como:

```
include /boot/loader.4th
start
```

então, crie o `/boot/loader.conf.local` e coloque a seguinte linha nele.

```
console=comconsole
```

ou

```
console=vidconsole
```

Visite a [loader.conf\(5\)](#) para mais informações.

Nota: Neste momento, o carregador de boot não tem equivalente para a opção `-P` como no bloco de boot, e não há o fornecimento de uma seleção automática de console interna ou serial baseadas na presença do teclado.

17.6.5.2. Utilizando outra porta serial que não seja a `sio0` para a console

Você precisa recompilar o carregador de boot para poder utilizar outra porta serial que não seja a `sio0` como console. Siga os procedimentos descritos na [Seção 17.6.4.2](#).

17.6.6. Advertências

A idéia aqui é a de permitir que pessoas configurem servidores dedicados que não requerem dispositivos gráficos ou teclados conectados. Desafortunadamente, enquanto a maioria dos sistemas permitirá que seja feito o boot sem um teclado, há um pequeno grupo de equipamentos que não permitirão o boot sem um adaptador gráfico(placa de vídeo). Computadores com AMI BIOS podem ser configurados para que façam o boot sem um adaptador gráfico instalado simplesmente definindo na configuração da CMOS que o "adaptador gráfico não foi instalado".

Entretanto, diversas máquinas não disponibilizarão esta opção e se negarão a continuarem até o boot sem um adaptador gráfico conectado ao equipamento. Com estas máquinas você terá que ter alguma placa gráfica conectada ao equipamento(mesmo que esta for uma péssima placa monocromática) ainda que você não conectará o monitor. Você também pode tentar instalar uma AMI BIOS.

Capítulo 18. PPP e SLIP

Índice

- 18.1. [Sinópse](#)
- 18.2. [Utilizando PPP em modo usuário](#)
- 18.3. [Usando PPP em modo kernel](#)
- 18.4. [Resolução de Problemas](#)
- 18.5. [Usando PPP sobre Ethernet \(PPPoE\)](#)
- 18.6. [Usando PPP sobre ATM \(PPPoA\)](#)
- 18.7. [Usando SLIP](#)

Reestruturado, reorganizado e atualizado por Jim Mock.

18.1. Sinópse

O FreeBSD possui várias formas de ligar um computador a outro. Para estabelecer uma rede ou conexão com a Internet através de um modem discado, ou para permitir que outros façam isso através de você, é necessário o uso de PPP ou SLIP. Esse capítulo descreve como configurar esses serviços de comunicação em detalhes.

Após ler este capítulo, você saberá:

- Como ajustar PPP em modo usuário.
- Como ajustar PPP em modo kernel.
- Como ajustar PPPoE (PPP over Ethernet).
- Como ajustar PPPoA (PPP over ATM).
- Como configurar e ajustar um cliente SLIP e um servidor.

Antes de ler este capítulo você deve:

- Estar familiarizado com a terminologia básica utilizada em redes.
- Ter noções básicas e entender o propósito de uma conexão discada, do protocolo PPP e/ou SLIP.

Você deve estar curioso para saber qual a diferença entre PPP em modo usuário e em modo kernel. A resposta é simples: em modo usuário o PPP processa o tráfego entrante e sai em um processo normal em vez de fazer isso internamente no kernel. Isso é caro em termos de cópias de dados geradas entre o kernel e o processo, mas permite uma implementação PPP muito mais rica em funcionalidades. PPP em modo usuário usa o dispositivo `tun` para se comunicar com o mundo externo onde o PPP em modo kernel utiliza o dispositivo `ppp`.

Nota: Durante todo este capítulo, PPP em modo usuário será simplesmente citado como **ppp** a não ser que uma distinção seja necessária entre este e qualquer outro software PPP como o **pppd**. A menos que indicado de outra maneira, todos os comandos explicados neste capítulo devem ser executados como `root`.

18.2. Utilizando PPP em modo usuário

Atualizado e melhorado por Tom Rhodes. Originalmente contribuído por Brian Somers. Com adições de Nik Clayton, Dirk Frömberg, e Peter Childs.

18.2.1. PPP em modo usuário

18.2.1.1. Suposições

Este documento assume que você tem o seguinte:

- Uma conta com um provedor de acesso a Internet (ISP) através do qual você se conecta utilizando PPP.
- Um modem ou outro dispositivo conectado ao seu sistema e configura corretamente que permite a você se conectar ao seu provedor.
- O número de discagem do seu provedor.
- Seu usuário e senha. (um par UNIX® de usuário e senha ou um par PAP ou CHAP).
- O endereço IP de um ou mais servidores de nomes. Normalmente você receberá dois endereços IP do seu provedor para esta finalidade. Se você não recebeu ao menos um endereço pode utilizar o comando `enable dns` no arquivo `ppp.conf` e o `ppp` irá ajustar os servidores de nome para você. Essa funcionalidade depende do suporte a negociação de DNS que a implementação PPP do seu provedor oferece.

As seguintes informações podem ser providas por seu provedor mas não são totalmente necessárias:

- O endereço IP do *gateway* do seu provedor. O *gateway* é uma máquina a qual você irá se conectar e que irá ser ajustada como sua *rota padrão*. Se você não tem essa informação nos podemos criar um endereço qualquer e o servidor PPP do seu provedor irá nos informar qual é o valor correto quando nós nos conectarmos.

Este endereço IP é referido como *HISADDR* pelo `ppp`.

- A máscara de rede que você deve usar. Se seu provedor não lhe informou um você pode seguramente usar `255.255.255.255`.
- Se seu provedor lhe oferece um endereço IP estático e um hostname, você pode digitá-los. Caso contrário, nós simplesmente deixamos que o servidor remoto defina qualquer endereço IP que ele achar conveniente.

Se você não tem nenhuma dessas informações, entre em contato com seu provedor de acesso.

Nota: Durante toda essa seção muitos dos exemplos que se utilizam do conteúdo de arquivos de configuração são numerados por linha. Esses números servem para ajudar na apresentação e discussão apenas e não devem ser usados em seus arquivos de configuração. Endentação adequada com tab ou espaços também é importante.

18.2.1.2. Criando nós para dispositivos PPP

Em circunstâncias normais a maioria dos usuários irão apenas precisar de um dispositivo `tun` (`/dev/tun0`). Referências ao `tun0` acima podem ser trocadas por `tunN` onde `N` é qualquer número correspondente em seu sistema.

Para instalações do FreeBSD que não tenham [devfs\(5\)](#) ativado (FreeBSD 4.X e anteriores), a existência do dispositivo `tun0` deve ser verificada (isto não é necessário se [devfs\(5\)](#) está ativado pois os nós de dispositivos serão criados sempre que necessário).

A maneira mais fácil de ter certeza que o dispositivo `tun0` está configurado corretamente é refazer o dispositivo. Para refazer o dispositivo faça o seguinte:

```
# cd /dev
# sh MAKEDEV tun0
```

Se você tem 16 dispositivos de túnel no seu kernel, você irá precisar criá-los. Isso pode ser feito executando-se os seguintes comandos:

```
# cd /dev
# sh MAKEDEV tun15
```

18.2.1.3. Configuração automática do PPP

Tanto `ppp` quanto `pppd` (a implementação do PPP em modo kernel) usam os arquivos de configuração localizados no diretório `/etc/ppp`. Exemplos para `ppp` em modo usuário podem ser encontrados em `/usr/share/examples/ppp/`.

A configuração do `ppp` requer que você edite vários arquivos, dependendo dos seus requerimentos. O que você coloca neles depende se seu provedor aloca endereços IP estaticamente (ex. foi dado a você um endereço IP que deve ser usado sempre) ou dinamicamente (ex. seu endereço IP é trocado cada vez que você se conecta).

18.2.1.3.1. PPP e endereços IP estáticos

Você terá de editar o arquivo de configuração `/etc/ppp/ppp.conf`. Ele deve ser similar ao exemplo abaixo.

Nota: Linhas que terminal em um `:` começam na primeira coluna (início da linha)-- todas as outras linhas devem ser endentadas como mostrado utilizando espaços ou tabs.

```
1  default:
2      set log Phase Chat LCP IPCP CCP tun command
3      ident user-ppp VERSION (built COMPILATIONDATE)
4      set device /dev/cuaa0
5      set speed 115200
6      set dial "ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 5 \
7      \\" AT OK-AT-OK ATE1Q0 OK \\dATDT\\T TIMEOUT 40 CONNECT"
8      set timeout 180
9      enable dns
10
11  nome_do_provedor:
```



```
12     set phone "(123) 456 7890"
13     set authname foo
14     set authkey bar
15     set login "TIMEOUT 10 \"\" \"\" gin:--gin: \\U word: \\P col:
ppp"
16     set timeout 300
17     set ifaddr x.x.x.x y.y.y.y 255.255.255.255 0.0.0.0
18     add default HISADDR
```

Linha 1:

Identifica a entrada padrão. Comandos nessa entrada são executados automaticamente quando o ppp é executado.

Linha 2:

Ativa parâmetros de logging. Quando a configuração estiver funcionando de acordo, essa linha deve ser reduzida para algo como

```
set log phase tun
para evitar que o arquivo de log fique com um tamanho excessivo.
```

Linha 3:

Diz ao PPP como se identificar para a ponta remota. PPP se identifica com a ponta remota se ele tem qualquer problema na negociação e no ajuste do *link*, provendo informações que o administrador remoto pode achar útil na investigação de problemas.

Linha 4:

Identifica o dispositivo ao qual o modem está conectado. COM1 é /dev/cuaa0 e COM2 é /dev/cuaa1.

Linha 5:

Define a velocidade que você deseja se conectar. Se 115200 não funcionar (deveria com qualquer modem relativamente novo) tente 38400.

Linhas 6 & 7:

A string de discagem. PPP em modo usuário usa uma *string* similar ao programa [chat\(8\)](#). Veja a página manual para informações sobre as funcionalidades dessa linguagem.

Note que esse comando continua na próxima linha por questões de legibilidade. Qualquer comando no `ppp.conf` pode fazer isso se o último caractere na linha é o caractere `\"`.

Linha 8:

Define o tempo máximo de inatividade do link. 180 segundos é o padrão e essa linha serve apenas para reforçar isso.

Linha 9:

Diz ao PPP para que ele peça a ponta remota que confirme as configurações do *resolver* local. Se você roda um servidor de nomes, essa linha deve ser comentada ou removida.

Linha 10:

Uma linha em branco para legibilidade. Linhas em branco são ignoradas pelo PPP.

Linha 11:

Identifica uma entrada para um provedor chamado ```provedor"`. Isso deve ser trocado pelo nome do seu provedor para que mais tarde você possa utilizar a opção *load ISP* para iniciar a conexão.

Linha 12:

Define o número telefônico para esse provedor. Vários números podem ser especificados usando dois pontos (:) ou o caractere *pipe* (/) como separação. A diferença entre os dois separadores está descrita em [ppp\(8\)](#). Resumindo, se você deseja que todos os números sejam utilizados de forma rotatória utilize os dois pontos. Se você quer que o PPP sempre tente o primeiro número e só utilize os próximos caso o primeiro falhe, utilize o *pipe*. Sempre coloque os números entre aspas.

Você deve colocar o número telefônico entre aspas (") se você tem a intenção de utilizar espaços entre os números. Se você não utilizar aspas quando houverem espaços um erro acontecerá.

Linha 13 & 14:

Identifica o nome do usuário e a senha. Quando se conectar utilizando um prompt de login no estilo UNIX esses valores são utilizados pelo comando `set login` usando as variáveis `\U` e `\P`. Quando se conectar utilizando PAP ou CHAP esses valores são usados no momento da autenticação.

Linha 15:

Se você está utilizando PAP ou CHAP, não haverá nenhum login e essa linha deve ser comentada ou removida. Veja [autenticação PAP e CHAP](#) para maiores detalhes.

A string de login tem a mesma sintaxe que a string de discagem. Nesse exemplo, a string funciona para um serviço que tem uma sessão de login que se parece com isso:

```
J. Random Provider
login: foo
password: bar
protocol: ppp
```

Você precisará alterar esse script para adequá-lo as suas necessidades. Quando você estiver alterando este script pela primeira vez você deve ter certeza de ter ativado o logging de ``chat" para que você consiga determinar se a conversação está se dando como esperado.

Linha 16:

Ajusta o tempo máximo de inatividade (em segundos) para a conexão. No exemplo a conexão será automaticamente fechada após 300 segundos de inatividade. Se você deseja que a conexão nunca seja fechada, defina este valor para zero ou utilize a opção *-ddial* na linha de comando.

Linha 17:

Ajusta o endereço da interface. A string *x.x.x.x* deve ser trocada pelo endereço IP que seu provedor lhe alocou. A string *y.y.y.y* deve ser trocada pelo endereço IP que seu provedor lhe indicou como sendo o seu gateway (a máquina a qual você se conecta). Se seu provedor não lhe forneceu um gateway, utilize *10.0.0.2/0*. Se você precisa ``chutar" um endereço, tenha certeza de que você criou uma entrada em */etc/ppp/ppp.linkup* de acordo com as instruções de [PPP e Endereços Dinâmicos](#). Se essa linha for omitida, o *ppp* não poderá rodar em modo *-auto*.

Linha 18:

Adiciona a rota padrão para o gateway do seu provedor. A palavra especial *HISADDR* é trocada pelo gateway especificado na linha 9. É importante que essa linha apareça após a linha 9, caso contrário a variável *HISADDR* não terá sido inicializada ainda.

Se você não deseja rodar o *ppp* com a opção *-auto*, essa linha deve ser movida para o arquivo *ppp.linkup*.

Não é necessário adicionar uma entrada ao arquivo *ppp.linkup* quando você tem um endereço IP estático e está rodando *ppp* com a opção *-auto* pois sua tabela de roteamento já estará correta antes de você se conectar. Entretanto você pode querer criar uma entrada para executar certos programas após a conexão ter sido estabelecida. Isso será explicado mais tarde com o exemplo do *sendmail*.

Arquivos de configuração de exemplo podem ser encontrados no diretório

/usr/share/examples/ppp/.

18.2.1.3.2. PPP e Endereços IP Dinâmicos

Se seu provedor não definiu nenhum endereço IP estático, o `ppp` pode ser configurado para negociar o endereço local e o remoto. Isso é feito quando você "chuta" um endereço IP qualquer e deixa que o `ppp` ajuste ele de acordo através da utilização do IP Configuration Protocol (IPCP) depois de se conectar. A configuração do `ppp.conf` é a mesma da configuração do [PPP e Endereços IP estáticos](#), com as seguintes mudanças:

```
17      set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.255
```

Novamente, não inclua o número da linha. Ele está lá apenas para referência. Endentação ou pelo menos um espaço é requerido.

Linha 17:

O número após o caractere `/` é o número de bits do endereço que o `ppp` vai insistir que seja utilizado na negociação. Você pode querer utilizar números IP mais adequados a sua situação mas o exemplo acima sempre irá funcionar.

O último argumento (`0.0.0.0`) diz ao PPP para iniciar qualquer negociação utilizando o endereço `0.0.0.0` invés do endereço `10.0.0.1` e é necessário para alguns provedores. Não use `0.0.0.0` como primeiro argumento de `set ifaddr` pois isso não permitirá que o PPP ajuste uma rota inicial quando a opção `-auto` for utilizada na linha de comando.

Se você não está rodando em modo `-auto`, você irá precisar criar uma entrada no `/etc/ppp/ppp.linkup`. Este arquivo é utilizado após a conexão ter sido estabelecida. Nesse momento, o `ppp` irá ter definido os endereços da interface e será agora possível adicionar entradas na tabela de roteamento:

```
1      nome_do_provedor:
2      add default HISADDR
```

Linha 1:

No momento de se estabelecer uma conexão, o `ppp` irá procurar por uma entrada no `ppp.linkup` seguindo as seguintes regras: Primeiro, tente encontrar o mesmo nome que nós utilizamos em `ppp.conf` para a conexão. Se isso falhar, procure por uma entrada para o endereço IP do nosso gateway. Se nós ainda não conseguimos encontrar uma entrada, procure pela entrada `MYADDR`.

Linha 2:

Essa linha diz ao `ppp` para adicionar a rota padrão que aponta para `HISADDR`. `HISADDR` será trocado pelo endereço IP do nosso gateway que foi negociado pelo IPCP.

Veja a entrada `pmdemand` nos arquivos

```
/usr/share/examples/ppp/ppp.conf.sample e
```

```
/usr/share/examples/ppp/ppp.linkup.sample para um exemplo mais detalhado.
```

18.2.1.3.3. Recebendo Chamadas

Quando você configura o **ppp** para receber ligações em uma máquina conectada a uma LAN, você precisa decidir se você deseja que ela passe adiante os pacotes para a LAN. Se você quer isso, você deve alocar para a ponta remota um endereço IP de sua LAN e utilizar o comando `enable proxy no /etc/ppp/ppp.conf`. Você deve também confirmar que o arquivo `/etc/rc.conf` contém a seguinte linha:

```
gateway_enable="YES"
```

18.2.1.3.4. Qual getty?

[Configurando o FreeBSD para Serviços Discados](#) fornece uma boa descrição de como ativar serviços discados usando [getty\(8\)](#).

Uma alternativa para o `getty` é o [mgetty](#), uma versão mais inteligente do `getty` desenvolvida para linhas discadas.

A vantagem de se utilizar o `mgetty` é que ele ativamente *fala* com os modems. Isso significa que se a porta está desligada no `/etc/ttys` então seu modem não irá atender o telefone.

Versões mais recentes do `mgetty` (de 0.99beta para frente) também suportam a detecção automática de conexões PPP, permitindo a seus clientes que se conectem ao seu servidor sem a utilização de scripts.

Veja [Mgetty e Auto-PPP](#) para maiores informações sobre o `mgetty`.

18.2.1.3.5. Permissões do PPP

O comando `ppp` precisa normalmente ser executado como `root`. Se entretanto, você deseja permitir que ele seja executado no modo servidor por um usuário normal, este usuário precisa que lhe seja garantida a permissão de execução para o `ppp` através da adição deste usuário ao grupo `network` no arquivo `/etc/group`.

Você também precisará dar acesso a uma ou mais seções do arquivo de configuração usando o comando `allow`:

```
allow users fred mary
```

Se o comando for utilizado na seção `default`, ele garante que os usuários especificados terão acesso a tudo.

18.2.1.3.6. Shells PPP e Usuários de IPs Dinâmicos

Crie um arquivo chamado `/etc/ppp/ppp-shell` contendo o seguinte:

```
#!/bin/sh
IDENT=`echo $0 | sed -e 's/^.*-\\(.*\\)$/\\1/'`
CALLEDAS="$IDENT"
TTY=`tty`
```

```

if [ x$IDENT = xdialup ]; then
    IDENT=`basename $TTY`
fi

echo "PPP for $CALLEDAS on $TTY"
echo "Starting PPP for $IDENT"

exec /usr/sbin/ppp -direct $IDENT

```

Esse script deve ser executável. Agora crie um link simbólico chamado `ppp-dialup` para este script usando o seguinte comando:

```
# ln -s ppp-shell /etc/ppp/ppp-dialup
```

Você deve usar este script como a *shell* para todos os seus usuários discados. Este é um exemplo do `/etc/passwd` para um usuário de discado chamado `pchilds` (lembre-se de não editar o arquivo de senhas diretamente, utilize o `vipw`).

```
pchilds:*:1011:300:Peter Childs PPP:/home/ppp:/etc/ppp/ppp-dialup
```

Crie um diretório chamado `/home/ppp` que possa ser lido por qualquer usuário contendo os seguintes arquivos vazios:

```

-r--r--r--  1 root    wheel    0 May 27 02:23 .hushlogin
-r--r--r--  1 root    wheel    0 May 27 02:22 .rhosts

```

que farão com que o `/etc/motd` não seja mostrado.

18.2.1.3.7. Shells PPP para Usuários de IPs Estáticos

Crie o arquivo `ppp-shell` como mostrado acima e para cada conta que tenha um IP estático definido crie um link simbólico para `ppp-shell`.

Por exemplo, se você tem três clientes discados, `fred`, `sam`, e `mary`, que você faz roteamento de redes classe C para eles, você deve digitar o seguinte:

```

# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-fred
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-sam
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-mary

```

Cada um desses usuários deve ter sua shell definida para o link simbólico criado acima (por exemplo, a shell do usuário `mary` deve ser `/etc/ppp/ppp-mary`).

18.2.1.3.8. Ajustando o `ppp.conf` para Usuários de IPs Dinâmicos

O arquivo `/etc/ppp/ppp.conf` deve conter algo parecido com o seguinte:

```

default:
    set debug phase lcp chat
    set timeout 0

```

```
ttyd0:
```

```
set ifaddr 203.14.100.1 203.14.100.20 255.255.255.255
enable proxy
```

```
ttyd1:
set ifaddr 203.14.100.1 203.14.100.21 255.255.255.255
enable proxy
```

Nota: A endentação é importante.

A seção *default*: é carregada para cada sessão. Para cada linha discada ativada no */etc/ttys* você deve criar uma entrada similar a *ttyd0*: abaixo. Cada linha deve ter um endereço IP único dentre os IPs disponíveis para os usuários dinâmicos.

18.2.1.3.9. Ajustando o *ppp.conf* para Usuários de IPs Estáticos

Junto com o conteúdo do arquivo de exemplo */usr/share/examples/ppp/ppp.conf* acima você deve adicionar uma seção para cada usuário estático. Nós iremos continuar com os usuários *fred*, *sam* e *mary* como exemplo.

```
fred:
set ifaddr 203.14.100.1 203.14.101.1 255.255.255.255

sam:
set ifaddr 203.14.100.1 203.14.102.1 255.255.255.255

mary:
set ifaddr 203.14.100.1 203.14.103.1 255.255.255.255
```

O arquivo */etc/ppp/ppp.linkup* deve também conter informações de roteamento para cada endereço IP estático se isso for necessário. A linha abaixo irá adicionar uma rota para a rede classe C 203.14.101.0 através do link PPP do cliente.

```
fred:
add 203.14.101.0 netmask 255.255.255.0 HISADDR

sam:
add 203.14.102.0 netmask 255.255.255.0 HISADDR

mary:
add 203.14.103.0 netmask 255.255.255.0 HISADDR
```

18.2.1.3.10. *mgetty* e *AutoPPP*

Configurar e compilar o *mgetty* com a opção *AUTO_PPP* ativada permite que o *mgetty* detecte a fase LCP da conexão PPP e automaticamente inicie uma shell ppp. Entretanto, já que a seqüência de usuário/senha padrão não ocorre é necessário autenticar os usuários usando PAP ou CHAP.

Essa seção assume que o usuário tenha configurado, compilado e instalado uma versão do *mgetty* com a opção *AUTO_PPP* (v0.99beta ou superior).

Tenha certeza de que o arquivo */usr/local/etc/mgetty+sendfax/login.config* contenha o seguinte:

```
/AutoPPP/ - - /etc/ppp/ppp-pap-dialup
```

Isso irá fazer com que o `mgetty` rode o script `ppp-pap-dialup` para as conexões PPP que ele detectar.

Crie um arquivo chamado `/etc/ppp/ppp-pap-dialup` contendo o seguinte (esse arquivo deve ser executável):

```
#!/bin/sh
exec /usr/sbin/ppp -direct pap$IDENT
```

Para cada linha discada ativado em `/etc/ttys`, crie uma entrada correspondente em `/etc/ppp/ppp.conf`. Isso irá coexistir normalmente com nossas definições acima.

```
pap:
  enable pap
  set ifaddr 203.14.100.1 203.14.100.20-203.14.100.40
  enable proxy
```

Cada usuário se conectando com esse método precisará de um usuário/senha no arquivo `/etc/ppp/ppp.secret` ou você pode adicionar a seguinte opção para autenticar os usuários através do arquivo `/etc/password`.

```
enable passwdauth
```

Se você deseja definir um endereço IP estático para alguns usuário você pode especificar o endereço como o terceiro argumento no `/etc/ppp/ppp.secret`. Veja o `/usr/share/examples/ppp/ppp.secret.sample` para mais exemplos.

18.2.1.3.11. Extensões MS

É possível configurar o PPP para fornecer endereços de DNS e NetBIOS de acordo com a demanda.

Para ativar essas extensões na versão 1.x do PPP, as seguintes linhas podem ser adicionadas a seção relevante do `/etc/ppp/ppp.conf`.

```
enable msextns
set ns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

E para o PPP versão 2 ou superior:

```
accept dns
set dns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

Isso irá fornecer aos clientes o endereço do servidor primário e secundário de DNS e o servidor de NetBIOS.

Na versão 2 ou superior, se a linha `set dns` for omitida, o PPP irá usar os valores encontrados no `/etc/resolv.conf`.

18.2.1.3.12. Autenticação PAP e CHAP

Alguns provedores definem seus sistemas de forma que a parte de autenticação da sua conexão é feita usando os mecanismos de autenticação PAP ou CHAP. Se esse for o caso, seu provedor não irá lhe dar um prompt `login:` quando você se conectar, mas sim irá iniciar a comunicação PPP imediatamente.

PAP é menos seguro que CHAP, mas segurança não é normalmente uma preocupação aqui já que as senhas, mesmo sendo enviadas em texto puro com PAP, estão sendo transmitidas através da linha serial apenas. Não existe muitas possibilidades para que crackers vejam ela.

Utilizando o que vimos nas seções [PPP e Endereços IP Estáticos](#) ou [PPP e Endereços IP Dinâmicos](#), as seguintes alterações devem ser feitas:

```
7      set login
...
12     set authname MeuUsuario
13     set authkey MinhaSenha
```

Linha 7:

Seu provedor não irá, normalmente, requerer que você entre no servidor se estiver usando PAP ou CHAP. Você deve desabilitar a string `set login` neste caso.

Linha 12:

Essa linha especifica seu nome de usuário PAP/CHAP. Você precisará inserir o valor correto para *MeuUsuario*.

Linha 13:

Essa linha especifica sua senha PAP/CHAP. Você precisará inserir o valor correto para *MinhaSenha*. Você pode adicionar uma linha adicional como abaixo:

```
15     accept PAP
```

ou

```
15     accept CHAP
```

para deixar claro que esta é sua intenção, mas PAP e CHAP são todos aceitos por padrão.

18.2.1.3.13. Mudando a Configuração do PPP "on the Fly"

É possível conversar com o programa `ppp` enquanto ele está executando em segundo plano, mas apenas se uma porta de diagnósticos for configurada. Para fazer isso, adicione a seguinte linha a sua configuração:

```
set server /var/run/ppp-tun%d DiagnosticPassword 0177
```

Isso fará com que o PPP ouça no socket UNIX especificado, perguntando aos clientes a senha antes de permitir o acesso. O *%d* no nome é trocado pelo número do dispositivo tun que está em uso.

Uma vez que um socket foi ajustado, o programa [pppctl\(8\)](#) pode ser utilizado em scripts que desejem manipular o programa em tempo de execução.

18.2.1.4. Usando o Recurso de NAT do PPP

O PPP tem a habilidade de fazer NAT (Network Address Translation) sem precisar utilizar as capacidades de *divert* do kernel. Essa funcionalidade pode ser ativada através da seguinte linha no `/etc/ppp/ppp.conf`:

```
nat enable yes
```

Outra maneira de ativar o NAT do PPP é pela linha de comando `-nat`. Existe também uma opção no `/etc/rc.conf` chamada `ppp_nat` que é ativada por padrão.

Se você for usar essa funcionalidade, você também pode achar útil as seguintes opções em `/etc/ppp/ppp.conf` para ativar o encaminhamento de conexões entrantes:

```
nat port tcp 10.0.0.2:ftp ftp
nat port tcp 10.0.0.2:http http
```

ou não confie no exterior

```
nat deny_incoming yes
```

18.2.1.5. Configuração Final do Sistema

Você agora tem o `ppp` configurado, mas existem mais algumas coisas a serem feitas antes que o show comece. Todas elas envolvem editar o arquivo `/etc/rc.conf`.

Certifique-se de que a linha `hostname=` está definida, ex.:

```
hostname="foo.example.com"
```

Se seu provedor lhe forneceu um endereço IP estático e um nome, é melhor que você use este nome como seu `hostname`.

Veja a variável `network_interfaces`. Se você quer configurar seu sistema de modo que ele disque para seu provedor conforme necessário, tenha certeza de que o dispositivo `tun0` está adicionado a lista, do contrário remova-o.

```
network_interfaces="lo0 tun0"
ifconfig_tun0=
```

Nota: A variável `ifconfig_tun0` deve estar vazia, e um arquivo chamado `/etc/start_if.tun0` deve ser criado. Esse arquivo deve conter a linha:

```
ppp -auto nome_do_provedor
```

Esse script é executado no momento da configuração de rede, iniciando seu daemon ppp em modo automático. Se você tem uma rede local para a qual esta máquina é o gateway, você pode desejar usar a opção *-alias*. Veja a página manual para maiores detalhes.

Define o programa roteador para *NO* com a seguinte linha no seu `/etc/rc.conf`:

```
router_enable="NO"
```

É importante que o daemon `routed` não seja iniciado (ele é iniciado por padrão), pois o `routed` tende a deletar as entradas na tabela de roteamento que foram criadas pelo `ppp`.

É importante ter certeza de que a variável `sendmail_flags` não contenha a opção *-q*, do contrário o `sendmail` irá tentar fazer uma pesquisa na rede em certos momentos, o que possivelmente irá fazer com que sua máquina disque e tente estabelecer uma conexão de rede. Você pode tentar:

```
sendmail_flags="-bd"
```

A desvantagem disso é que você precisa forçar o `sendmail` a re-examinar sua fila de correio sempre que o link `ppp` for estabelecido digitando na linha de comando o seguinte:

```
# /usr/sbin/sendmail -q
```

Se você deseja utilizar o comando `!bg` no arquivo `ppp.linkup` para fazer isso automaticamente:

```
1 nome_do_provedor:
2 delete ALL
3 add 0 0 HISADDR
4 !bg sendmail -bd -q30m
```

Se você não gosta disso, é possível ajustar um `dfilter` para bloquear o tráfego SMTP. Veja os arquivos de exemplo para maiores detalhes.

Agora o último passo é reiniciar a máquina. Depois disso, você pode digitar:

```
# ppp
```

e então `dial nome_do_provedor` para iniciar uma sessão PPP, ou, se você quer que o `ppp` estabeleça as sessões automaticamente quando existir tráfego de saída (e você não criou o script `start_if.tun0`, digite:

```
# ppp -auto nome_do_provedor
```

18.2.1.6. Sumário

Recapitulando, os seguintes passos são necessários para ajustar o `ppp` pela primeira vez:

No cliente:

1. Certificar-se de que o dispositivo `tun` está definido no seu kernel.
2. Certificar-se de que o dispositivo `tunN` está disponível no diretório `/dev`.
3. Criar uma entrada no `/etc/ppp/ppp.conf`. O arquivo de exemplo `ppmdemand` deve ser suficiente para a maioria dos provedores.
4. Se você tem um endereço IP dinâmico, crie uma entrada no arquivo `/etc/ppp/ppp.linkup`.
5. Atualize seu arquivo `/etc/rc.conf`.
6. Crie o script `start_if.tun0` se você precisa que o sistema ligue de acordo com a demanda.

No Servidor:

1. Certificar-se de que o dispositivo `tun` está no seu kernel.
2. Certificar-se de que o dispositivo `tunN` está disponível no diretório `/dev`.
3. Criar uma entrada no `/etc/passwd` (usando o programa [vipw\(8\)](#)).
4. Crie um perfil nesse diretório de usuários que rode o comando `ppp -direct direct-server` ou similar.
5. Crie uma entrada no arquivo `/etc/ppp/ppp.conf`. O arquivo de exemplo `direct-server` deve ser suficiente.
6. Crie uma entrada no `/etc/ppp/ppp.linkup`.
7. Atualize seu arquivo `/etc/rc.conf`.

18.3. Usando PPP em modo kernel

Partes originalmente contribuídas por Gennady B. Sorokopud e Robert Huff.

18.3.1. PPP em modo kernel

Antes que você inicie o ajuste do PPP na sua máquina, certifique-se de que o `pppd` está no diretório `/usr/sbin` e que o diretório `/etc/ppp` existe.

O `pppd` pode funcionar de dois modos:

1. Como um "cliente" -- você quer conectar sua máquina com o mundo exterior através de uma conexão PPP serial ou de um de modem.
2. Como um "servidor" -- sua máquina está localizada em uma rede e é utilizada para conectar outros computadores através de PPP.

Nos dois casos você deverá ajustar o arquivo de opções (`/etc/ppp/options` ou `~/ppp.rc` se você tem mais de um usuário na máquina que utiliza PPP).

Você irá também precisar de um software para modem/serial (preferencialmente [comms/kermit](#)), para que você possa discar e estabelecer uma conexão com a máquina remota.

18.3.2. Usando o `pppd` como Cliente

Baseado em informações providas por Trev Roydhouse.

O arquivo `/etc/ppp/options` abaixo pode ser utilizado para se conectar a um servidor PPP da Cisco.

```
crtscts      # ativa controle de fluxo por hardware
modem        # linha de controle do modem
noipdefault  # o servidor PPP remoto deve definir um endereço IP para
você
              # se a máquina remota não definir o endereço durante a
              # negociação IPCP, remova esta opção
passive      # espera por pacotes LCP
domain ppp.foo.com      # coloque seu domínio aqui

:<remote_ip>  # ponha o endereço da máquina remota aqui
              # isso será utilizado para rotear os pacotes pelo link PPP
              # se você não especificou a opção noipdefault
              # mude essa linha para <local_ip>:<remote_ip>

defaultroute # coloque isso se você deseja que o servidor PPP seja
              # o seu roteador padrão
```

Para conectar:

1. Disque para o *host* remoto usando o **kermit** (ou outro programa para o modem) e digite seu usuário e senha (ou o que quer que seja necessário para ativar PPP no *host* remoto).
2. Saia do **kermit** (sem desconectar a linha).
3. Digite o seguinte:
4. `# /usr/src/usr.sbin/pppd.new/pppd /dev/tty01 19200`

Certifique-se de usar a velocidade e o nome do dispositivo apropriados.

Agora seu computador está conectado por PPP. Se a conexão falhar, você pode adicionar a opção *debug* ao arquivo `/etc/ppp/options` e verificar as mensagens no *console* para diagnosticar o problema.

O script `/etc/ppp/pppup` abaixo irá fazer os 3 estágios automaticamente:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing pppd, PID=' ${pid}
    kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing kermit, PID=' ${pid}
    kill -9 ${pid}
fi

ifconfig ppp0 down
ifconfig ppp0 delete
```

```
kermit -y /etc/ppp/kermit.dial
pppd /dev/tty01 19200
```

/etc/ppp/kermit.dial é um script do **kermit** que disca e faz toda a autorização necessária no *host* remoto (um exemplo de tal script está anexado ao final deste documento).

Use o seguinte script /etc/ppp/pppdown para desconectar a linha PPP:

```
#!/bin/sh
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ X${pid} != "X" ] ; then
    echo 'killing pppd, PID=' ${pid}
    kill -TERM ${pid}
fi

ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing kermit, PID=' ${pid}
    kill -9 ${pid}
fi

/sbin/ifconfig ppp0 down
/sbin/ifconfig ppp0 delete
kermit -y /etc/ppp/kermit.hup
/etc/ppp/ppptest
```

Verifique se o pppd está ainda executando através do comando /usr/etc/ppp/ppptest, que deverá se parecer com o seguinte:

```
#!/bin/sh
pid=`ps ax| grep pppd |grep -v grep|awk '{print $1;}'`
if [ X${pid} != "X" ] ; then
    echo 'pppd running: PID=' ${pid-NONE}
else
    echo 'No pppd running.'
fi
set -x
netstat -n -I ppp0
ifconfig ppp0
```

Para desligar, execute o /etc/ppp/kermit.hup, que deve conter:

```
set line /dev/tty01      ; put your modem device here
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
```

```
out +++
inp 5 OK
out ATH0\13
echo \13
exit
```

Um método alternativo é usar o comando `chat` invés do `kermit`:

Os dois arquivos seguintes são suficientes para estabelecer uma conexão com `pppd`.

`/etc/ppp/options`:

`/dev/cuaa1 115200`

```
crtscts      # ativa controle de fluxo por hardware
modem        # linha de controle do modem
connect      "/usr/bin/chat -f /etc/ppp/login.chat.script"
noipdefault  # o servidor PPP remoto deve definir um endereço IP
para você
              # se a máquina remota não definir o endereço durante a
              # negociação IPCP, remova esta opção
passive      # espera por pacotes LCP
domain <your.domain> # coloque seu domínio aqui
              # ponha o endereço da máquina remota aqui
              # isso será utilizado para rotear os pacotes pelo link
PPP
              # se você não especificou a opção noipdefault
              # mude essa linha para <local_ip>:<remote_ip>
defaultroute # coloque isso se você deseja que o servidor PPP seja
              # o seu roteador padrão
```

`/etc/ppp/login.chat.script`:

Nota: O conteúdo abaixo deve ficar em uma única linha:

```
ABORT BUSY ABORT 'NO CARRIER' "" AT OK ATDT<phone.number>
CONNECT "" TIMEOUT 10 ogin:-\r-ogin: <login-id>
TIMEOUT 5 sword: <password>
```

Uma vez que eles estiverem instalados e modificados corretamente, tudo que você precisa fazer é executar o `pppd` da seguinte maneira:

```
# pppd
```

18.3.3. Usando o `pppd` como Servidor

O arquivo `/etc/ppp/options` deve conter algo similar ao seguinte:

```
crtscts      # Controle de fluxo por hardware
netmask 255.255.255.0 # máscara de rede (não requerido)
192.114.208.20:192.114.208.165 # IP's da máquina remota e local
              # IP local pode ser diferente do qual
              # você definiu para a interface ethernet
              # na sua máquina.
              # IP remoto é o endereço IP que será definido
```

```

                # para a máquina remota.
domain ppp.foo.com      # seu domínio
passive                # esperar por LCP
modem                  # linha do modem

```

O script `/etc/ppp/pppserv` abaixo irá fazer com que o **pppd** se comporte como um servidor:

```

#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing pppd, PID=' ${pid}
    kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing kermit, PID=' ${pid}
    kill -9 ${pid}
fi

# reset ppp interface
ifconfig ppp0 down
ifconfig ppp0 delete

# enable autoanswer mode
kermit -y /etc/ppp/kermit.ans

# run ppp
pppd /dev/tty01 19200

```

Use o script `/etc/ppp/pppservdown` para para o servidor:

```

#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing pppd, PID=' ${pid}
    kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing kermit, PID=' ${pid}
    kill -9 ${pid}
fi
ifconfig ppp0 down
ifconfig ppp0 delete

kermit -y /etc/ppp/kermit.noans

```

O script para **kermit** abaixo (`/etc/ppp/kermit.ans`) irá ativar/desativar o modo de atendimento automático do seu modem. Ele deve se parecer com o seguinte:

```

set line /dev/tty01
set speed 19200
set file type binary
set file names literal

```



```

set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATH0\13
inp 5 OK
echo \13
out ATS0=1\13 ; mude isso para ATS0=0\13 se você quer desabilitar
                ; o modo de atendimento automático.
inp 5 OK
echo \13
exit

```

O script chamado `/etc/ppp/kermit.dial` é utilizado para discagem e autenticação na máquina remota. Você precisará customizá-lo para suas necessidades. Coloque seu usuário e senha nesse script; você também precisará mudar a indicação de entrada dependendo das respostas do seu modem e da máquina remota.

```

;
; ponha a linha serial a qual o modem está ligado:
;
set line /dev/tty01
;
; velocidade do modem:
;
set speed 19200
set file type binary ; full 8 bit file xfer
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none
set modem hayes
set dial hangup off
set carrier auto ; Then SET CARRIER if necessary,
set dial display on ; Then SET DIAL if necessary,
set input echo on
set input timeout proceed
set input case ignore
def \%x 0 ; login prompt counter
goto slhup

:slcmd ; Poe o modem em modo de comando
echo Put the modem in command mode.
clear ; Limpa caracteres não lidos do buffer de entrada
pause 1
output +++ ; hayes escape sequence
input 1 OK\13\10 ; wait for OK
if success goto slhup
output \13

```

```

pause 1
output at\13
input 1 OK\13\10
if fail goto slcmd          ; if modem doesn't answer OK, try again

:slhup                      ; desliga o telefone
clear                      ; Limpa caracteres não lidos do buffer de entrada
pause 1
echo Hanging up the phone.
output ath0\13             ; hayes command for on hook
input 2 OK\13\10
if fail goto slcmd        ; se não houve resposta OK, poe o modem em
modo de comando

:sldial                    ; disca o número
pause 1
echo Dialing.
output atdt9,550311\13\10 ; ponha o número do telefone aqui
assign \%x 0              ; zera o contador de tempo

:look
clear                      ; Limpa caracteres não lidos do buffer de entrada
increment \%x             ; Conta os segundos
input 1 {CONNECT }
if success goto sllogin
reinput 1 {NO CARRIER\13\10}
if success goto sldial
reinput 1 {NO DIALTONE\13\10}
if success goto slnodial
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}
if success goto slhup
if < \%x 60 goto look
else goto slhup

:sllogin                   ; login
assign \%x 0              ; Zera o contador de tempo
pause 1
echo Looking for login prompt.

:slloop
increment \%x            ; Conta os segundos
clear                   ; Limpa caracteres não lidos do buffer de entrada
output \13
;
; ponha o prompt de login esperado aqui:
;
input 1 {Username: }
if success goto sluid
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}
if success goto slhup
if < \%x 10 goto slloop ; tenta 10 vezes conseguir o prompt de login
else goto slhup        ; desliga e tenta denovo se falhou 10 vezes

:sluid
;
; ponha seu usuário aqui:
;

```

```
output ppp-login\13
input 1 {Password: }
;
; ponha sua senha aqui:
;
output ppp-password\13
input 1 {Entering SLIP mode.}
echo
quit

:slnodial
echo \7No dialtone. Check the telephone line!\7
exit 1

; local variables:
; mode: csh
; comment-start: ";"
; comment-start-skip: ";"
; end:
```

18.4. Resolução de Problemas

Contribuído por Tom Rhodes.

Esta seção cobre alguns dos problemas que podem aparecer quando você utiliza PPP sobre uma conexão com modem. Por exemplo, talvez você precise saber exatamente qual o *prompt* que o sistema remoto irá lhe apresentar. Alguns provedores usam o prompt *ssword* e outros usam *password*; se o script `ppp` não for escrito de acordo a autenticação irá falhar. A maneira mais fácil de diagnosticar problemas nas conexões `ppp` e tentar se conectar manualmente. A seguinte informação vai lhe ajudar a realizar este processo passo a passo.

18.4.1. Verificar os Nós de Dispositivos

Se você reconfigurou seu kernel você se lembra do dispositivo `sio`. Se você não reconfigurou seu kernel não precisa se preocupar. Apenas verifique a saída do comando `dmesg` e procure pelo dispositivo do modem dessa forma:

```
#dmesg | grep sio
```

Você deve ver alguma informações sobre dispositivos `sio`. Essas são as portas COM que nós precisamos. Se seu modem age como uma porta serial padrão então você deve ele listado como `sio1`, ou COM2. Neste caso você não precisa recompilar seu kernel, você precisa apenas criar o dispositivo serial. Você pode fazer isso indo em `/dev` e rodando o script `MAKEDEV` como acima. Agora crie os dispositivos seriais com:

```
# sh MAKEDEV cuaa0 cuaa1 cuaa2 cuaa3
```

que irá criar os dispositivos seriais para seu sistema. Se seu modem estiver em `sio1` ou COM2 se você está no DOS, então este modem será `/dev/cuaa1`.

18.4.2. Conectando Manualmente

Conectar a Internet através do controle manual do `ppp` é rápido, fácil e uma boa forma de depurar uma conexão ou apenas ver como seu provedor gerencia conexões de clientes `ppp`. Vamos iniciar o **PPP** pela linha de comando. Note que todos os nossos exemplos irão utilizar *exemplo* como o `hostname` da máquina rodando o **PPP**. Você inicia o `ppp` digitando apenas `ppp`:

```
# ppp
```

Nós iniciamos o `ppp`.

```
ppp ON exemplo> set device /dev/cuaa1
```

Ajustamos o dispositivo do modem, nesse caso `cuaa1`.

```
ppp ON exemplo> set speed 115200
```

Ajusta a velocidade de conexão, nesse caso estamos utilizando 115,200 kbps.

```
ppp ON exemplo> enable dns
```

Dizemos ao `ppp` que configure nosso *resolver* e adicione as linhas *nameserver* ao `/etc/resolv.conf`. Se o `ppp` não puder determinar seu *hostname* nós podemos ajustá-lo manualmente mais tarde.

```
ppp ON exemplo> term
```

Mude para o modo `terminal` para que possamos controlar o modem manualmente.

```
deflink: Entering terminal mode on /dev/cuaa1
type '~h' for help
at
    OK
    atdt123456789
```

Utilize `at` para inicializar o modem, e então `atdt` e o número do seu provedor para iniciar o processo de discagem.

```
CONNECT
```

Sinal de confirmação da conexão. Se nós formos ter qualquer problema, não relacionado ao hardware, é aqui que tentaremos resolvê-lo.

```
ISP Login:usuário
```

Aqui um usuário é requerido. Digite o usuário que lhe foi informado pelo provedor.

```
ISP Pass:senha
```

Agora é a vez da senha, apenas responda com a senha que o provedor lhe informou. Da mesma maneira como entramos no sistema, a senha não será mostrada aqui.

```
Shell or PPP:ppp
```

Dependendo do seu provedor este prompt pode não aparecer. Aqui estamos sendo indagados se nós gostaríamos de usar uma *shell* no provedor ou iniciar o `ppp`. Neste exemplo nós escolhemos `ppp` pois queremos uma conexão com a Internet.

```
Ppp ON exemplo>
```

Note que neste exemplo o primeiro *p* está maiúsculo agora. Isso mostra que nós nos conectamos com sucesso ao provedor.

```
PPP ON exemplo>
```

Nós nos autenticamos com sucesso junto ao provedor e estamos esperando o endereço IP que vai ser fornecido.

```
PPP ON exemplo>
```

Nós fizemos um acordo sobre o endereço IP e completamos com sucesso nossa conexão.

```
PPP ON exemplo>add default HISADDR
```

Aqui nós adicionar a rota padrão. Precisamos fazer isso antes de conversar com o mundo exterior pois no momento a única conexão estabelecida é com a ponta remota. Se isso falhar devido a já haver uma rota existente você pode colocar o caractere *!* na frente da opção *add*. Você também pode fazer isso antes de iniciar a conexão e o programa irá negociar uma rota de acordo.

Se tudo correu bem nós devemos ter uma conexão ativa com a Internet agora, que deve ser jogada para o segundo plano usando **CTRL+Z**. Se você notar que o `PPP` retornou para `ppp` então nós perdemos nossa conexão. É bom saber isso pois ele lhe informa o estado atual da conexão. P's maiúsculos indicam que nós temos uma conexão com o provedor e p's minúsculos indicam que a conexão foi perdido por um motivo qualquer. O comando `ppp` tem apenas dois estados.

18.4.2.1. Depuração

Se você tem uma linha direta e não consegue realizar a conexão tente desativar controle de fluxo por hardware CTS/RTS com a opção *set ctsrts off*. Isso pode ocorrer caso você esteja conectado a um servidor PPP onde o **PPP** trava quando ele tenta enviar os dados pelo seu *link* de comunicação. Ele estaria esperando por um sinal CTS, ou Clear To Send (Livre para Enviar), que pode nunca chegar. Entretanto se você ativar esta opção você deve também usar *set accmap* que pode ser requerido para cancelar o hardware que depende do envio de certos caracteres de ponta a ponta, na maior parte do tempo XON/XOFF. Veja a página manual [ppp\(8\)](#) para maiores informações sobre esta opção e como ela é utilizada.

Se você tem um modem antigo pode ter que utilizar a opção *set parity even*. Paridade é ajustada para *none* por padrão, mas é utilizada para verificação de erros (com um

aumento no tráfego) em modems antigos e alguns provedores. Você pode precisar desta opção no provedor Compuserve.

O **PPP** pode não retornar para o modo de comando, o que geralmente indica um erro de negociação onde o provedor está aguardando que você inicie a negociação. Nesse ponto, usar o comando `~P` irá forçar o ppp a enviar as informações de configuração.

Se você nunca ver um prompt de login você terá de utilizar PAP ou CHAP para fazer a autenticação em vez do estilo UNIX® no exemplo acima. Para usar PAP ou CHAP apenas adicione as seguintes opções no **PPP** antes de entrar em modo terminal:

```
ppp ON exemplo> set authname usuario
```

Onde *usuario* deve ser trocado pelo usuário que foi definido pelo seu provedor.

```
ppp ON exemplo> set authkey senha
```

Onde *senha* deve ser trocado pela senha que foi definida pelo seu provedor.

Se você conectar sem problemas mas não conseguir achar nenhum domínio, tente utilizar o [ping\(8\)](#) com um endereço IP e ver se você consegue alguma informação. Se você experimentar 100% de perda de pacotes (*packet loss*) então poderá indicar que uma rota padrão não foi definida. Verifique novamente que a opção *add default HISADDR* foi definida durante a conexão. Se você conseguir conectar a um endereço IP remoto então é possível que um servidor de nomes não foi adicionado ao `/etc/resolv.conf`. Esse arquivo deve se parecer com:

```
domain exemplo.com.br
nameserver x.x.x.x
nameserver y.y.y.y
```

Onde *x.x.x.x* e *y.y.y.y* devem ser trocados pelos endereços IP dos servidores de DNS do seu provedor. Essa informação pode ser definida pelo seu provedor, ou não. Ligue para seu provedor e peça que lhe informem estes endereços.

Você pode fazer com que o [syslog\(3\)](#) ative a função de *logging* para sua conexão **PPP**

```
!ppp
*.* /var/log/ppp.log
```

no `/etc/syslog.conf`. Na maioria dos casos esta funcionalidade já existe.

18.5. Usando PPP sobre Ethernet (PPPoE)

Contribuído (de <http://node.to/freebsd/how-tos/how-to-freebsd-pppoe.html>) por Jim Mock.

Esta seção descreve como configurar o PPP sobre Ethernet (PPPoE).

18.5.1. Configurando o Kernel

Não é necessária nenhuma configuração no kernel para o PPPoE. Se o suporte a netgraph não estiver compilado no kernel ele será carregado dinamicamente pelo **ppp**.

18.5.2. Ajustando o `ppp.conf`

Abaixo temos um exemplo de um `ppp.conf` funcional:

```
default:
    set log Phase tun command # você pode adicionar mais logging se
    quiser
    set ifaddr 10.0.0.1/0 10.0.0.2/0

nome_do_provedor:
    set device PPPoE:x11 # troque x11 pelo seu dispositivo ethernet
    set authname usuario
    set authkey senha
    set dial
    set login
    add default HISADDR
```

18.5.3. Executando o `ppp`

Como root, você pode executar:

```
# ppp -ddial nome_do_provedor
```

18.5.4. Iniciando o `ppp` na Inicialização

Adicione o seguinte ao seu arquivo `/etc/rc.conf`:

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_nat="YES" # se você quer habilitar NAT para sua rede, ou não.
ppp_profile="nome_do_provedor"
```

18.5.5. Usando uma Service Tag no PPPoE

Algumas vezes será necessário usar uma *service tag* para estabelecer sua conexão. Service tags são usadas para distinguir entre diferentes serviços de PPPoE em uma dada rede.

Você deve ter recebido qualquer informação que é requerida do seu provedor para service tags. Se você não tiver, pergunte a equipe técnica do seu provedor.

Como uma último recurso você pode tentar o método sugerido pelo programa [Roaring Penguin PPPoE](#) que pode ser localizado na coleção de [ports](#). Tenha em mente que isso pode desprogramar seu modem e deixá-lo inutilizável. Portanto pense duas vezes antes de fazer isto. Simplesmente instale o programa fornecido com seu modem pelo provedor. Então acesse o menu System pelo programa. O nome do seu perfil deve estar listado. Ele geralmente é *provedor*.

O perfil (service tag) será utilizado na configuração do PPPoE no `ppp.conf` como a parte do provedor no comando `set device` (veja [ppp\(8\)](#) para maiores detalhes). Ele deve se parecer com:

```
set device PPPoE:x11:ISP
```

Não esqueça de mudar `x11` pelo dispositivo correto para sua placa de rede Ethernet.

Não esqueça de trocar *provedor* pelo perfil que você acabou de achar.

Para mais informações veja:

- [Cheaper Broadband with FreeBSD on DSL](#) por Renaud Waldura.
- [Nutzung von T-DSL und T-Online mit FreeBSD](#) por Udo Erdelhoff (in German).

18.5.6. PPPoE com um 3Com® HomeConnect® ADSL Modem Dual Link

Este modem não segue o [RFC 2516](#) (*A Method for transmitting PPP over Ethernet (PPPoE)*), escrito por L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, and R. Wheeler). Em vez disso, códigos diferentes para os pacotes são utilizados em vez de quadros Ethernet. Por favor reclame com a [3Com](#) se você acha que ele deveria seguir a especificação PPPoE.

Para que o FreeBSD seja capaz de se comunicar com este dispositivo uma `sysctl` precisa ser ajustada. Isso pode ser feito automaticamente na inicialização do sistema através da atualização do arquivo de configuração `/etc/sysctl.conf`:

```
net.graph.nonstandard_pppoe=1
```

ou pode ser utilizado o comando `sysctl net.graph.nonstandard_pppoe=1` para um efeito imediato.

Infelizmente essa modificação tem efeito no sistema inteiro e não é possível conversar com um servidor PPPoE normal e com um 3Com® HomeConnect® ADSL Modem ao mesmo tempo.

18.6. Usando PPP sobre ATM (PPPoA)

O seguinte descreve como ajustar o PPP sobre ATM (PPPoA). PPPoA é uma escolha popular entre provedores DSL da Europa.

18.6.1. Usando PPPoA com o Alcatel SpeedTouch™ USB

Suporte PPPoA para este dispositivo é fornecido como um port no FreeBSD pois o *firmware* é fornecido através de um [acordo de licenciamento](#) com a Alcatel e não pode ser distribuído livremente com o sistema base do FreeBSD.

Para instalar o programa simplesmente use a coleção de [ports](#). Instale o port [net/pppoa](#) e siga as instruções fornecidas.

Como muitos dispositivos USB, o Alcatel SpeedTouch™ USB precisa fazer o download do firmware do computador para operar de acordo. É possível automatizar este processo no FreeBSD de modo que essa transferência aconteça no momento que o dispositivo foi ligado na porta USB. A seguinte informação pode ser adicionada ao arquivo `/etc/usbd.conf` para ativar essa transferência automática. O arquivo precisar ser editado como o usuário `root`.

```
device "Alcatel SpeedTouch USB"
  devname "ugen[0-9]+"
```

vendor 0x06b9
product 0x4061
attach "/usr/local/sbin/modem_run -f /usr/local/libdata/mgmt.o"

Para ativar o daemon USB, **usbd**, coloque a seguinte linha no `/etc/rc.conf`:

```
usbd_enable="YES"
```

Também é possível ajustar o **ppp** para discar na inicialização do sistema. Para fazer isso adicione as seguintes linhas no `/etc/rc.conf`. De novo, para realizar este procedimento você deverá estar *logado* como `root`.

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_profile="adsl"
```

Para que isto funcione corretamente você terá de usar o `ppp.conf` de exemplo fornecido com o port [net/pppoa](#).

18.6.2. Usando o mpd

Você pode usar o **mpd** para se conectar a uma variedade de serviços, em particular serviços PPTP. Você pode achar o **mpd** na coleção de ports, [net/mpd](#). Muitos modems ADSL requerem a criação de um túnel PPTP entre o modem e o computador. Um desses modems é o Alcatel SpeedTouch Home.

Primeiro você precisa instalar o port e então você pode configurar o **mpd** para os seus requerimentos e as definições do provedor. O port põe um conjunto de arquivos de configuração de exemplo, que são bem documentados, em *PREFIX/etc/mpd/*. Note que *PREFIX* significa o diretório em que seus ports são instalados. O padrão é */usr/local*. Um guia completo de como configurar o **mpd** em HTML fica disponível assim que o port for instalado. Ele é colocado em *PREFIX/share/mpd/*. Aqui vai um exemplo de configuração para se conectar a um serviço ADSL com o **mpd**. A configuração esta espalhada por dois arquivos. Primeiro o `mpd.conf`:

```
default:
    load adsl

adsl:
    new -i ng0 adsl adsl
    set bundle authname usuario ❶
    set bundle password senha ❷
    set bundle disable multilink

    set link no pap acfcomp protocomp
    set link disable chap
    set link accept chap
    set link keep-alive 30 10

    set ipcp no vjcomp
    set ipcp ranges 0.0.0.0/0 0.0.0.0/0

    set iface route default
    set iface disable on-demand
    set iface enable proxy-arp
    set iface idle 0

open
```

❶

O usuário usado para se autenticar com seu provedor.

❷

A senha usada para se autenticar com seu provedor.

O arquivo `mpd.links` contém informações sobre o link, ou links, que você deseja estabelecer. Um exemplo para acompanhar o exemplo acima é dado abaixo:

```
adsl:
    set link type pptp
    set pptp mode active
    set pptp enable originate incoming outcall
    set pptp self 10.0.0.1 ❶
    set pptp peer 10.0.0.138 ❷
```

❶

O endereço IP do computador que está rodando o mpd.

❷

O endereço IP do seu modem ADSL. Para o Alcatel SpeedTouch Home o padrão é 10.0.0.138.

É possível iniciar a conexão simplesmente executando o seguinte comando como `root`:

```
# mpd -b adsl
```

Você pode ver o status da conexão com o seguinte comando:

```
% ifconfig ng0
ng0: flags=88d1<UP,POINTOPOINT,RUNNING,NOARP,SIMPLEX,MULTICAST> mtu
1500
    inet 216.136.204.117 --> 204.152.186.171 netmask 0xffffffff
```

Usar o **mpd** é a maneira recomendada de se conectar ao serviço de ADSL com o FreeBSD.

18.6.3. Usando o `pptpclient`

É também possível usar o FreeBSD para se conectar a outros serviços de ADSL usando o [net/pptpclient](#).

Para usar o [net/pptpclient](#) para se conectar ao serviço ADSL, instale o port ou pacote e edite o arquivo `/etc/ppp/ppp.conf`. Você precisará estar como `root` para realizar estas duas operações. Uma seção de exemplo do `ppp.conf` é dada abaixo. Para maiores informações sobre as opções do `ppp.conf` consulte a página manual, [ppp\(8\)](#).

```
adsl:
set log phase chat lcp ipcp ccp tun command
set timeout 0
enable dns
set authname usuario ❶
set authkey senha ❷
set ifaddr 0 0
add default HISADDR
```

❶

O usuário usado para se autenticar com seu provedor.

❷

A senha usada para sua conta.

Atenção Como você precisa entrar com a senha da sua conta no `ppp.conf` em texto puro você deve ter certeza de que ninguém conseguirá ler o conteúdo deste arquivo. A seguinte série de comandos irá fazer com que o arquivo seja apenas lido pelo `root`. Veja as páginas manuais do [chmod\(1\)](#) e [chown\(8\)](#) para mais informações.

```
# chown root:wheel /etc/ppp/ppp.conf
# chmod 600 /etc/ppp/ppp.conf
```

Isso irá abrir um túnel para uma sessão PPP com seu roteador DSL. Modems ADSL com interfaces Ethernet possuem um endereço de IP local ao qual você pode se conectar. No caso do Alcatel SpeedTouch Home este endereço é o 10.0.0.138. A documentação do seu roteador deve lhe dizer qual endereço o dispositivo utiliza. Abra um túnel e inicie uma sessão PPP com o seguinte comando:

```
# pptp endereço adsl
```

Dica: Você pode querer adicionar um "E" comercial (`&") ao final do comando anterior pois o **pptp** não irá retornar o prompt se você não fizer isso.

Um túnel virtual `tun` será criado para a interação entre o **pptp** e o **ppp**. Uma vez que o prompt tiver retornado, ou que o processo **pptp** confirmou a conexão, você pode examinar o túnel da seguinte forma:

```
% ifconfig tun0
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1500
    inet 216.136.204.21 --> 204.152.186.171 netmask 0xffffffff00
    Opened by PID 918
```

Se não foi possível conectar, verifique a configuração do roteador, que geralmente é acessível através de **telnet** ou com um navegador. Se você ainda não puder se conectar examine a saída do comando `pptp` e o conteúdo do arquivo de log do **ppp** em `/var/log/ppp.log` para encontrar algumas pistas.

18.7. Usando SLIP

Originalmente contribuído por Satoshi Asami. Com adições de Guy Helmer e Piero Serini.

18.7.1. Ajustando um cliente SLIP

Mostraremos uma maneira de ajustar uma máquina FreeBSD para SLIP numa rede de endereços estáticos. Para ajuste dinâmico de endereços (seu endereço IP muda cada vez que se conecta) uma configuração mais complexa deve ser feita.

Primeiro determine em qual porta serial seu modem está conectado. Muitas pessoas ajustam um link simbólico, `/dev/modem`, que aponta para o nome real do dispositivo, `/dev/cuaaN`. Isso permite que você abstraia o nome do dispositivo caso você precise mover o modem para outra porta. Isso pode ser bastante útil quando você precisa arrumar um monte de arquivos no `/etc` e arquivos `.kermrc` por todo o sistema!

Nota: `/dev/cuaa0` é COM1, `cuaa1` é COM2, etc.

Tenha certeza de ter o seguinte na configuração do seu kernel:

```
pseudo-device    sl        1
```

Isso é incluído no kernel `GENERIC`, portanto não deve ser um problema a menos que você o removeu.

18.7.1.1. Coisas Que Você Tem De Fazer Apenas Uma Vez

1. Adicione sua máquina, gateway e servidores de nomes ao arquivo `/etc/hosts`.
O meu se parece com isso:
 2. `127.0.0.1 localhost loghost`
 3. `136.152.64.181 water.CS.Example.EDU water.CS water`
 4. `136.152.64.1 inr-3.CS.Example.EDU inr-3 slip-gateway`
 5. `128.32.136.9 ns1.Example.EDU ns1`
 6. `128.32.136.12 ns2.Example.EDU ns2`
7. Tenha certeza de que você tem `hosts` antes de `bind` no seu `/etc/host.conf` em versões do FreeBSD anteriores a 5.0. Desde o FreeBSD 5.0, o sistema usa o arquivo `/etc/nsswitch.conf` portanto tenha certeza de ter `files` antes de `dns` na linha `hosts` deste arquivo. Sem estes parâmetros coisas estranhas podem acontecer.
8. Edite o arquivo `/etc/rc.conf`.
 1. Defina seu `hostname` editando a linha que diz:
 2. `hostname="meunome.my.dominio"`

O `hostname` completo da sua máquina deve ser posto lá.

3. Adicione `s10` a listagem de interfaces de rede editando a linha que diz:
4. `network_interfaces="lo0"`

para:

```
network_interfaces="lo0 s10"
```

5. Defina algumas opções de inicialização da `s10` editando a linha:
6. `ifconfig_s10="inet ${hostname} slip-gateway netmask 0xffffffff00 up"`
7. Defina o roteador padrão editando a linha:
8. `defaultrouter="NO"`

para:

```
defaultrouter="slip-gateway"
```

9. Crie um arquivo `/etc/resolv.conf` que contenha:
 10. `domain CS.Exemplo.BR`
 11. `nameserver 128.32.136.9`
 12. `nameserver 128.32.136.12`

Como você pode ver, estas opções definem os servidores de nomes. É claro que o nome do domínio e o endereços dos servidores dependem do seu ambiente.

13. Ajuste uma senha para `root` e `toor` (e qualquer outra conta que não tenha uma senha).
14. Reinicie sua máquina e tenha certeza de que ela iniciou com o `hostname` correto.

18.7.1.2. Criando uma Conexão SLIP

1. Disque, digitar `slip` no prompt, entre com o nome da sua máquina e a senha. O que é necessário para estabelecer a conexão varia de acordo com seu ambiente. Se você usa `kermit`, pode tentar um script como o abaixo:
2. `# kermit setup`
3. `set modem hayes`
4. `set line /dev/modem`
5. `set speed 115200`
6. `set parity none`
7. `set flow rts/cts`
8. `set terminal bytesize 8`
9. `set file type binary`
10. `# A próxima macro vai discar e logar`
11. `define slip dial 643-9600, input 10 =>, if failure stop, -`
12. `output slip\x0d, input 10 Username:, if failure stop, -`
13. `output silvia\x0d, input 10 Password:, if failure stop, -`
14. `output ***\x0d, echo \x0aCONNECTED\x0a`

É claro que você precisa trocar o `hostname` e a senha. Depois de fazer isso você pode digitar `slip` do prompt do `kermit` para se conectar.

Nota: Deixar sua senha em texto puro no sistema de arquivos geralmente é uma *má* idéia.

15. Deixe o `kermit` lá (você pode suspendê-lo com um **Ctrl-z**) e, como `root`, digite:
16. `# slattach -h -c -s 115200 /dev/modem`

Se você conseguir *pingar* alguma máquina no outro lado do roteador, você está conectado! Se isso não funcionar você pode tentar usar a opção `-a` em vez de `-c` como um argumento para `slattach`.

18.7.1.3. Como Desligar uma Conexão

Faça o seguinte:

```
# kill -INT `cat /var/run/slattach.modem.pid`
```

para matar o `slattach`. Tenha em mente que você precisa ser `root` para realizar esta operação. Volte para o `kermit` (através da execução do comando `fg` se você o suspendeu) e saia dele (com o comando `q`).

O página manual do `slattach` diz que você tem de usar `ifconfig sl0 down` para marcar a interface como desligada, mas isso não parece fazer nenhuma diferença para mim. (`ifconfig sl0` diz a mesma coisa.)

Algumas vezes, seu modem pode se recusar a liberar a linha (o meu faz isso as vezes). Nesse caso, simplesmente inicie o `kermit` e saia novamente. Ele geralmente libera a linha na segunda tentativa.

18.7.1.4. Resolução de Problemas

Se isso não funcionar, sintá-se livre para me perguntar. As coisas que as pessoas têm feito até agora:

- Não usar `-c` ou `-a` no `slattach` (Isso não deve ser fatal, mas alguns usuários tem informado que isso resolve os problemas deles).
- Usar `s10` em vez de `slo` (pode ser difícil de ver a diferença em algumas fontes).
- Tente `ifconfig s10` para ver o status da sua interface. Por exemplo, você pode ver:
 - `# ifconfig s10`
 - `s10: flags=10<POINTOPOINT>`
 - `inet 136.152.64.181 --> 136.152.64.1 netmask ffffffff0`
- Se você ver "no route to host" do ping, pode haver um problema com sua tabela de roteamento. Você pode usar o comando `netstat -r` para mostra as rotas correntes:
 - `# netstat -r`
 - Routing tables
 - Destination Gateway Flags Refs Use IfaceMTU
 - Rtt Netmasks:
 -
 - (root node)
 - (root node)
 -
 - Route Tree for Protocol Family inet:
 - (root node) =>
 - default inr-3.Example.EDU UG 8 224515 s10 - -
 - localhost.Exampl localhost.Example. UH 5 42127 lo0 - 0.438
 - inr-3.Example.ED water.CS.Example.E UH 1 0 s10 - -
 - water.CS.Example localhost.Example. UGH 34 47641234 lo0 - 0.438
 - (root node)

Os exemplos anteriores são de um sistema relativamente carregado. Os números no seu sistema vão variar dependendo da atividade de rede.

18.7.2. Configurando um Servidor SLIP

Esse documento fornece sugestões para se ajustar um servidor SLIP em um sistema FreeBSD, o que tipicamente significa configurar o sistema para que inicie as conexões automaticamente assim que ocorrer o login de clientes remotos.

18.7.2.1. Pré-requisitos

Essa seção é bastante técnica em sua natureza, portanto conhecimento sobre o assunto é requerido. Assumimos que você esteja familiar com a suíte de protocolos TCP/IP, e em particular, endereçamento, máscaras de rede, *subnetting*, roteamento e protocolos de roteamento, como o RIP. Configurar serviços SLIP em um servidor de acesso discado requer conhecimento de todos esses conceitos, e se você não está familiar com eles, por favor leia o livro *TCP/IP Network Administration* de Craig Hunt publicado pela

O'Reilly & Associates, Inc. (ISBN 0-937175-82-X), ou os livros de Douglas Comer sobre TCP/IP.

É assumido que você já ajustou seu modem(s) e configurou os arquivos do sistema apropriados para permitir logins através de seus modems. Se você não preparou o sistema ainda para isso, por favor veja o tutorial sobre como configurar serviços de acesso discado; se você tem um navegador disponível, navegue pela lista de tutoriais disponíveis em <http://www.FreeBSD.org/>. Você também pode verificar as páginas manuais [sio\(4\)](#) para informações sobre o driver de dispositivo para portas seriais e [ttyps\(5\)](#), [gettytab\(5\)](#), [getty\(8\)](#), & [init\(8\)](#) para informação sobre como configurar o sistema para aceitar logins de modems, e talvez [stty\(1\)](#) para informações sobre como ajustar os parâmetros da porta serial (como *clocal* para interfaces seriais diretamente conectadas).

18.7.2.2. Rápida Introdução

Na configuração típica, usar o FreeBSD como um servidor SLIP funciona da seguinte forma: um usuário SLIP disca para seu servidor SLIP e se autentica com um login especial SLIP que usa `/usr/sbin/sliplogin` como sua shell. O programa `sliplogin` navega pelo arquivo `/etc/sliphome/slip.hosts` para achar uma linha que contenha esse usuário, e se ele encontra, conecta a linha serial com uma interface SLIP disponível e então roda o script shell `/etc/sliphome/slip.login` para configurar a interface SLIP.

18.7.2.2.1. Exemplo de um Login

Por exemplo, se um usuário SLIP fosse `Shelmerg`, a entrada de `Shelmerg` no arquivo `/etc/master.passwd` iria ser parecida com isso:

```
Shelmerg:password:1964:89::0:0:Guy Helmer -
SLIP:/usr/users/Shelmerg:/usr/sbin/sliplogin
```

Quando `Shelmerg` entrasse, `sliplogin` iria procurar por uma linha em `/etc/sliphome/slip.hosts` que contivesse o usuário; por exemplo, pode haver uma linha no `/etc/sliphome/slip.hosts` que fosse assim:

```
Shelmerg    dc-slip  sl-helmer          0xffffffffc00      autocomp
```

`sliplogin` vai encontrar essa linha, ligar a linha serial à próxima interface SLIP disponível e então executar `/etc/sliphome/slip.login` desta forma:

```
/etc/sliphome/slip.login 0 19200 Shelmerg dc-slip sl-helmer 0xffffffffc00
autocomp
```

Se tudo correr bem, `/etc/sliphome/slip.login` irá executar o `ifconfig` para a interface SLIP a qual `sliplogin` se ligou (interface `0`, no exemplo acima, que era o primeiro parâmetro na lista dada a `slip.login`) para ajustar o endereço IP local (`dc-slip`), o endereço IP remoto (`sl-helmer`), a máscara de rede (`0xffffffffc00`), e qualquer flag adicional (*autocomp*). Se alguma coisa acontecer de errado, `sliplogin` geralmente grava uma mensagem em `/var/log/messages` (veja a página manual do [syslogd\(8\)](#) e

[syslog.conf\(5\)](#) e talvez verifique o `/etc/syslog.conf` para ver o que o `syslogd` está gravando e onde ele está gravando).

OK, chega de exemplos -- vamos ajustar o sistema.

18.7.2.3. Configuração do Kernel

O kernel padrão do FreeBSD vem com duas interfaces SLIP definidas (`s10` e `s11`); você pode usar `netstat -i` para ver se alguma dessas interfaces está definida no seu kernel.

Exemplo do `netstat -i`:

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
ed0	1500	<Link>	0.0.c0.2c.5f.4a	291311	0	174209	0	133
ed0	1500	138.247.224	ivory	291311	0	174209	0	133
lo0	65535	<Link>		79	0	79	0	0
lo0	65535	loop	localhost	79	0	79	0	0
s10*	296	<Link>		0	0	0	0	
s11*	296	<Link>		0	0	0	0	

As interfaces `s10` e `s11` mostradas pelo `netstat -i` indicam que existem duas interfaces SLIP no kernel. (O asterisco depois de `s10` e `s11` indica que as interfaces estão ``desligadas".)

Entretanto, o kernel padrão do FreeBSD não vem configurado para encaminhar pacotes (por padrão, uma máquina FreeBSD não funciona como roteador) devido aos requerimentos do RFC para hosts da Internet (ver RFCs 1009 [Requeriments for Internet Gateways], 1222 [Requirements for Internet Hosts -- Communication Layers], e talvez 1127 [A Perspective on the Host Requirements]). Se você quer que seu servidor SLIP haja como um roteador, edite o arquivo `/etc/rc.conf` e troque o valor da opção `gateway_enable` para *YES*.

Você precisará reiniciar para que as novas definições entrem em ação.

Você notará que perto do final da configuração do kernel padrão (`/sys/i386/conf/GENERIC`) haverá uma linha assim:

```
pseudo-device sl 2
```

Essa é a linha que define o número de dispositivos SLIP disponíveis no kernel; o número no final da linha é o máximo número de conexões SLIP que podem estar operando ao mesmo tempo.

Por favor veja [Capítulo 9](#) em Configurando o kernel do FreeBSD para maiores informações sobre como reconfigurar seu kernel.

18.7.2.4. Configuração do Sliplogin

Como mencionado anteriormente, existem três arquivos no diretório `/etc/sliphome` que são parte da configuração do `/usr/sbin/sliplogin` (veja [sliplogin\(8\)](#) para a página manual do `sliplogin`): `slip.hosts`, que define os usuários SLIP associados

aos endereços IP; `slip.login`, que geralmente configura a interface SLIP; e (opcionalmente) `slip.logout`, que desfaz o que o `slip.login` fez quando a conexão serial for terminada.

18.7.2.4.1. Configuração do `slip.hosts`

`/etc/sliphome/slip.hosts` contém linhas que têm até quatro itens separados por espaços:

- Login ID do usuários SLIP
- Endereço local (local para o servidor SLIP) do link SLIP
- Remote address of the SLIP link
- Máscara de Rede

Os endereços local e remoto podem ser nomes (traduzidos para endereços IP pelo `/etc/hosts` ou por um serviço de nomes, dependendo das suas especificações no `/etc/nsswitch.conf` no FreeBSD 5.X ou `/etc/host.conf` no FreeBSD 4.X), e a máscara de rede pode ser um nome que será traduzido pelo `/etc/networks`. Em um sistema de exemplo o arquivo `/etc/sliphome/slip.hosts` pode se parecer com isso:

```
#
# login local-addr      remote-addr      mask      opt1      opt2
#                       (normal,compress,noicmp)
#
Shelmerg dc-slip      sl-helmerg      0xfffffc00      autocomp
```

No final da linha existe uma ou mais opções.

- *normal* -- sem compressão de cabeçalho
- *compress* -- comprime os cabeçalhos
- *autocomp* -- comprime os cabeçalhos se a ponta remota permitir isso
- *noicmp* -- desabilita pacotes ICMP (portanto qualquer pacote "ping" será descartado em vez de utilizar sua conexão)

Sua escolha de endereços locais e remotos para os links SLIP depende se você irá dedicar uma sub-rede ou se irá utilizar "proxy ARP" no servidor SLIP (não é proxy ARP "verdadeiro" mas essa é a terminologia utilizada nessa seção para descrever isso). Se você não tem certeza de qual método escolher ou como definir os endereços IP, por favor veja os livros sobre TCP/IP na listagem de pré-requisitos ([Seção 18.7.2.1](#)) e/ou consulte seu gerente de rede.

Se você vai utilizar uma rede separada para seus clientes SLIP, você deverá alocar os endereços IP para seus clientes dentre os disponíveis para essa rede escolhida. Então, você provavelmente precisará configurar uma rota estática para essa rede SLIP através do seu servidor SLIP no roteador mais próximo.

Caso você use "proxy ARP", você precisará definir os endereços IP dos seus clientes SLIP dentro os disponíveis na rede Ethernet do seu servidor e também deverá ajustar o `/etc/sliphome/slip.login` e `/etc/sliphome/slip.logout` para usarem [arp\(8\)](#) para gerenciar as entradas proxy-ARP na tabela ARP do servidor SLIP.

18.7.2.4.2. Configuração do `slip.login`

Um `/etc/sliphome/slip.login` típico se parece com o abaixo:

```
#!/bin/sh -
#
#      @(#)slip.login 5.1 (Berkeley) 7/1/90
#
# arquivo de login genérico para uma linha slip. sliplogin invoca
# isso com os seguintes parâmetros:
#      1      2      3 4      5 6      7-n
#  slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
```

Esse arquivo `slip.login` meramente executa o comando `ifconfig` para a interface SLIP apropriada com os endereços local e remoto e a máscara de rede da interface.

Se você decidiu utilizar "proxy ARP" (em vez de alocar uma sub-rede para seus clientes SLIP) seu `/etc/sliphome/slip.login` irá se parecer com o abaixo:

```
#!/bin/sh -
#
#      @(#)slip.login 5.1 (Berkeley) 7/1/90
#
# arquivo de login genérico para uma linha slip. sliplogin invoca
# isso com os seguintes parâmetros:
#      1      2      3 4      5 6      7-n
#  slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
# Responde requisições ARP para o cliente SLIP com nosso endereço
Ethernet
/usr/sbin/arp -s $5 00:11:22:33:44:55 pub
```

Essa linha adicional, `arp -s $5 00:11:22:33:44:55 pub` cria uma entrada na tabela ARP do servidor. Essa entrada faz com que o servidor responda com o endereço MAC do próprio servidor quando alguma outra máquina na rede Ethernet desejar falar com endereço IP do cliente SLIP.

Quando estiver utilizando o exemplo acima tenha certeza de trocar o endereço MAC (00:11:22:33:44:55) pelo endereço da placa de rede Ethernet do seu sistema ou o "proxy ARP" definitivamente não funcionará! Você pode descobrir o endereço MAC do seu servidor olhando a saída do comando `netstat -i`; a segunda linha deve se parecer com isso:

```
ed0    1500 <Link>0.2.c1.28.5f.4a  191923 0    129457    0    116
```

Isso indica que o endereço da placa de rede, o MAC address, é 00:02:c1:28:5f:4a -- os pontos no endereço fornecido por `netstat -i` precisam ser trocados por dois pontos e devem ser adicionados zeros em qualquer número hexadecimal que tenha um só dígito

para converter para o formato que o [arp\(8\)](#) deseja; veja a página manual do [arp\(8\)](#) para informações completas de utilização.

Nota: Quando você cria o `/etc/sliphome/slip.login` e `/etc/sliphome/slip.logout`, o bit ```execute"` (`chmod 755 /etc/sliphome/slip.login /etc/sliphome/slip.logout`) precisa estar definida, ou `sliplogin` não conseguirá executá-los.

18.7.2.4.3. Configuração do `slip.logout`

O `/etc/sliphome/slip.logout` não é estritamente necessário (a não ser que você esteja implementando ```proxy ARP"`), mas se você decidir criá-lo abaixo temos um exemplo básico do script:

```
#!/bin/sh -
#
#      slip.logout

#
# arquivo de login genérico para uma linha slip. sliplogin invoca
# isso com os seguintes parâmetros:
#      1      2      3 4      5 6      7-n
#      slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
```

Se você estiver utilizando ```proxy ARP"` irá querer que o `/etc/sliphome/slip.logout` remova a entrada ARP do cliente SLIP:

```
#!/bin/sh -
#
#      @(#)slip.logout

#
# arquivo de login genérico para uma linha slip. sliplogin invoca
# isso com os seguintes parâmetros:
#      1      2      3 4      5 6      7-n
#      slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
# Para de responder requisições ARP para o cliente SLIP
/usr/sbin/arp -d $5
```

O comando `arp -d $5` remove a entrada ARP que o `slip.login` adicionou quando o cliente SLIP se conectou.

É necessário repetir: tenha certeza de que o `/etc/sliphome/slip.logout` tem o bit de execução definido depois que você criá-lo (por exemplo, `chmod 755 /etc/sliphome/slip.logout`).

18.7.2.5. Considerações de Roteamento

Se você não está utilizando "proxy ARP" para rotear os pacotes entre seus clientes SLIP e o resto de sua rede (e talvez a Internet), você deve provavelmente ter rotas estáticas para seus roteadores para rotear a sub-rede dos seus clientes SLIP via seu servidor SLIP.

18.7.2.5.1. Rotas Estáticas

Adicionar rotas estáticas para seus roteadores mais próximos pode ser problemático (ou impossível se você não tem autoridade para isso). Se você tem uma rede com diversos roteadores, alguns deles, como os feitos pela Cisco e Proteon, podem não apenas precisar ter as rotas configuradas mas também precisam saber sobre quais rotas eles devem informar os outros roteadores, portanto alguma experiência e certos ajustes podem ser necessários para fazer com que sua rede baseada em rotas estáticas funcione.

18.7.2.5.2. Executando o *gated*

Nota: O *gated* é um programa proprietário que não tem seu código disponível para o público mais (informações adicionais no site do [gated](#)). Essa seção só existe para manter compatibilidade para as pessoas que ainda estão utilizando a versão antiga.

Uma alternativa para as dores de cabeça com rotas estáticas é instalar o *gated* no seu servidor SLIP e configurar ele para usar os protocolos de roteamento apropriado (RIP/OSPF/BGP/EGP) para dizer aos outros roteadores sobre sua rede SLIP. Você precisará editar um arquivo `/etc/gated.conf` para configurar seu *gated*; aqui está um exemplo, similar ao que o autor utilizou em um servidor SLIP FreeBSD:

```
#
# gated configuration file for dc.dsu.edu; for gated version 3.5alpha5
# Only broadcast RIP information for xxx.xxx.yy out the ed Ethernet
interface
#
#
# tracing options
#
traceoptions "/var/tmp/gated.output" replace size 100k files 2 general
;

rip yes {
    interface sl noripout noripin ;
    interface ed ripin ripout version 1 ;
    traceoptions route ;
} ;

#
# Turn on a bunch of tracing info for the interface to the kernel:
kernel {
    traceoptions remnants request routes info interface ;
} ;

#
# Propagate the route to xxx.xxx.yy out the Ethernet interface via RIP
#
```

```

export proto rip interface ed {
    proto direct {
        xxx.xxx.yy mask 255.255.252.0 metric 1; # SLIP connections
    } ;
} ;

#
# Accept routes from RIP via ed Ethernet interfaces

import proto rip interface ed {
    all ;
} ;

```

O exemplo acima de um arquivo `gated.conf` envia informação de roteamento sobre a sub-rede SLIP `xxx.xxx.yy` via RIP por Ethernet; se você está utilizando um driver Ethernet diferente do `ed` você precisará trocar as referencias a `ed` pela interface apropriada. Esse exemplo também configura a gravação de informações de *debug* sobre a atividade do `gated`; você pode certamente desligar isso se o `gated` funciona OK para você. Você precisará trocar o `xxx.xxx.yy` pelo endereço da sua rede SLIP (tenha certeza de trocar a máscara de rede na cláusula *proto direct* também).

Uma vez que você instalou e configurou o `gated` no seu sistema você precisará dizer aos script de inicialização do FreeBSD para rodarem o `gated` no lugar do `routed`. A maneira mais fácil é definir as variáveis `router` e `router_flags` no `/etc/rc.conf`. Por favor veja a página manual do `gated` para mais informações.

Capítulo 19. Uso Avançado de Redes

Índice

- 19.1. [Índice](#)
- 19.2. [Gateways e Rotas](#)
- 19.3. [Rede Sem Fio \(Wireless\)](#)
- 19.4. [Bluetooth](#)
- 19.5. [Ligação \(Bridging\)](#)
- 19.6. [NFS](#)
- 19.7. [Operação Sem Disco \(Diskless Operation\)](#)
- 19.8. [Rede Digital de Serviços Integrados - RDSI \(ISDN\)](#)
- 19.9. [NIS/YP](#)
- 19.10. [DHCP](#)
- 19.11. [DNS](#)
- 19.12. [NTP](#)
- 19.13. [NAT - Tradução de Endereço de Rede](#)
- 19.14. [O "Super-Servidor" `inetd`](#)
- 19.15. [PLIP - IP em Linha Paralela](#)
- 19.16. [IPv6](#)

19.1. Índice

Este capítulo aborda alguns serviços de redes usados com mais frequência em sistemas UNIX®. Vamos abordar como definir, configurar, testar e manter todos os serviços de rede que o FreeBSD usa. Adicionalmente, existem exemplos de arquivos de configuração inclusos pelo capítulo para seu usufruto.

Após a leitura deste capítulo, você saberá:

- O básico sobre gateways e rotas.
- Como fazer o FreeBSD atuar como uma ponte.
- Como ativar um sistema de arquivos em rede.
- Como ativar inicialização remota em uma máquina sem disco.
- Como ativar um servidor de informação de rede.
- Como ativar configurações automáticas de rede usando DHCP.
- Como ativar um servidor de nome de domínio.
- Como sincronizar a hora e data, e ativar um servidor de hora com o protocolo NTP.
- Como ativar tradução de endereço de rede.
- Como gerenciar o daemon **inetd**.
- Como conectar dois computadores via PLIP.
- Como ativar o IPv6 em uma máquina FreeBSD.

Antes de ler este capítulo, você deve:

- Compreender os princípios dos scripts `/etc/rc`.
- Estar familiarizado com a terminologia básica de redes.

19.2. Gateways e Rotas

Contribuição de Coranth Gryphon.

Para uma máquina ser capaz de encontrar outra através de uma rede, é necessário um mecanismo que descreva como ir de uma para a outra. Isto é chamado *roteamento*. Uma ``rota" é um par definido de endereços: um ``destino" e um ``gateway". O par indica que se você estiver tentando alcançar este *destino*, deve se comunicar através deste *gateway*. Existem três tipos de destinos: máquinas individuais, subredes e ``padrão". A ``rota padrão" é usada se nenhuma das outras rotas for aplicável. Vamos falar um pouco mais sobre rotas padrão mais tarde. Também existem três tipos de gateways: máquinas individuais, interfaces (também chamadas de ``enlaces") e endereços físicos de hardware Ethernet (endereços MAC).

19.2.1. Um Exemplo

Para ilustrar diferentes aspectos do roteamento, vamos usar o seguinte exemplo do `netstat`:

```
% netstat -r
Routing tables

Destination      Gateway          Flags    Refs      Use      Netif
Expire

default          outside-gw      UGSc     37        418      ppp0
localhost        localhost       UH        0         181      lo0
test0            0:e0:b5:36:cf:4f UHLW     5        63288    ed0      77
10.20.30.255     link#1          UHLW     1         2421
exemplo.com      link#1          UC        0         0
host1            0:e0:a8:37:8:1e UHLW     3         4601    lo0
host2            0:e0:a8:37:8:1e UHLW     0         5        lo0 =>
host2.exemplo.com link#1          UC        0         0
224             link#1          UC        0         0
```

As duas primeiras linhas especificam a rota padrão (a qual será abordada na [próxima seção](#)) e a rota para `localhost` (a própria máquina).

A interface (coluna *Netif*) que esta tabela de roteamento usa para especificar o *localhost* é `lo0`, também conhecido como o dispositivo de loopback. Isto quer dizer para manter interno todo o tráfego para este destino, ao invés de enviá-lo para fora pela rede local, uma vez que simplesmente deverá retornar ao local de origem.

As próximas coisas que chamam a atenção são os endereços começando com `0:e0:`. Estes são endereços de hardware Ethernet, também conhecidos como endereços MAC. O FreeBSD irá identificar automaticamente quaisquer máquinas (no exemplo `test0`) na Ethernet local e adicionar uma rota direta para aquela máquina pela interface Ethernet, `ed0`. Há, também, um tempo de expiração (timeout) associado a este tipo de rota, que é usado em caso de não se receber mais transmissões da máquina dentro de um prazo específico de tempo. Quando isto acontece, a rota para esta máquina será automaticamente removida. Estas máquinas são identificadas usando um mecanismo conhecido como RIP (Protocolo de Informação de Roteamento - Routing Information Protocol), que calcula rotas para as máquinas locais baseado na determinação do menor caminho.

O FreeBSD também adicionará rotas de subrede para a subrede local (`10.20.30.255` é o endereço de difusão (broadcast) para a subrede `10.20.30`, e `exemplo.com` é o nome de domínio associado àquela subrede). A designação *link#1* refere-se à primeira placa Ethernet na máquina. Você irá notar que nenhuma interface adicional é especificada para aquelas.

Ambos os grupos (máquinas de rede local e subredes locais) têm suas rotas configuradas automaticamente por um daemon chamado **routed**. Se ele não estiver

sendo executado, então somente existirão rotas definidas estaticamente (p.ex. configuradas explicitamente).

A linha *host1* refere-se à nossa máquina, conhecida pelo endereço Ethernet. Uma vez que somos a máquina transmissora, o FreeBSD sabe fazer uso da interface loopback (100) ao invés de transmitir pela interface Ethernet afora.

As duas linhas *host2* são um exemplo do que acontece quando usamos um apelido [ifconfig\(8\)](#) (veja a seção sobre Ethernet por razões pelas quais faríamos isso). O símbolo => após a interface 100 diz que não somente estamos usando a loopback (uma vez que este endereço também se refere à máquina local), mas especificamente que é um apelido. Tais rotas somente surgem nas máquinas que suportam o apelido; todas as outras máquinas na rede local simplesmente terão uma linha *link#1* para tais rotas.

A última linha (subrede de destino 224) trata do uso de multicast, que será abordado em outra seção.

Finalmente, diversos atributos das rotas podem ser vistos na coluna *Flags*. Abaixo está uma breve tabela contendo algumas destas flags e seus significados:

U Up: A rota é ativa.

H Host: O destino da rota é uma única máquina.

G Gateway: Envie qualquer coisa para este destino por este sistema remoto, que o mesmo irá calcular a partir de lá para onde enviá-la. it.

S Static: Esta rota foi configurada manualmente, e não gerada automaticamente pelo sistema.

C Clone: Gera uma nova rota baseada nesta rota para máquinas a quem nos conectarmos. Este tipo de rota é normalmente usada em redes locais.

W WasCloned: Indica uma rota que foi autoconfigurada baseada em uma rota (Clone) da rede local.

L Link: Rota envolve referências ao hardware Ethernet.

19.2.2. Rotas Padrão

Quando um sistema local precisa fazer uma conexão a um sistema remoto, ele verifica a tabela de roteamento para determinar se existe um caminho conhecido. Se o sistema remoto estiver em uma das subredes que já sabemos como alcançar (rotas Clonadas), então o sistema verifica se pode se conectar por aquela interface.

Se falharem todos os caminhos conhecidos, o sistema tem uma última opção: a rota ``padrão". Esta rota é um tipo especial rota de saída, ou gateway (normalmente a única presente no sistema), e é sempre indicada por um *c* no campo flags. Para sistemas em uma rede local, este gateway é configurado para qualquer máquina que tenha uma

conexão direta para o mundo exterior (seja via enlace PPP, DSL, modem a cabo, T1 ou outra interface de rede).

Se você estiver configurando a rota padrão em uma máquina que esteja funcionando como o gateway para o mundo exterior, então a rota padrão será a máquina gateway localizada no seu Provedor de Serviços Internet (PSI).

Vamos dar uma olhada em um exemplo de rotas padrão. Esta é uma configuração comum:

```
[Local2] <--ether--> [Local1] <--PPP--> [ISP-Serv] <--ether--> [T1-GW]
```

Os sistemas `Local1` e `Local2` estão em seu sítio local. `Local1` está conectado a um PSI por uma conexão PPP discada. Este servidor PPP está conectado por uma rede local a outro computador gateway através de uma interface externa ao canal Internet do PSI.

As rotas padrão para cada uma das suas máquinas serão:

Sistema Gateway Padrão Interface

Local2	Local1	Ethernet
Local1	T1-GW	PPP

Uma pergunta comum é ``Por que (ou como) nós configuramos o `T1-GW` para ser o gateway padrão para `Local1`, ao invés do servidor do PSI ao qual está conectado?''.

Lembre-se, uma vez que a interface PPP está usando um endereço da rede local do PSI para o seu lado da conexão, rotas para qualquer outra máquina na rede local do PSI serão criadas automaticamente. Assim, uma vez que você já sabe como alcançar a máquina `T1-GW`, então não há necessidade do passo intermediário de enviar tráfego para o servidor do PSI.

Como observação final, é comum usar o endereço `x.x.x.1` como o endereço do gateway para sua rede local. Então (usando o mesmo exemplo), se o seu espaço de endereçamento classe-C era `10.20.30` e seu PSI estava usando `10.9.9` então as rotas padrão seriam:

Sistema	Rota Padrão
Local2 (10.20.30.2)	Local1 (10.20.30.1)
Local1 (10.20.30.1, 10.9.9.30)	T1-GW (10.9.9.1)

19.2.3. Sistemas Duplamente Conectados

Existe um outro tipo de configuração que devemos abordar, que se trata de um sistema pertencente a duas redes diferentes. Tecnicamente, qualquer máquina funcionando como um gateway (no exemplo acima, usando uma conexão PPP) conta como um sistema duplamente conectado. Mas, na verdade, o termo somente é usado para se referir a uma máquina conectada a duas redes locais diferentes.

Em um caso, a máquina tem duas placas Ethernet, cada uma possuindo um endereço em subredes separadas. Alternativamente, a máquina pode ter somente uma placa Ethernet e estar usando um apelido [ifconfig\(8\)](#). O primeiro caso é usado se duas redes Ethernet fisicamente separadas estão em uso, e o segundo se há somente um segmento físico Ethernet, mas duas subredes separadas logicamente.

De um jeito ou de outro, as tabelas de roteamento são configuradas de forma que cada subrede saiba que esta máquina é o gateway definido (rota de entrada) para a outra subrede. Esta configuração, com a máquina atuando como um roteador entre duas subredes, é muitas vezes usado quando precisamos implementar filtragem de pacotes ou um firewall de segurança em cada uma ou em ambas as direções.

Se você deseja que esta máquina, na prática, passe adiante pacotes entre as duas interfaces, você precisa dizer ao FreeBSD para permitir esta capacidade.

19.2.4. Construindo um Roteador

Um roteador de rede é simplesmente um sistema que repassa pacotes de uma interface para outra. Padrões da Internet e boas práticas de engenharia impedem o Projeto FreeBSD de permitir isto por padrão no FreeBSD. Você pode permitir esta característica alterando a seguinte variável para *YES* no [rc.conf\(5\)](#):

```
gateway_enable=YES    # Set to YES if this host will be a gateway
```

Esta opção irá ajustar a variável do [sysctl\(8\)](#) chamada *net.inet.ip.forwarding* para *1*. Se você precisar parar o roteamento temporariamente, você pode reajustá-la para *0*.

Seu novo roteador precisará de rotas para saber aonde enviar o tráfego. Se sua rede for simples o suficiente você pode usar rotas estáticas. O FreeBSD também vem com o daemon de roteamento padrão do BSD [routed\(8\)](#), o qual fala RIP (ambas as versões 1 e 2) e IRDP. O suporte para BGP v4, OSPF v2 e outros protocolos de roteamento sofisticados estão disponíveis com o pacote [net/zebra](#). Produtos comerciais, como o *gated*, também estão disponíveis para soluções mais complexas de roteamento.

Mesmo quando o FreeBSD é configurado desta forma, ele ainda não cumpre completamente com os requisitos padrões da Internet para roteadores. Mas chega bem perto para uso normal, entretanto.

19.2.5. Propagação do Roteamento

Nós já falamos sobre como definimos nossas rotas para o mundo exterior, mas não sobre como o mundo exterior nos encontra.

Já sabemos como as tabelas de roteamento podem ser configuradas de forma que todo o tráfego para um espaço de endereçamento em particular (nos nossos exemplos, uma subrede classe-C) pode ser enviado a um determinado sistema naquela rede, o qual irá repassar os pacotes para dentro.

Quando você obtiver a designação de um espaço de endereçamento para seu sítio, seu provedor de serviços irá configurar suas tabelas de roteamento de forma que todo o tráfego para sua subrede será enviado através do seu enlace PPP ao seu sítio. Mas como os sítios através do país saberão enviar ao seu PSI?

Há um sistema (muito parecido com a informação DNS distribuída) que mantém registros de todos os espaços de endereçamento designados, e que define seu ponto de conexão à Espinha Dorsal Internet. A "Espinha Dorsal" são os principais canais que carregam o tráfego da Internet através do país e ao redor do mundo. Cada máquina de espinha dorsal tem uma cópia de um conjunto mestre de tabelas, as quais direcionam tráfego a uma determinada rede e a um provedor de espinha dorsal específico e, de lá, abaixo na cadeia de provedores de serviços até atingir a sua rede.

É a tarefa do seu provedor de serviços anunciar para os sítios da espinha dorsal que eles são o ponto de conexão (e, assim, o caminho de entrada) para seu sítio. Isto é conhecido como propagação de rotas.

19.2.6. Resolução de Problemas

Às vezes, há um problema com propagação de rotas, e alguns sítios ficam incapazes de se conectar a você. Talvez o comando mais útil para tentar descobrir onde o roteamento está em pane é o [traceroute\(8\)](#). Ele é igualmente útil se você não parece conseguir realizar uma conexão a uma máquina remota (p.ex. o [ping\(8\)](#) falhou).

O comando [traceroute\(8\)](#) é executado com o nome do sistema remoto que você está tentando se conectar. Ele vai mostrar os sistemas gateway ao longo do caminho da tentativa, eventualmente alcançando o sistema-alvo, ou terminando por falta de conectividade.

Para maiores informações, veja a página do manual do [traceroute\(8\)](#).

19.2.7. Roteamento Multicast

O FreeBSD suporta aplicações multicast e roteamento multicast de forma nativa. Aplicações multicast não requerem qualquer configuração especial no FreeBSD; aplicações geralmente irão funcionar sem necessidade de ajustes. O roteamento multicast requer que o suporte seja compilado no núcleo, ou kernel:

```
options MROUTING
```

Além disso, o daemon de roteamento multicast [mrouted\(8\)](#) precisa ser configurado para ativar túneis e DVMRP através do `/etc/mrouted.conf`. Mais detalhes sobre as configurações multicast podem ser encontradas nas páginas de manual do mrouted.

19.3. Rede Sem Fio (Wireless)

Escrito por Eric Anderson.

19.3.1. Introdução

Pode ser muito útil a possibilidade de usar um computador sem o incômodo de ter um cabo de rede conectado o tempo todo. O FreeBSD pode ser usado como um cliente sem fio e até mesmo como um "ponto de acesso" sem fio.

19.3.2. Modos de Operação Sem Fio

Há duas formas diferentes de configurar dispositivos sem fio 802.11: BSS e IBSS.

19.3.2.1. Modo BSS

BSS é o modo tipicamente utilizado. O modo BSS é também chamado de modo infraestrutura. Neste modo, uma quantidade de pontos de acesso sem fio está conectada a uma rede cabeada. Cada rede sem fio tem seu próprio nome. Este nome é chamado SSID da rede.

Clientes sem fio conectam-se a estes pontos de acesso. O padrão IEEE 802.11 define o protocolo que redes sem fio usam para conexões. Um cliente sem fio pode ser ligado a uma rede específica, quando um SSID é configurado. Um cliente sem fio também pode conectar-se a qualquer rede desde que não configure explicitamente um SSID.

19.3.2.2. Modo IBSS

O modo IBSS, também chamado modo ad-hoc, é projetado para conexões ponto-a-ponto. Existem, de fato, dois tipos de modo ad-hoc. Um é o modo IBSS, também chamado de ad-hoc ou modo ad-hoc do IEEE. Este modo é definido pelos padrões IEEE 802.11. O segundo modo é chamado modo ad-hoc demo ou modo ad-hoc da Lucent (e, às vezes, confusamente, modo ad-hoc). Este é o antigo modo ad-hoc, pré-802.11 e somente deve ser usado em instalações legadas. Não iremos abordar nenhum dos modos ad-hoc mais adiante.

19.3.3. Modo Infra-estrutura

19.3.3.1. Pontos de Acesso

Pontos de acesso são dispositivos de rede sem fio que permitem um ou mais clientes sem fio utilizar o dispositivo como um concentrador central. Ao usar um ponto de

acesso, todos os clientes comunicam-se através do ponto de acesso. Múltiplos pontos de acesso são freqüentemente usados para cobrir uma área completa como uma casa, negócio ou parque, com uma rede sem fio.

Pontos de acesso tipicamente possuem múltiplas conexões de rede: o cartão sem fio e um ou mais adaptadores Ethernet para conexão ao resto da rede.

Pontos de acesso podem ser comprados montados ou você pode construir o seu próprio com FreeBSD e um cartão sem fio suportado. Diversos fornecedores fazem pontos de acesso sem fio e cartões sem fio com características variadas.

19.3.3.2. Construindo um Ponto de Acesso FreeBSD

19.3.3.2.1. Requisitos

Para ativar um ponto de acesso com FreeBSD, você precisa ter uma placa sem fio compatível. Atualmente, somente são suportadas as placas com o chipset Prism. Você também vai precisar de uma placa de rede com fio suportada pelo FreeBSD (não deve ser difícil encontrar, o FreeBSD suporta diversos diferentes dispositivos). Neste guia, vamos assumir que você deseja fazer uma ponte ([bridge\(4\)](#)) de todo o tráfego entre o dispositivo sem fio e a rede conectada na placa de rede com fio.

A funcionalidade hostap que o FreeBSD usa para implementar o ponto de acesso funciona melhor com certas versões de firmware. Placas Prism 2 devem usar o firmware versão 1.3.4 ou mais recente. Placas Prism 2.5 e Prism 3 devem usar o firmware 1.4.9. Versões anteriores de firmware podem ou não funcionar corretamente. Neste momento, a única forma de atualizar placas é com ferramentas de atualização para Windows® disponibilizadas pelo fabricante de sua placa.

19.3.3.2.2. Ativando o Ponto de Acesso

Primeiro, certifique-se de que seu sistema reconhece a placa sem fio:

```
# ifconfig -a
wi0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet6 fe80::202:2dff:fe2d:c938%wi0 prefixlen 64 scopeid 0x7
    inet 0.0.0.0 netmask 0xff000000 broadcast 255.255.255.255
    ether 00:09:2d:2d:c9:50
    media: IEEE 802.11 Wireless Ethernet autoselect (DS/2Mbps)
    status: no carrier
    ssid ""
    stationname "FreeBSD Wireless node"
    channel 10 authmode OPEN powersavemode OFF powersavesleep 100
    wepmode OFF weptxkey 1
```

Não se preocupe com os detalhes agora, apenas certifique-se de que algo é mostrado, indicando que você tem uma placa sem fio instalada. Se você tiver problemas em encontrar a interface sem fio e estiver usando um PC card, você pode querer verificar as páginas de manual [pccardc\(8\)](#) e [pccardd\(8\)](#) para maiores informações.

A seguir, você vai precisar carregar um módulo para ter o FreeBSD funcionando como ponte e pronto para o ponto de acesso. Para carregar o módulo [bridge\(4\)](#), simplesmente execute o comando:

```
# kldload bridge
```

Não deve ser mostrado nenhum erro ao se carregar o módulo. Se houve algum, talvez você precise compilar o código [bridge\(4\)](#) no kernel do seu sistema. A seção [Ligação](#) do handbook deve ajudá-lo a executar esta tarefa.

Agora que você tem a parte de ligação pronta, precisamos dizer ao kernel do FreeBSD quais interfaces deve ligar. Fazemos isto usando [sysctl\(8\)](#):

```
# sysctl net.link.ether.bridge=1
# sysctl net.link.ether.bridge_cfg="wi0 xl0"
# sysctl net.inet.ip.forwarding=1
```

Agora é a hora de configurar a placa sem fio.

O seguinte comando vai ativar a placa como um ponto de acesso:

```
# ifconfig wi0 ssid my_net channel 11 media DS/11Mbps mediaopt hostap
up stationname "FreeBSD AP"
```

A linha [ifconfig\(8\)](#) ativa a interface `wi0`, configura seu SSID para `my_net`, e ajusta o nome da estação para `FreeBSD AP`. A opção `media DS/11Mbps` configura a placa em modo 11Mbps e é necessária para qualquer parâmetro `mediaopt` ter efeito. A opção `mediaopt hostap` coloca a interface em modo ponto de acesso. A página de manual [wicontrol\(8\)](#) tem as opções válidas para canais de seu controlador de domínio.

Agora você deve ter um ponto de acesso completamente funcional ativado e funcionando. Recomendamos a leitura de [wicontrol\(8\)](#), [ifconfig\(8\)](#), e [wi\(4\)](#) para maiores informações.

Também é sugerido que você leia a seção sobre criptografia a seguir.

19.3.3.2.3. Informações de Estado

Uma vez que o ponto de acesso está configurado e operacional, os operadores vão querer ver os clientes associados ao ponto de acesso. Em qualquer momento, o operador pode digitar:

```
# wicontrol -l
l station:
00:09:b7:7b:9d:16 asid=04c0, flags=3<ASSOC,AUTH>, caps=1<ESS>,
rates=f<1M,2M,5.5M,11M>, sig=38/15
```

Isto mostra que há uma estação associada, junto com seus parâmetros. O sinal indicado deve ser usado somente como uma indicação relativa de potência. Sua tradução para dBm ou outras unidades variam entre diferentes revisões de firmware.

19.3.3.3. Clientes

Um cliente sem fio é um sistema que acessa um ponto de acesso ou outro cliente diretamente.

Tipicamente, os clientes sem fio possuem somente um dispositivo de rede, a placa de rede sem fio.

Existem algumas poucas formas diferentes de configurar um cliente sem fio. Estas configurações são baseadas nos diferentes modos sem fio, geralmente BSS (modo infraestrutura, que requer um ponto de acesso), e IBSS (ad-hoc, ou modo ponto-a-ponto). Em nosso exemplo, vamos usar o mais popular dos dois, modo BSS, para comunicação com um ponto de acesso.

19.3.3.3.1. Requisitos

Há somente um único requisito real para ativar o FreeBSD como um cliente sem fio. Você precisa de uma placa sem fio suportada pelo FreeBSD.

19.3.3.3.2. Ativando um Cliente FreeBSD Sem Fio

Antes de começar, precisa saber algumas coisas sobre a rede sem fio à qual estará se conectando. Neste exemplo, estamos nos conectando a uma rede que tem o nome de *my_net*, e criptografia desligada.

Neste exemplo, não estamos usando criptografia o que é uma situação de perigo. Na próxima seção, você aprenderá como ativar criptografia, porque é importante fazê-lo e porque algumas tecnologias de criptografia ainda não o protegem completamente.

Certifique-se de que sua placa é reconhecida pelo FreeBSD:

```
# ifconfig -a
wi0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet6 fe80::202:2dff:fe2d:c938%wi0 prefixlen 64 scopeid 0x7
    inet 0.0.0.0 netmask 0xff000000 broadcast 255.255.255.255
    ether 00:09:2d:2d:c9:50
    media: IEEE 802.11 Wireless Ethernet autoselect (DS/2Mbps)
    status: no carrier
    ssid ""
    stationname "FreeBSD Wireless node"
    channel 10 authmode OPEN powersavemode OFF powersavesleep 100
    wepmode OFF weptxkey 1
```

Agora vamos configurar a placa com os parâmetros corretos para nossa rede:

```
# ifconfig wi0 inet 192.168.0.20 netmask 255.255.255.0 ssid my_net
```

Substitua 192.168.0.20 e 255.255.255.0 com um endereço IP e máscara de subrede válidos em sua rede com fio. Lembre-se que nosso ponto de acesso está fazendo a ligação (ponte) de dados entre a rede sem fio e a rede com fio, de forma que vai parecer aos outros dispositivos na rede que você está na rede com fio, assim como eles.

Feito isto, você deve ser capaz de usar ping nos sistemas na rede com fio assim como se estivesse conectado usando uma conexão padrão, com fio.

Se estiver com problemas em sua conexão sem fio, certifique-se que está associado (conectado) ao ponto de acesso:

```
# ifconfig wi0
```

deve retornar alguma informação, e você deve ver:

```
status: associated
```

Se não mostrar que está associado, então você deve estar fora de alcance do ponto de acesso, estar sem criptografia ativada, ou possivelmente tem um problema de configuração.

19.3.3.4. Criptografia

Criptografia em uma rede sem fio é importante porque você não mais terá a capacidade de manter a rede contida em uma área bem protegida. Seus dados sem fio serão transmitidos através de toda a sua vizinhança, então quem quiser poderá ler seus dados. Aqui entra a criptografia. Criptografando os dados transmitidos por ondas aéreas, você torna muito mais difícil a qualquer um capturar seus dados do ar.

As duas formas mais comuns de criptografar os dados entre seu cliente e o ponto de acesso são WEP, e [ipsec\(4\)](#).

19.3.3.4.1. WEP

WEP é uma abreviação para Protocolo para Equivalência em Fio (Wired Equivalency Protocol). WEP é uma tentativa de tornar as redes sem fio tão seguras quanto uma rede com fio. Infelizmente, foi quebrado e é bem trivial quebrá-lo. Isto também significa que não é uma coisa para se contar quando se trata de criptografar dados sensíveis.

É melhor que nada, então faça o seguinte para ativar WEP em seu novo ponto de acesso FreeBSD:

```
# ifconfig wi0 inet up ssid my_net wepmode on wepkey 0x1234567890  
media DS/11Mbps mediaopt hostap
```

E você pode ativar WEP em um cliente com este comando:

```
# ifconfig wi0 inet 192.168.0.20 netmask 255.255.255.0 ssid my_net  
wepmode on wepkey 0x1234567890
```

Note que você deve substituir o *0x1234567890* por uma chave mais exclusiva.

19.3.3.4.2. IPsec

[ipsec\(4\)](#) é uma ferramenta muito mais robusta e poderosa para criptografar dados através de uma rede. Esta é definitivamente a forma preferida de criptografar dados em

uma rede sem fio. Você pode ler mais sobre segurança [ipsec\(4\)](#) e como implementá-la na seção [IPsec](#) do handbook.

19.3.3.5. Ferramentas

Há um pequeno número de ferramentas disponível para uso na determinação de problemas e ativação de sua rede sem fio, e aqui vamos tentar descrever algumas delas e o que fazem.

19.3.3.5.1. O Pacote *bsd-airtools*

O pacote **bsd-airtools** é um conjunto de ferramentas completo que inclui ferramentas de auditoria sem fio para quebrar chaves WEP, detectar pontos de acesso etc.

Os utilitários **bsd-airtools** podem ser instalados a partir do port net/bsd-airtools. Informações sobre como instalar ports podem ser encontradas em [Capítulo 4](#) do handbook.

O programa `dstumbler` é a ferramenta no pacote que permite a descoberta de pontos de acesso e representação gráfica da razão entre sinal e ruído. Se você estiver com dificuldades para ativar seu ponto de acesso, `dstumbler` pode ajudá-lo a familiarizar-se.

Para testar a segurança de sua rede sem fio, você pode escolher usar ```dweputils``` (`dwepccrack`, `dwepdump` e `dwepkeygen`) para ajudá-lo a determinar se WEP é a solução adequada para suas necessidades de segurança sem fio.

19.3.3.5.2. Os Utilitários *wicontrol*, *ancontrol* e *raycontrol*

Estas são as ferramentas que você deve usar para controlar como sua placa sem fio se comporta na rede sem fio. Nos exemplos acima, escolhemos usar [wicontrol\(8\)](#), uma vez que nossa placa sem fio é uma interface `wi0`. Se você tivesse um dispositivo sem fio Cisco, ele apareceria como `an0`, e assim você deveria usar [ancontrol\(8\)](#).

19.3.3.5.3. O Comando *ifconfig*

O [ifconfig\(8\)](#) pode ser usado para fazer muitas das mesmas configurações que [wicontrol\(8\)](#), entretanto é desprovido de algumas poucas opções. Verifique [ifconfig\(8\)](#) para opções e parâmetros de linha de comando.

19.3.3.6. Placas Suportadas

19.3.3.6.1. Pontos de Acesso

As únicas placas que são suportadas atualmente para modo BSS (como um ponto de acesso) são dispositivos baseados nos chipsets Prism 2, 2.5 ou 3. Para uma lista completa, veja [wi\(4\)](#).

19.3.3.6.2. Clientes

Quase todas as placas sem fio 802.11b são suportadas atualmente pelo FreeBSD. A maioria das placas baseadas em Prism, Spectrum24, Hermes, Aironet e Raylink vão funcionar como uma placa de rede em modo IBSS (ad-hoc, ponto-a-ponto e BSS).

19.4. Bluetooth

Escrito por Pav Lucistnik.

19.4.1. Introdução

Bluetooth é uma tecnologia sem fio para criar redes pessoais operando na banda não licenciada de 2.4GHz, com alcance de 10 metros. As redes geralmente são criadas sob medida para este propósito, a partir de dispositivos portáteis como telefones celulares, handhelds e laptops. Diferente de outras tecnologias sem fio populares, como Wi-Fi, o Bluetooth oferece perfis de serviço de alto nível, por exemplo: servidores de arquivos similares ao FTP, distribuição de arquivos, transporte de voz, emulação de linha serial e outras.

A pilha Bluetooth em FreeBSD é implementada usando o framework Netgraph (veja [netgraph\(4\)](#)). Uma ampla variedade de plugues USB Bluetooth é suportada pelo gerenciador [ng_ubt\(4\)](#). Os dispositivos baseado no chip Bluetooth Broadcom BCM2033 são suportados através dos gerenciadores [ubtbcmfw\(4\)](#) e [ng_ubt\(4\)](#). O cartão PC Bluetooth 3Com 3CRWB60-A é suportado pelo gerenciador [ng_bt3c\(4\)](#). Dispositivos Bluetooth baseados em serial e UART são suportados via [sio\(4\)](#), [ng_h4\(4\)](#) e [hcseriald\(8\)](#). Este capítulo descreve o uso do plugue Bluetooth USB. O suporte ao Bluetooth está disponível em sistemas com FreeBSD 5.0 e mais recentes.

19.4.2. Conectando o Dispositivo

Por padrão, os gerenciadores de dispositivos Bluetooth estão disponíveis na forma de módulos de kernel. Antes de ligar um dispositivo, você vai precisar carregar o gerenciador no kernel.

```
# kldload ng_ubt
```

Se o dispositivo Bluetooth está presente no sistema durante a inicialização, carregue o módulo de `/boot/loader.conf`.

```
ng_ubt_load="YES"
```

Conecte seu plugue USB. Uma saída similar à seguinte será exibida na console (ou no syslog).

```
ubt0: vendor 0x0a12 product 0x0001, rev 1.10/5.25, addr 2
```

```
ubt0: Interface 0 endpoints: interrupt=0x81, bulk-in=0x82, bulk-
out=0x2
ubt0: Interface 1 (alt.config 5) endpoints: isoc-in=0x83, isoc-
out=0x3,
      wMaxPacketSize=49, nframes=6, buffer size=294
```

Copie `/usr/share/examples/netgraph/bluetooth/rc.bluetooth` para algum lugar conveniente como `/etc/rc.bluetooth`. Este script é usado para iniciar e parar a pilha Bluetooth. É uma boa idéia parar a pilha antes de desplugar o dispositivo, mas, se o dispositivo for desplugado sem a parada da pilha, isto (geralmente) não é fatal. Quando iniciar a pilha, você poderá ver uma saída similar à seguinte:

```
# /etc/rc.bluetooth start ubt0
BD_ADDR: 00:02:72:00:d4:1a
Features: 0xff 0xff 0xf 00 00 00 00 00
<3-Slot> <5-Slot> <Encryption> <Slot offset>
<Timing accuracy> <Switch> <Hold mode> <Sniff mode>
<Park mode> <RSSI> <Channel quality> <SCO link>
<HV2 packets> <HV3 packets> <u-law log> <A-law log> <CVSD>
<Paging scheme> <Power control> <Transparent SCO data>
Max. ACL packet size: 192 bytes
Number of ACL packets: 8
Max. SCO packet size: 64 bytes
Number of SCO packets: 8
```

19.4.3. Interface Controladora de Sistema (HCI)

A Interface Controladora de Sistema, ou, Host Controller Interface (HCI) fornece uma interface de comando para o controlador de banda base e o gerenciador de ligação, e acesso ao estado do hardware e registradores de controle. Esta interface fornece um método uniforme de acesso às capacidades de banda base do Bluetooth. A camada HCI no Sistema troca dados e comandos com o firmware HCI no hardware Bluetooth. O gerenciador da Camada de Transporte Controladora do Sistema - Host Controller Transport Layer - (barramento físico) fornece a ambas as camadas HCI a capacidade de trocar informações com a outra.

Um único nó Netgraph do tipo `hci` é criado para um único dispositivo Bluetooth. O nó HCI normalmente é conectado ao nó do gerenciador de dispositivo Bluetooth (downstream) e ao nó L2CAP (upstream). Todas as operações HCI devem ser desempenhadas no nó HCI e não no nó do gerenciador de dispositivo. O nome padrão para o nó HCI é `devicehci`. Para maiores detalhes consulte a página de manual [ng hci\(4\)](#).

Uma das tarefas mais comuns é o descobrimento de dispositivos Bluetooth nas proximidades de radiofrequência, RF. Esta operação chama-se *investigação* (inquiry). A investigação e outras operações relativas a HCI são feitas com o utilitário [hcccontrol\(8\)](#). O exemplo abaixo exhibe como encontrar quais dispositivos Bluetooth estão dentro de alcance. Você deve receber a lista de dispositivos em poucos segundos. Note que um dispositivo remoto somente vai responder à investigação se estiver em modo "encontrável" (discoverable).

```
% hcccontrol -n ubt0hci inquiry
Inquiry result, num_responses=1
Inquiry result #0
```

```
BD_ADDR: 00:80:37:29:19:a4
Page Scan Rep. Mode: 0x1
Page Scan Period Mode: 00
Page Scan Mode: 00
Class: 52:02:04
Clock offset: 0x78ef
Inquiry complete. Status: No error [00]
```

O *BD_ADDR* é o endereço único de um dispositivo Bluetooth, similar ao endereço MAC de uma placa de rede. Este endereço é necessário para promover comunicações com um dispositivo. É possível designar um nome legível por humanos a um *BD_ADDR*. O arquivo `/etc/bluetooth/hosts` contém informação relativas aos sistemas Bluetooth conhecidos. O seguinte exemplo mostra como obter um nome legível por humanos que foi designado a um dispositivo remoto.

```
% hccontrol -n ubt0hci remote_name_request 00:80:37:29:19:a4
BD_ADDR: 00:80:37:29:19:a4
Name: Pav's T39
```

Se você executar uma investigação em um dispositivo Bluetooth remoto, ele irá encontrar seu computador como ```nome.do.seu.sistema (ubt0)```. O nome designado para o dispositivo local pode ser trocado a qualquer hora.

O sistema Bluetooth fornece conexão ponto-a-ponto (somente duas unidades envolvidas), ou uma conexão ponto-a-multiponto. Na conexão ponto-a-multiponto, a mesma é compartilhada entre diversos dispositivos Bluetooth. O exemplo a seguir mostra como obter a lista de conexões de banda base para o dispositivo local.

```
% hccontrol -n ubt0hci read_connection_list
Remote BD_ADDR      Handle Type Mode Role Encrypt Pending Queue State
00:80:37:29:19:a4   41 ACL   0 MAST  NONE      0      0 OPEN
```

Um *identificador de conexão (handle)* é útil quando é necessário o encerramento de uma conexão de banda base. Note que geralmente não é necessário se fazer isto manualmente. A pilha automaticamente irá encerrar conexões de banda base inativas.

```
# hccontrol -n ubt0hci disconnect 41
Connection handle: 41
Reason: Connection terminated by local host [0x16]
```

Consulte `hccontrol help` para uma listagem completa dos comandos HCI disponíveis. A maioria dos comandos HCI não requerem privilégios de superusuário.

19.4.4. Protocolo de Controle Lógico da Ligação e de Adaptação (L2CAP)

O Protocolo de Controle Lógico da Ligação e de Adaptação - Logical Link Control and Adaptation Protocol - (L2CAP), fornece serviços de dados orientados à conexão e desconectados (connectionless) aos protocolos dos níveis superiores com capacidade de multiplexação e operação de segmentação e remontagem. O L2CAP permite aos protocolos dos níveis superiores e aplicações transmitirem e receberem pacotes de dados L2CAP de até 64 kilobytes de comprimento.

O L2CAP é baseado em torno do conceito de *canais (channels)*. Um canal é uma conexão lógica sobre uma conexão de banda base. Cada canal está ligado a um único protocolo de modo muitos-para-um. Múltiplos canais podem ser ligados ao mesmo protocolo, mas um canal não pode ser ligado a múltiplos protocolos. Cada pacote L2CAP recebido em um canal é direcionado ao protocolo de nível superior apropriado. Múltiplos canais podem compartilhar a mesma conexão banda base.

Um único nó Netgraph do tipo *l2cap* é criado para um único dispositivo Bluetooth. O nó L2CAP geralmente é conectado a um nó Bluetooth HCI (downstream) e nós de dispositivos de envio/recepção (sockets) Bluetooth (upstream). O nome padrão para o nó L2CAP é ``device:l2cap". Para maiores detalhes consulte a página de manual [ng_l2cap\(4\)](#).

Um comando útil é o [l2ping\(8\)](#), que pode ser usado para testar (ping) outros dispositivos. Algumas implementações Bluetooth podem não responder a todos os dados a elas enviado, então o resultado *0 bytes* no exemplo abaixo é considerado normal.

```
# l2ping -a 00:80:37:29:19:a4
0 bytes from 0:80:37:29:19:a4 seq_no=0 time=48.633 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=1 time=37.551 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=2 time=28.324 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=3 time=46.150 ms result=0
```

O utilitário [l2control\(8\)](#) é usado para executar diversas operações em nós L2CAP. Este exemplo mostra como obter a lista de conexões lógicas (canais) e a lista de conexões banda base para o dispositivo local.

```
% l2control -a 00:02:72:00:d4:1a read_channel_list
L2CAP channels:
Remote BD_ADDR      SCID/ DCID   PSM IMTU/ OMTU State
00:07:e0:00:0b:ca   66/   64     3  132/  672 OPEN
% l2control -a 00:02:72:00:d4:1a read_connection_list
L2CAP connections:
Remote BD_ADDR      Handle Flags Pending State
00:07:e0:00:0b:ca   41  0     0  0 OPEN
```

Outra ferramenta de diagnóstico é [btsockstat\(1\)](#). Esta ferramenta faz um trabalho similar ao de [netstat\(1\)](#), mas para estruturas de dados Bluetooth relacionadas à rede. O exemplo abaixo mostra a mesma conexão lógica que [l2control\(8\)](#), acima.

```
% btsockstat
Active L2CAP sockets
PCB      Recv-Q Send-Q Local address/PSM      Foreign address  CID
State
c2afe900      0      0 00:02:72:00:d4:1a/3    00:07:e0:00:0b:ca 66
OPEN
Active RFCOMM sessions
L2PCB    PCB      Flag MTU   Out-Q DLCs State
c2afe900 c2b53380 1    127    0    Yes OPEN
Active RFCOMM sockets
PCB      Recv-Q Send-Q Local address      Foreign address  Chan DLCI
State
c2e8bc80      0    250 00:02:72:00:d4:1a 00:07:e0:00:0b:ca 3    6
OPEN
```

19.4.5. Protocolo RFCOMM

O protocolo RFCOMM fornece emulação de portas seriais sobre o protocolo L2CAP. O protocolo é baseado no padrão ETSI TS 07.10. O RFCOMM é um protocolo de transporte simples, com provisões adicionais para emular os nove circuitos das portas seriais RS-232 (EIA/TIA-232-E). O protocolo RFCOMM suporta até 60 conexões simultâneas (canais RFCOMM) entre dois dispositivos Bluetooth.

Para os propósitos do RFCOMM, um caminho completo de comunicação envolve duas aplicações rodando em dispositivos diferentes (as pontas da comunicação) com um segmento de comunicação entre eles. O RFCOMM é planejado para cobrir aplicações que utilizem as portas seriais dos dispositivos em que residem. O segmento de comunicação é uma ligação Bluetooth de um dispositivo para outro (conexão direta).

O RFCOMM preocupa-se somente com o caso da conexão direta entre os dispositivos, ou com o caso de rede entre um dispositivo e um modem. O RFCOMM pode suportar outras configurações, como módulos que comunicam-se via tecnologia sem fio Bluetooth em um lado, e fornecem uma interface cabeada na outra ponta.

No FreeBSD o protocolo RFCOMM é implementado na camada de sockets Bluetooth.

19.4.6. União de Dispositivos (Pairing)

Por padrão, a comunicação Bluetooth não é autenticada e qualquer dispositivo pode falar para qualquer outro dispositivo. Um dispositivo Bluetooth (por exemplo, um telefone celular) pode escolher exigir autenticação para fornecer um serviço particular (por exemplo, acesso discado). A autenticação no Bluetooth é feita normalmente com *código PIN*. Um código PIN é uma seqüência de caracteres ASCII de até 16 caracteres de comprimento. É preciso que o usuário entre com o mesmo código PIN em ambos os dispositivos. Uma vez que o usuário entrou com o código PIN, ambos os dispositivos vão gerar uma *chave de ligação (link key)*. Depois, a chave de ligação pode ser armazenada nos próprios dispositivos ou em um armazenamento persistente. Da próxima vez, ambos os dispositivos vão usar a chave de ligação previamente armazenada. O procedimento acima descrito é chamada de *união (pairing)*. Note que se a chave de ligação for perdida por qualquer dispositivo, então a união precisará ser repetida.

O daemon [hcsecd\(8\)](#) é responsável por tratar todas as solicitações de autenticação Bluetooth. O arquivo de configuração padrão é `/etc/bluetooth/hcsecd.conf`. Abaixo é mostrada uma seção de exemplo para um telefone celular com o código PIN configurado arbitrariamente para `1234`.

```
device {
    bdaddr 00:80:37:29:19:a4;
    name    "Pav's T39";
    key     nokey;
    pin     "1234";
}
```

Não há limites em códigos de PIN (exceto tamanho). Alguns dispositivos (por exemplo, fones Bluetooth) podem ter um código PIN fixo embutido. O parâmetro `-d` força o

daemon [hcsecd\(8\)](#) a se manter rodando em primeiro plano, de forma que seja fácil observar o que está acontecendo. Ajuste o dispositivo remoto para receber união e iniciar a conexão Bluetooth ao dispositivo remoto. O dispositivo remoto deveria dizer que a união foi aceita, e solicitar o código PIN. Entre o mesmo código PIN que você tem em `hcsecd.conf`. Agora seu PC e o dispositivo remoto estão unidos. Alternativamente, você pode iniciar a união no dispositivo remoto. Abaixo, a amostra da saída do comando `hcsecd`.

```
hcsecd[16484]: Got Link_Key_Request event from 'ubt0hci', remote
bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4,
name 'Pav's T39', link key doesn't exist
hcsecd[16484]: Sending Link_Key_Negative_Reply to 'ubt0hci' for remote
bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Got PIN_Code_Request event from 'ubt0hci', remote
bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4,
name 'Pav's T39', PIN code exists
hcsecd[16484]: Sending PIN_Code_Reply to 'ubt0hci' for remote bdaddr
0:80:37:29:19:a4
```

19.4.7. Protocolo de Descoberta de Serviço (SDP)

O Protocolo de Descoberta de Serviço - Service Discovery Protocol (SDP) - fornece os meios para as aplicações clientes descobrirem a existência de serviços fornecidos por aplicações servidoras, bem como os atributos destes serviços. Os atributos de um serviço incluem o tipo de classe de serviço oferecido e a informação do protocolo ou mecanismo necessários para utilizar o serviço.

O SDP envolve comunicação entre um servidor SDP e um cliente SDP. O servidor mantém uma lista de registros de serviços que descrevem as características dos serviços associados com o servidor. Cada registro de serviço contém informação sobre um único serviço. Um cliente pode recuperar informação de um serviço mantido pelo servidor SDP emitindo uma solicitação SDP. Se um cliente ou uma aplicação associada com o cliente decide usar um serviço, ele precisa abrir uma conexão separada ao provedor para utilizá-lo. O SDP fornece um mecanismo para descoberta de serviços e seus atributos, mas não fornece um mecanismo para utilização destes serviços.

Normalmente, um cliente SDP busca por serviços baseados em algumas características desejadas dos serviços. Entretanto, há momentos nos quais é desejável descobrir quais tipos de serviços estão descritos por registros de serviço de um servidor SDP sem qualquer informação a priori sobre os mesmos. Este processo de procurar por quaisquer serviços oferecidos é chamado de *navegação (browsing)*.

Atualmente, o cliente e o servidor SDP são implementados em uma pacote externo, **sdp-1.5**, que pode ser copiado [daqui](#). O **sdptool** é um cliente SDP de linha de comando. O seguinte exemplo mostra como executar uma solicitação de navegação SDP.

```
# sdptool browse 00:80:37:29:19:a4
Browsing 00:80:37:29:19:A4 ...
Service Name: Dial-up Networking
Protocol Descriptor List:
  "L2CAP" (0x0100)
```



```
"RFCOMM" (0x0003)
  Channel: 1
```

```
Service Name: Fax
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 2
```

```
Service Name: Voice gateway
Service Class ID List:
  "Headset Audio Gateway" (0x1112)
  "Generic Audio" (0x1203)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 3
```

...e assim por diante. Note que cada serviço tem uma lista de atributos (canal RFCOMM, por exemplo). Dependendo do serviço você pode precisar anotar alguns dos atributos. Algumas implementações Bluetooth não suportam navegação nos serviços, e podem retornar uma lista vazia. Neste caso, é possível buscar pelo serviço específico. O exemplo abaixo mostra como buscar pelo serviço OBEX Object Push (OPUSH).

```
# sdptool search --bdaddr 00:07:e0:00:0b:ca OPUSH
```

Oferecendo serviços em FreeBSD a clientes Bluetooth é feito com o servidor **sdpd**.

```
# sdpd
```

O **sdptool** também é usado para registrar um serviço no servidor SDP local. O exemplo abaixo mostra como registrar o serviço de Acesso à Rede com PPP (LAN). Repare que alguns serviços exigem atributos (canal RFCOMM, por exemplo).

```
# sdptool add --channel=7 LAN
```

A lista de serviços registrados com o servidor SDP local pode ser obtida emitindo uma requisição de navegação SDP a um BD_ADDR ``especial".

```
# sdptool browse ff:ff:ff:00:00:00
```

19.4.8. Perfis de Acesso discado (DUN) e Acesso à Rede com PPP (LAN)

O perfil de acesso discado - Dial-Up Networking (DUN) - é mais usado com modems e telefones celulares. Os cenários cobertos por este perfil são os seguintes:

- uso de um telefone celular ou modem por um computador como modem sem-fio (wireless) para conectar à Internet através de um servidor de acesso discado, ou usar outros serviços de acesso discado;
- uso de um telefone celular ou modem por um computador para receber chamadas de dados.

O perfil de Acesso à Rede com PPP (LAN) pode ser usado nas seguintes situações:

- Acesso na Rede Local (LAN) para um único dispositivo Bluetooth;
- Acesso à Rede Local para múltiplos dispositivos Bluetooth;
- PC a PC (usando rede PPP sobre emulação de cabo serial).

No FreeBSD ambos os perfis são implementados com [ppp\(8\)](#) e [rfcomm_pppd\(8\)](#) - uma cobertura (wrapper) que converte a conexão Bluetooth RFCOMM em algo com o qual o PPP pode operar. Antes que qualquer perfil possa ser usado, um novo rótulo PPP deve ser criado em `/etc/ppp/ppp.conf`. Consulte a página de manual [rfcomm_pppd\(8\)](#) para ver exemplos.

No seguinte exemplo, [rfcomm_pppd\(8\)](#) será usado para abrir uma conexão RFCOMM ao dispositivo remoto com o BD_ADDR 00:80:37:29:19:a4 no canal RFCOMM DUN. O número do canal RFCOMM atual será obtido do dispositivo remoto via SDP. É possível especificar o canal RFCOMM manualmente, e, neste caso, [rfcomm_pppd\(8\)](#) não irá executar uma requisição SDP. Use **sdptool** para descobrir o canal RFCOMM no dispositivo remoto.

```
# rfcomm_pppd -a 00:80:37:29:19:a4 -c -C dun -l rfcomm-dialup
```

Para fornecer o serviço de Acesso à Rede com PPP (LAN), o servidor **sdpd** deve estar sendo executado. Também é necessário registrar o serviço de Rede Local com o servidor SDP local. Note que o serviço de rede exige o atributo do canal RFCOMM. Uma nova entrada para clientes da Rede Local precisa ser criada no arquivo `/etc/ppp/ppp.conf`. Consulte a página de manual [rfcomm_pppd\(8\)](#) para ver exemplos. Finalmente, o servidor PPP RFCOMM precisa estar rodando e ouvindo no mesmo canal RFCOMM como foi registrado com o servidor SDP local. O exemplo abaixo mostra como iniciar o servidor PPP RFCOMM.

```
# rfcomm_pppd -s -C 7 -l rfcomm-server
```

19.4.9. Perfil OBEX Push (OPUSH)

O OBEX é um protocolo amplamente usado para transferências de arquivos simples entre dispositivos móveis. Seu principal uso é nas comunicação em infravermelho, onde é usado para transferências de arquivos genéricos entre notebooks ou dispositivos Palm, e para enviar cartões de visita ou entradas em calendários entre telefones celulares e outros dispositivos com aplicações de Gerenciamento de Informações Pessoal (PIM).

O cliente e o servidor OBEX são implementados como pacotes de terceiros **obexapp-1.0** que pode ser copiada [daqui](#). O pacote requer a biblioteca **openobex** (já incluída) e o port [devel/glib12](#). Note que **obexapp** não requer privilégios de superusuário (root) para operar.

O cliente OBEX é usado para remeter e/ou copiar objetos do servidor OBEX. Um objeto pode, por exemplo, ser um cartão de visitas ou um agendamento. O cliente OBEX pode obter um número de canal RFCOMM do dispositivo remoto via SDP. Isto pode ser feito especificando-se o nome do serviço ao invés do número do canal RFCOMM. Nomes de dispositivos suportados são: IrMC, FTRN e OPUSH. É possível especificar o canal RFCOMM como um número. Abaixo está um exemplo de sessão

OBEX, onde o objeto de informação do dispositivo é copiado do telefone celular e, um novo objeto (cartão de visita) é remetido à agenda do telefone.

```
% obexapp -a 00:80:37:29:19:a4 -C IrMC
obex> get
get: remote file> telecom/devinfo.txt
get: local file> devinfo-t39.txt
Success, response: OK, Success (0x20)
obex> put
put: local file> new.vcf
put: remote file> new.vcf
Success, response: OK, Success (0x20)
obex> di
Success, response: OK, Success (0x20)
```

Para fornecer o serviço de envio OBEX, o servidor **sdpd** precisa estar sendo executado. Também é necessário registrar o serviço OPUSH no servidor SDP local. Note que o serviço OPUSH requer o atributo do canal RFCOMM. Uma pasta raiz, onde todos os objetos recebidos serão armazenados, precisa ser criado. O caminho padrão para a pasta raiz é `/var/spool/obex`. Finalmente, o servidor OBEX precisa estar rodando e ouvindo no mesmo canal RFCOMM como registrado com o servidor SDP local. O exemplo abaixo mostra como iniciar o servidor OBEX.

```
# obexapp -s -C 10
```

19.4.10. Perfil da Porta Serial (SP)

A porta serial - Serial Port (SP) - permite a um dispositivo Bluetooth executar uma emulação de cabo serial RS232 (ou similar). O cenário abordado neste perfil trata de aplicações legadas usando o Bluetooth como substituto do cabo, através de uma abstração de porta serial virtual.

O utilitário [rfcomm_sppd\(1\)](#) implementa o perfil de Porta Serial. Um pseudo tty é usado como uma abstração de porta serial virtual. O exemplo abaixo mostra que você não tem que especificar o canal RFCOMM - [rfcomm_sppd\(1\)](#) pode obtê-lo do dispositivo remoto via SDP. Se você quiser ativar manualmente isto, especifique o canal RFCOMM na linha de comando.

```
# rfcomm_sppd -a 00:07:E0:00:0B:CA -t /dev/tty6
rfcomm_sppd[94692]: Starting on /dev/tty6...
```

Uma vez conectado, o pseudo tty pode ser usado como porta serial.

```
# cu -l tty6
```

19.4.11. Localização de Defeitos

19.4.11.1. Um dispositivo remoto não consegue conectar-se

Alguns dispositivos Bluetooth mais antigos não suportam troca de função. Por padrão, quando o FreeBSD está aceitando uma nova conexão, ele tenta efetuar uma troca de função e tornar-se um mestre. Os dispositivos que não suportarem esta característica, não serão capazes de se conectar. Note que a troca de função é executada quando uma

nova conexão está sendo estabelecida, de forma que não seja possível perguntar ao dispositivo remoto se ele suporta troca de função. Há uma opção HCI para desabilitar a troca de função na ponta local.

```
# hccontrol -n ubt0hci write_node_role_switch 0
```

19.4.11.2. Algo está errado, posso ver o que exatamente está acontecendo?

Sim, você pode. Use o pacote de terceiros **hcidump-1.5**, que pode ser copiado [daqui](#). O utilitário **hcidump** é similar ao [tcpdump\(1\)](#). Ele pode ser usado para mostrar o conteúdo dos pacotes Bluetooth no terminal, e jogar pacotes em um arquivo.

19.5. Ligação (Bridging)

Escrito por Steve Peterson.

19.5.1. Introdução

Às vezes é útil dividir uma rede física (como um segmento Ethernet) em dois segmentos sem ter que criar subredes IP e usar um roteador para conectar os segmentos. Um dispositivo que conecta duas redes juntas desta maneira é chamado de "ligação", ou "ponte". Um sistema FreeBSD com duas interfaces de rede pode atuar como uma ligação.

A ligação funciona aprendendo os endereços da camada MAC (endereços Ethernet) dos dispositivos em cada uma de suas interfaces de rede. Ela repassa o tráfego entre duas redes somente quando sua origem e destino estão em redes diferentes.

Em muitos aspectos, uma ligação é como um comutador Ethernet com muito poucas portas.

19.5.2. Situações Convenientes Para Uso de Ligação

Atualmente, existem duas situações comuns nas quais ligações são implementadas.

19.5.2.1. Alto Tráfego em um Segmento

A primeira situação é onde seu segmento de rede físico está sobrecarregado com tráfego mas você não quer, por quaisquer razões, subdividir a rede e interconectar as subredes com um roteador.

Vamos considerar um exemplo de um jornal onde os departamentos Editorial e de Produção estão na mesma subrede. Todos os usuários do Editorial usam o servidor A para servir arquivos e os usuários de Produção estão no servidor B. Uma Ethernet é usada para conectar todos os usuários e altas cargas de tráfego na rede estão degradando o desempenho.

Se os usuários do Editorial pudessem ser segregados em um segmento de rede e os usuários da Produção em outra, ambos segmentos de rede podem ser conectados com uma ligação. Somente o tráfego destinado a interfaces no "outro" lado da ligação poderia ser enviado para a outra rede, reduzindo a congestão em cada segmento de rede.

19.5.2.2. Filtering/Traffic Shaping Firewall

A segunda situação comum é aquela onde a funcionalidade de firewall é necessária sem uso de mascaramento de IP (NAT).

Um exemplo é uma pequena companhia conectada via DSL ou ISDN ao seu PSI. Eles possuem 13 endereços IP de seu PSI globalmente acessíveis e 10 PCs em sua rede. Nesta situação, é difícil usar um firewall baseado em roteador por questões de divisões de subrede.

Um firewall baseado em ponte (bridge) pode ser configurado e inserido no caminho somente de entrada do roteador DSL/ISDN sem quaisquer questões de numeração IP.

19.5.3. Configurando uma Ponte (Bridge)

19.5.3.1. Seleção de Placa Interface de Rede

Uma ponte requer pelo menos duas interfaces de rede para funcionar. Infelizmente, nem todas as placas interface de rede da época do FreeBSD 4.0 suportam ligações. Leia [bridge\(4\)](#) para detalhes sobre placas suportadas.

Instale e teste as duas placas de rede antes de continuar.

19.5.3.2. Alterações nas Configurações do Kernel

Para ativar suporte no kernel para ligações, adicione:

```
options BRIDGE
```

ao seu arquivo de configuração do kernel e recompile o mesmo.

19.5.3.3. Suporte a Firewall

Se você estiver planejando usar a ponte como firewall, você também vai precisar adicionar a opção *IPFIREWALL*. Leia [Seção 10.8](#) para informações gerais sobre configuração de ponte como firewall.

Se precisar permitir pacotes não-IP (como ARP) trafegarem através da ponte, há uma opção não documentada de firewall que precisa ser ativada. Esta opção é *IPFIREWALL_DEFAULT_TO_ACCEPT*. Note que isto altera a regra padrão para que o firewall aceite qualquer pacote. Certifique-se de que compreende como isto muda o sentido de seu conjunto de regras antes de ativá-la.

19.5.3.4. Suporte a Formatação de Tráfego (Traffic Shaping)

Se você desejar usar a ponte como formatador de tráfego, você vai precisar adicionar a opção *DUMMYNET* à sua configuração de kernel. Leia [dummynet\(4\)](#) para maiores informações.

19.5.4. Ativando a Ponte

Adicione a linha:

```
net.link.ether.bridge=1
```

ao `/etc/sysctl.conf` para ativar a ponte em tempo de execução, e a linha:

```
net.link.ether.bridge_cfg=if1,if2
```

para ativar a ligação nas interfaces especificadas (substitua *if1* e *if2* com os nomes de suas duas interfaces de rede). Se quiser que os pacotes de ligação sejam filtrados por [ipfw\(8\)](#), você também deve adicionar:

```
net.link.ether.bridge_ipfw=1
```

ao arquivo de configuração.

19.5.5. Outras Informações

Se você precisar efetuar um telnet para a ponte a partir da rede, não há problemas em se atribuir um endereço IP a uma das placas de interface de rede. O consenso é que não é boa idéia atribuir endereços a ambas as placas.

Se você tem múltiplas pontes em sua rede, não pode haver mais que um caminho entre duas estações de trabalho. Tecnicamente, isto significa que não há suporte para gerenciamento de enlaces spanning tree.

Uma ponte pode acrescentar latência ao tempo de resposta de pings, especialmente no tráfego entre segmentos.

19.6. NFS

Reorganizado e aumentado por Tom Rhodes. Escrito por Bill Swingle.

Dentre os diversos sistemas de arquivos suportados pelo FreeBSD está o Sistema de Arquivo em Rede (Network File System), também conhecido como NFS. O NFS permite que um sistema compartilhe seus diretórios e arquivos com outros sistemas através de uma rede. Usando NFS, os usuários e programas podem acessar arquivos em sistemas remotos quase como se fossem arquivos locais.

Alguns dos benefícios mais notáveis que o NFS pode oferecer são:

- Estações locais usam menos espaço em disco porque dados freqüentemente usados podem ser armazenados em uma única máquina e ainda permanecerem acessíveis a outras pela rede.
- Não há necessidade de usuários terem diretórios pessoais separados em cada máquina da rede. Diretórios pessoais podem ser configurados no servidor NFS e serem disponibilizados através da rede.
- Dispositivos de armazenamento como disquetes, leitores de CD-ROM e leitores de ZIP podem ser usados por outras máquinas na rede. Isto pode reduzir o número de leitores de mídia removível em toda a rede.

19.6.1. Como o NFS Funciona

O NFS consiste de, pelo menos, duas partes principais: um servidor e um ou mais clientes. O cliente acessa remotamente os dados armazenados na máquina servidora. Para que isto funcione direito, alguns poucos processos precisam estar configurados e rodando:

Nota: No FreeBSD 5.X, o utilitário **portmap** foi substituído pelo utilitário `rpcbind`. Assim, no FreeBSD 5.X é preciso que o usuário substitua cada instância do **portmap** pelo `rpcbind` nos exemplos a seguir.

O servidor precisa ter rodando os seguintes daemons:

Daemon	Descrição
<code>nfsd</code>	O daemon NFS, que atende requisições dos clientes NFS.
<code>mountd</code>	O daemon de montagem NFS, que executa as solicitações que o nfsd(8) lhe passa.
<code>portmap</code>	O daemon portmapper permite que clientes NFS descubram qual porta o servidor NFS está utilizando.

O cliente também pode rodar um daemon, conhecido como **nfsiod**. O daemon **nfsiod** atende às solicitações do servidor NFS. Isto é opcional e aumenta o desempenho, mas não é requerido para a operação normal e correta. Veja a página de manual [nfsiod\(8\)](#) para mais informações.

19.6.2. Configurando o NFS

A configuração do NFS é um processo relativamente claro e direto. Os processos que precisam estar rodando podem todos ser ativados no momento da inicialização do sistema, com poucas modificações no seu arquivo `/etc/rc.conf`.

No servidor NFS, certifique-se de que as seguintes opções estão configuradas no arquivo `/etc/rc.conf`:

```
portmap_enable="YES"
nfs_server_enable="YES"
mountd_flags="-r"
```

O `mountd` é automaticamente executado quando o servidor NFS é ativado.

No cliente, certifique-se de que esta opção está presente no `/etc/rc.conf`:

```
nfs_client_enable="YES"
```

O arquivo `/etc/exports` especifica quais sistemas de arquivos o NFS deve exportar (às vezes chamados de ``compartilhamentos"). Cada linha no `/etc/exports` determina um sistema de arquivo a ser exportado e quais máquinas têm acesso àquele sistema de arquivo. Junto com quais máquinas têm acesso ao sistema de arquivos, opções de acesso também podem ser especificadas. Existem muitas opções que podem ser usadas neste arquivo, mas somente umas poucas serão mencionadas aqui. Você pode facilmente descobrir outras opções lendo a página de manual [exports\(5\)](#).

Aqui estão uns poucos exemplos de entradas no `/etc/exports`:

Os seguintes exemplos dão uma idéia de como exportar sistemas de arquivos, ainda que os parâmetros possam ser diferentes, dependendo do seu ambiente e das configurações de rede. Por exemplo, para exportar o diretório `/cdrom` para três máquinas de exemplo, que possuem o mesmo nome de domínio que o servidor (por conseguinte a ausência do nome de domínio em cada uma) ou possuem entradas no seu arquivo `/etc/hosts`. O parâmetro `-ro` torna o sistema de arquivo exportado somente-para-leitura (read-only). Com este parâmetro, o sistema remoto não será capaz de gravar quaisquer mudanças no sistema de arquivo exportado.

```
/cdrom -ro host1 host2 host3
```

A linha abaixo exporta o `/home` para três sistemas através de seus endereços IP. Esta é uma configuração útil, se você tem uma rede privada, sem servidor DNS configurado. Opcionalmente, o arquivo `/etc/hosts` poderia ser configurado para nomear os sistemas internos; por favor reveja [hosts\(5\)](#) para maiores informações. O parâmetro `-alldirs` permite que os subdiretórios sejam pontos de montagem. Em outras palavras, não vai montar os subdiretórios, mas vai permitir que o cliente monte somente os diretórios que são requeridos os necessários.

```
/home -alldirs 10.0.0.2 10.0.0.3 10.0.0.4
```

A seguinte linha exporta `/a` de forma que dois clientes de diferentes domínios acessem o sistema de arquivos. O parâmetro `-maproot=root` permite que o usuário `root` do sistema remoto grave dados no sistema de arquivos exportado como `root`. Se o parâmetro `-maproot=root` não for especificado, então, mesmo que um usuário tenha acesso de `root` no sistema remoto, ele não será capaz de modificar arquivos no sistema de arquivos exportado.

```
/a -maproot=root host.exemplo.com box.exemplo.org
```


Para que um cliente tenha acesso a um sistema de arquivos exportado, o cliente precisa ter permissão para tal. Certifique-se de que o cliente está listado no seu arquivo `/etc/exports`.

No `/etc/exports`, cada linha representa a informação de exportação para um sistema de arquivo para um sistema. Um sistema remoto pode ser especificado somente uma vez por sistema de arquivo, e pode ter somente uma entrada padrão. Por exemplo, assuma que `/usr` é um único sistema de arquivo. O `/etc/exports` abaixo seria inválido:

```
/usr/src cliente
/usr/ports cliente
```

Um sistema de arquivo, `/usr`, tem duas linhas especificando exportações para o mesmo sistema, `cliente`. O formato correto para esta situação é:

```
/usr/src /usr/ports cliente
```

As propriedades de um sistema de arquivo exportado para um dado sistema precisa estar em uma mesma linha. Linhas sem um cliente especificado são tratadas como um único sistema. Isto limita como você pode exportar sistemas de arquivos, mas para a maioria das pessoas isto não é um problema.

Abaixo segue um exemplo de uma lista de exportação válida, onde `/usr` e `/exports` são sistemas de arquivos locais:

```
# Exporte src e ports ao cliente01 e cliente02, mas somente o
# cliente01 tem privilégios de root nele
/usr/src /usr/ports -maproot=root cliente01
/usr/src /usr/ports cliente02
# As máquinas cliente possuem acesso de root e podem montar em
# qualquer lugar de /exports. Qualquer um no mundo pode montar
/exports/obj somente-leitura.
/exports -alldirs -maproot=root cliente01 cliente02
/exports/obj -ro
```

Você precisa reiniciar o `mountd` todas as vezes que modificar o `/etc/exports` de forma que as mudanças tenham efeito. Isto pode ser executado enviando o sinal HUP ao processo `mountd`:

```
# kill -HUP `cat /var/run/mountd.pid`
```

Alternativamente, uma reinicialização (reboot) fará o FreeBSD configurar tudo corretamente. Entretanto, uma reinicialização não é necessária. Executando o seguinte comando como `root` deverá ativar tudo.

No servidor NFS:

```
# portmap
# nfsd -u -t -n 4
# mountd -r
```

No cliente NFS:

```
# nfsiod -n 4
```

Agora tudo deve estar pronto para, na prática, montar um sistema de arquivo remoto. Nestes exemplos, o nome do servidor será *server* e o nome do cliente será *client*. Se você quer somente montar temporariamente um sistema de arquivo ou prefere testar a configuração, apenas execute um comando como este como `root` no cliente:

```
# mount server:/home /mnt
```

Isto irá montar o diretório `/home` do servidor em `/mnt` no cliente. Se tudo estiver configurado corretamente, você deve ser capaz de entrar no `/mnt` no cliente e ver todos os arquivos que estão no servidor.

Se você quer montar automaticamente um sistema de arquivo remoto cada vez que o computador iniciar, adicione o sistema de arquivos no arquivo `/etc/fstab`. Eis um exemplo:

```
server:/home /mnt nfs rw 0 0
```

A página de manual [fstab\(5\)](#) lista todas as opções disponíveis.

19.6.3. Usos Práticos

O NFS tem muitos usos práticos. Alguns dos usos mais comuns estão listados abaixo:

- Configurar diversas máquinas para compartilhar um CDROM ou outra mídia entre eles. Isto é mais barato e frequentemente um método mais conveniente de instalar software em múltiplas máquinas.
- Em grandes redes, pode ser mais conveniente configurar um servidor central para armazenar todos os diretórios pessoais dos usuários. Estes diretórios pessoais então podem ser exportados para a rede, de forma que os usuários sempre tenham o mesmo diretório pessoal, sem se preocupar com qual estação de trabalho eles utilizem.
- Diversas máquinas poderiam ter um diretório `/usr/ports/distfiles` comum. Desta forma, quando você precisar instalar um port em diversas máquinas, você pode rapidamente acessar o fonte sem precisar baixá-lo em cada máquina.

19.6.4. Montagens Automáticas com amd

Contribuído por Wylie Stilwell. Reescrito por Chern Lee.

O daemon de automontagem, [amd\(8\)](#) (automatic mounter daemon), monta automaticamente um sistema de arquivo remoto todas as vezes que um arquivo ou diretório naquele sistema de arquivo é acessado. Sistemas de arquivos que ficam inativos por um período de tempo também serão automaticamente desmontados pelo **amd**. Usar o **amd** é uma alternativa para montagens permanentes, uma vez que as montagens permanentes geralmente estão listadas no `/etc/fstab`.

O **amd** opera anexando-se como um servidor NFS aos diretórios `/host` e `/net`. Quando um arquivo é acessado dentro destes diretórios, o **amd** procura pelo ponto de montagem remoto correspondente e o monta automaticamente. O `/net` é usado para montar um

sistema de arquivo exportado de um endereço IP, enquanto o `/host` é usado para montar uma exportação de um nome de sistema remoto.

Um acesso a um arquivo dentro de `/host/foobar/usr` diria ao **amd** para tentar montar a exportação `/usr` no sistema `foobar`.

Exemplo 19-1. Montando uma Exportação com amd

Você pode ver as montagens disponíveis de um sistema remoto com o comando `showmount`. Por exemplo, para ver as montagens de um sistema chamado `foobar`, você pode usar:

```
% showmount -e foobar
Exports list on foobar:
/usr                10.10.10.0
/a                 10.10.10.0
% cd /host/foobar/usr
```

Como visto no exemplo, o `showmount` mostra `/usr` como uma exportação. Quando trocar de diretórios para `/host/foobar/usr`, o **amd** tenta resolver o nome do sistema `foobar` e automaticamente monta a exportação desejada.

O **amd** pode ser iniciado pelos scripts de inicialização colocando as seguintes linhas no `/etc/rc.conf`:

```
amd_enable="YES"
```

Adicionalmente, parâmetros personalizados podem ser passados ao **amd** pela opção `amd_flags`. Por padrão, o `amd_flags` é ajustado para:

```
amd_flags="-a /.amd_mnt -l syslog /host /etc/amd.map /net
/etc/amd.map"
```

O arquivo `/etc/amd.map` define as opções padrão com as quais as exportações serão montadas. O arquivo `/etc/amd.conf` define algumas das características mais avançadas do **amd**.

Consulte as páginas de manual [amd\(8\)](#) e [amd.conf\(5\)](#) para maiores informações.

19.6.5. Problemas na Integração com Outros Sistemas

Contribuído por John Lind.

Certos adaptadores Ethernet para sistemas PC ISA possuem limitações que podem causar sérios problemas de rede, particularmente com NFS. Esta dificuldade não é específica do FreeBSD, mas sistemas FreeBSD são afetados por ela.

O problema quase sempre ocorre quando sistemas PC (FreeBSD) são conectados em rede com estações de trabalho de alto desempenho com as feitas pela Silicon Graphics, Inc., e Sun Microsystems, Inc. As montagens NFS funcionarão bem, e algumas operações podem ser bem sucedidas, mas repentinamente, o servidor vai parecer não mais responder ao cliente, mesmo que as solicitações de e para outros sistemas

continuem a ser processadas. Isto acontece ao sistema cliente, se o cliente for o sistema FreeBSD ou a estação de trabalho. Em muitos sistemas, não há forma de parar o sistema elegantemente, uma vez que este problema se manifestou. A única solução é, freqüentemente, reinicializar o cliente, porque a situação do NFS não pode ser resolvida.

Apesar da solução ``correta" ser a de obter um adaptador de maior desempenho e capacidade para o sistema FreeBSD, há um contorno simples que vai permitir a operação satisfatória. Se o sistema FreeBSD for o *servidor*, inclua a opção `-w=1024` na montagem a partir do cliente. Se o sistema FreeBSD é o *cliente*, então monte o sistema de arquivo NFS com a opção `-r=1024`. Estas opções podem ser especificadas usando o quarto campo da entrada `fstab` no cliente para montagens automáticas, ou, usando o parâmetro `-o` do comando `mount` para montagens manuais.

Deve ser observado que há um problema diferente, algumas vezes confundido com este, quando os servidores e clientes NFS estão em redes diferentes. Se for este o caso, esteja *certo* de que seus roteadores estão roteando a informação UDP necessária, ou você não vai chegar a lugar algum, não importando o que mais você estiver fazendo.

Nos seguintes exemplos, `fastws` é o nome do sistema (interface) de uma estação de trabalho de alto desempenho, e `freebox` é o nome do sistema (interface) de um sistema FreeBSD com um adaptador Ethernet de baixo desempenho. Também, `/sharedfs` será o sistema de arquivo NFS exportado (veja [exports\(5\)](#)), e `/project` será o ponto de montagem no cliente para o sistema exportado. Em todos os casos, note que opções adicionais, como *hard* ou *soft* e *bg* podem ser desejáveis em sua aplicação.

Exemplos para o sistema FreeBSD (`freebox`) como sendo o cliente no `/etc/fstab` de `freebox`:

```
fastws:/sharedfs /project nfs rw,-r=1024 0 0
```

Como um comando para montagem manual em `freebox`:

```
# mount -t nfs -o -r=1024 fastws:/sharedfs /project
```

Exemplos para o sistema FreeBSD como o servidor no `/etc/fstab` em `fastws`:

```
freebox:/sharedfs /project nfs rw,-w=1024 0 0
```

Como comando para montagem manual em `fastws`:

```
# mount -t nfs -o -w=1024 freebox:/sharedfs /project
```

Quase qualquer adaptador Ethernet permitirá operação sem as restrições acima no tamanho de leitura ou escrita.

Para qualquer um que se importe, aqui está o que acontece quando a falha ocorre, o que também explica porque é irrecoverável. O NFS trabalha tipicamente com um tamanho de ``bloco" de 8 k (apesar de que pode criar fragmentos de tamanhos menores). Uma vez que o tamanho máximo do quadro Ethernet é em torno de 1500 bytes, o ``bloco" NFS é dividido em múltiplos quadros Ethernet, mesmo que ainda seja uma única

unidade para o código de nível superior, e precisa ser recebido, montado e *confirmado* (*acknowledged*) como uma unidade. As estações de alto desempenho podem bombear os pacotes que compõem a unidade NFS para fora, um após o outro, tão próximos quanto permitido pelo padrão. Nas interfaces menores, de baixa capacidade, os últimos pacotes ultrapassam os primeiros pacotes da mesma unidade antes que possam ser reconstruídos ou confirmados. Como resultado, a estação de trabalho vai expirar e tentar novamente, mas tentará novamente com a unidade de 8 K inteira, e o processo se repetirá, ao infinito.

Mantendo o tamanho da unidade menor que a limitação do tamanho do quadro Ethernet, garantimos que qualquer quadro Ethernet completo recebido pode ser confirmado individualmente, evitando a situação de congelamento (deadlock).

Ultrapassagens ainda poderão ocorrer quando uma estações de alto desempenho estiverem golpeando dados para um sistema PC, mas com placas melhores, tais ultrapassagens não são garantidas nas "unidades" NFS. Quando uma ultrapassagem acontecer, as unidades afetadas serão retransmitidas, e haverá uma boa chance de que serão recebidas, montadas e confirmadas.

19.7. Operação Sem Disco (Diskless Operation)

Atualizado por Jean-François Dockès.

Uma máquina FreeBSD pode ser inicializada pela rede e operar sem um disco local, usando sistemas de arquivos montados de um servidor NFS. Nenhuma modificação é necessária no sistema, além dos arquivos de configuração. Tal sistema é fácil de configurar porque todos os elementos necessários estão prontamente disponíveis:

- Existem pelo menos dois métodos possíveis para carregar o kernel através da rede:
 - *PXE*: O sistema Ambiente de Execução Pré-inicialização (Preboot Execution Environment) é uma forma de ROM de partida inteligente embutida em algumas placas de rede ou placas-mãe. Veja [pxeboot\(8\)](#) para maiores detalhes.
 - O port **etherboot** ([net/etherboot](#)) produz código gravável em ROM para inicializar kernels através da rede. O código pode ser gravado em uma PROM de inicialização em uma interface de rede, ou carregada de um leitor de disquetes (ou disco rígido) local, ou de um sistema rodando MS-DOS®. Muitas placas de rede são suportadas.
- Um script de amostra (`/usr/share/examples/diskless/clone_root`) facilita a criação e manutenção do sistema de arquivo raiz da estação de trabalho no servidor. O script provavelmente vai requerer uma pequena personalização, mas vai lhe permitir começar muito rapidamente.
- Os arquivos de inicialização padrão do sistema estão em `/etc` para detectar e suportar a partida de um sistema sem disco.

- Se necessário fazer swapping, poderá ser feito em um arquivo NFS ou em um disco local.

Existem muitas formas de configurar estações sem disco. Muitos elementos estão envolvidos, e a maioria pode ser personalizada para atender uma preferência local. A seguir será descrita a configuração de um sistema completo, enfatizando a simplicidade e compatibilidade com os scripts padrão do FreeBSD. O sistema descrito tem as seguintes características:

- As estações sem disco usam sistemas de arquivos `raiz` e `/usr` somente-leitura compartilhados.

O sistema de arquivo `raiz` é uma cópia de uma raiz padrão do FreeBSD (tipicamente a do servidor), com alguns arquivos de configuração cancelados por alguns específicos para operação sem disco ou, possivelmente, para a estação de trabalho à qual pertencem.

As partes da `raiz` que precisam ser graváveis são cobertas com sistemas de arquivos [mfs\(8\)](#). Quaisquer mudanças serão perdidas quando o sistema for reinicializado.

- O kernel é carregado pelo **etherboot**, usando DHCP (ou BOOTP) e TFTP.

Cuidado Como descrito, este sistema é inseguro. Ele deve morar em uma área protegida da rede, e não ser confiado por outras máquinas.

19.7.1. Instruções de Configuração

19.7.1.1. Configurando o DHCP/BOOTP

Existem dois protocolos que são comumente usados para inicializar uma estação de trabalho que recupera seus arquivos de configuração através da rede: BOOTP e DHCP. Eles são usados em diversos pontos na partida da estação de trabalho:

- O **etherboot** usa o DHCP (por padrão) ou BOOTP (precisa de uma opção de configuração) para encontrar o kernel. (PXE usa DHCP).
- O kernel usa BOOTP para localizar a raiz NFS.

É possível configurar um sistema para usar somente BOOTP. O programa servidor [bootpd\(8\)](#) está incluído no sistema base do FreeBSD.

Entretanto, o DHCP tem diversas vantagens sobre o BOOTP (melhores arquivos de configuração, possibilidade de usar PXE, mais muitas outras não diretamente relacionadas à operação sem disco), e nós vamos descrever ambas as configurações, BOOTP puro e BOOTP+DHCP, com ênfase no último, o qual vai usar o pacote de software DHCP da ISC.

19.7.1.1.1. Configuração Usando o DHCP da ISC

O servidor **isc-dhcp** pode responder tanto requisições BOOTP como DHCP.

Desde o lançamento (release) 4.4, o **isc-dhcp 3.0** não é parte do sistema base. Você primeiro vai precisar instalar o port [net/isc-dhcp3](#) ou o pacote (package) correspondente. Por favor, veja [Capítulo 4](#) para informações gerais sobre ports e packages.

Uma vez que o **isc-dhcp** estiver instalado, ele precisa de um arquivo de configuração para rodar, (normalmente chamado `/usr/local/etc/dhcpd.conf`). Aqui segue um exemplo comentado:

```
default-lease-time 600;
max-lease-time 7200;
authoritative;

option domain-name "exemplo.com";
option domain-name-servers 192.168.4.1;
option routers 192.168.4.1;

subnet 192.168.4.0 netmask 255.255.255.0 {
    use-host-decl-names on; ❶
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.4.255;

    host margaux {
        hardware ethernet 01:23:45:67:89:ab;
        fixed-address margaux.exemplo.com;
        next-server 192.168.4.4; ❷
        filename "/tftpboot/kernel.diskless"; ❸
        option root-path "192.168.4.4:/data/misc/diskless"; ❹
    }
}
```

❶

Esta opção diz ao `dhcpd` para enviar o valor nas declarações *host* como o nome do sistema para a máquina sem disco. Uma alternativa seria adicionar um *option hostname margaux* dentro das declarações *host*.

❷

A diretiva *next-server* designa o servidor TFTP (o padrão é usar o mesmo sistema que o servidor DHCP).

❸

A diretiva *filename* define que o arquivo que o **etherboot** vai carregar como o kernel.

Nota: O PXE parece preferir um nome de arquivo relativo, e ele carrega `pxeboot`, não o kernel (*option filename "pxeboot"*).

❹

A opção *root-path* define o caminho do sistema de arquivo raiz, na notação NFS usual.

19.7.1.1.2. Configuração Usando o BOOTP

Aqui segue uma configuração `bootpd` equivalente. Isto seria encontrado em `/etc/bootptab`.

Por favor, note que o **etherboot** precisa ser compilado com a opção não padrão `NO_DHCP_SUPPORT` para usar o BOOTP, e que o PXE *precisa* do DHCP. A única vantagem óbvia do **bootpd** é que ele existe no sistema base.

```
.def100:\
:hn:ht=1:sa=192.168.4.4:vm=rfc1048:\
:sm=255.255.255.0:\
:ds=192.168.4.1:\
:gw=192.168.4.1:\
:hd="/tftpboot":\
:bf="/kernel.diskless":\
:rp="192.168.4.4:/data/misc/diskless":

margaux:ha=0123456789ab:tc=.def100
```

19.7.1.2. Preparando um Programa de Boot com Etherboot

O [Sítio Web do Etherboot](#) contém [extensa documentação](#) direcionada principalmente a sistemas Linux, mas apesar de tudo, contendo informações úteis. O texto a seguir vai apenas delinear como você usaria o **etherboot** em um sistema FreeBSD.

Primeiro você precisa instalar o port ou pacote [net/etherboot](#). O port **etherboot** pode normalmente ser encontrado em `/usr/ports/net/etherboot`. Se a árvore dos ports está instalada em seu sistema, apenas digitando *make* neste diretório deve resolver tudo. Do contrário, veja [Capítulo 4](#) para informações sobre ports e pacotes (packages).

Para nossa configuração, usaremos um disquete de inicialização. Para outros métodos (PROM, ou programa DOS), veja, por favor, a documentação do **etherboot**.

Para fazer um disquete de inicialização, insira o disquete no leitor na máquina onde você instalou o **etherboot**, então mude seu diretório atual para o diretório `src` na árvore **etherboot** e digite:

```
# gmake bin32/devicetype.fd0
```

O *devicetype* depende do tipo de interface Ethernet na estação de trabalho sem disco. Veja o arquivo `NIC` no mesmo diretório para determinar o *devicetype* correto.

19.7.1.3. Configurando os Servidores TFTP e NFS

você precisa ativar o `tftpd` no servidor TFTP:

1. Crie um diretório de onde o `tftpd` vai fornecer arquivos, p.ex. `/tftpboot`.

2. Adicione esta linha ao seu `/etc/inetd.conf`:
3. `tftp dgram udp wait root /usr/libexec/tftpd
tftpd -s /tftpboot`

Nota: Parece que pelo menos algumas versões de PXE querem a versão TCP do TFTP. Neste caso, adicione a segunda linha, substituindo `dgram udp` por `stream tcp`.

4. Diga ao `inetd` para reler seu arquivo de configuração:
5. `# kill -HUP `cat /var/run/inetd.pid``

Você pode colocar o diretório `tftpboot` em qualquer lugar no servidor. Certifique-se de que a localização está configurada em ambos `inetd.conf` e `dhcpd.conf`.

Você também precisa ativar o NFS e exportar o sistema de arquivo adequado no servidor NFS.

1. Adicione isto ao `/etc/rc.conf`:
2. `nfs_server_enable="YES"`
3. Exporte o sistema de arquivo onde está localizado o diretório raiz da estação de trabalho sem disco, adicionando o seguinte ao `/etc/exports` (ajuste o ponto de montagem do volume e troque *margaux* pelo nome da estação de trabalho sem disco):
4. `/data/misc -alldirs -ro margaux`
5. Diga ao `mountd` para reler seu arquivo de configuração. Se você realmente precisasse ativar o NFS no `/etc/rc.conf` no primeiro passo, você provavelmente iria preferir reinicializar.
6. `# kill -HUP `cat /var/run/mountd.pid``

19.7.1.4. Construindo um Kernel Sem Disco

Crie um arquivo de configuração de kernel para o cliente sem disco com as seguintes opções (em acréscimo às opções usuais):

```
options      BOOTP          # Use BOOTP to obtain IP address/hostname
options      BOOTP_NFSROOT  # NFS mount root filesystem using
BOOTP info
options      BOOTP_COMPAT  # Workaround for broken bootp
daemons.
```

Você também pode querer usar `BOOTP_NFSV3` e `BOOTP_WIRED_TO` (veja o `LINT`).

Construa o kernel (Veja [Capítulo 9](#)), e copie-o para o diretório do `tftp`, sob o nome listado em `dhcpd.conf`.

19.7.1.5. Preparando o Sistema de Arquivo Raiz

você precisa criar um sistema de arquivo raiz para as estações de trabalho sem disco, no local listado como *root-path* no `dhcpd.conf`.

A forma mais fácil de fazê-lo é usar o shell script

`/usr/share/examples/diskless/clone_root`. Este script precisa ser personalizado, pelo menos para ajustar o lugar onde o sistema de arquivo será criado (a variável *DEST*).

Refira aos comentários no topo do script para instruções. Elas explicam como o sistema de arquivo base é construído, e como os arquivos podem ser seletivamente cancelados por versões específicas para operação sem disco, para uma subrede ou, para uma estação de trabalho individual. Eles também dão exemplos para os arquivos `/etc/fstab` e `/etc/rc.conf` para operação sem disco.

Os arquivos README em `/usr/share/examples/diskless` contêm muitas informações técnicas, mas, juntamente com outros exemplos no diretório `diskless`, eles na realidade documentam um método de configuração que é diferente do usado por `clone_root` e `/etc/rc.diskless[12]`, o que é um pouco confuso. Use-os apenas para referência, exceto se você preferir o método por eles descrito, sendo que, neste caso, você vai precisar de scripts `rc` modificados.

19.7.1.6. Configurando o Arquivo de Troca (Swap)

Se necessário, um arquivo de troca (swap) localizado no servidor pode ser acessado via NFS. As opções exatas do `bootptab` ou `dhcpd.conf` não são claramente documentadas neste momento. As seguintes sugestões de configuração foram reportadas como funcionais em algumas instalações usando o `isc-dhcp 3.0rc11`.

1. Adicione as seguintes linhas ao `dhcpd.conf`:
2. # Global section
3. option swap-path code 128 = string;
4. option swap-size code 129 = integer 32;
- 5.
6. host margaux {
7. ... # Standard lines, see above
8. option swap-path "192.168.4.4:/netswapvolume/netswap";
9. option swap-size 64000;
10. }
- 11.

A idéia é que, pelo menos para o cliente FreeBSD, a opção código 128 é o caminho para o arquivo de troca (swap) do NFS, e a opção código 129 é o tamanho do arquivo de troca, em kilobytes. Versões anteriores do `dhcpd` permitiam uma sintaxe `option option-128 "...`, que parece não mais funcionar.

O `/etc/bootptab`, ao invés, usaria a seguinte sintaxe:

```
T128="192.168.4.4:/netswapvolume/netswap":T129=64000
```

12. No servidor de arquivo de troca (swap), crie o(s) arquivo(s) de troca:
13. # mkdir /netswapvolume/netswap
14. # cd /netswapvolume/netswap
15. # dd if=/dev/zero bs=1024 count=64000
of=swap.192.168.4.6
16. # chmod 0600 swap.192.168.4.6
- 17.

192.168.4.6 é o endereço IP do cliente sem disco.

18. No servidor de arquivo de troca (swap) NFS, adicione a seguinte linha ao
/etc/exports:
19. /netswapvolume -maproot=0:10 -alldirs margaux
20.

Então diga ao **mountd** para reler o arquivo exports, como acima.

19.7.1.7. Casos Diversos

19.7.1.7.1. Rodando com um /usr Somente-Leitura

Se estação de trabalho sem disco está configurada para rodar X, você precisará ajustar o arquivo de configuração xdm, o qual coloca o registro (log) de erros em /usr, por padrão.

19.7.1.7.2. Usando um Servidor Não-FreeBSD

Quando o servidor do sistema de arquivo raiz não roda FreeBSD, você terá que criar o sistema de arquivo raiz em uma máquina FreeBSD e depois copiá-lo para seu destino, usando o tar ou cpio.

Nesta situação, às vezes existem alguns problemas com arquivos especiais em /dev, por causa de tamanhos inteiros maior/menor (major/minor) diferentes. A solução para este problema é exportar um diretório da máquina não-FreeBSD para criar as entradas de dispositivos corretas (FreeBSD 5.0 e posteriores usam [devfs\(5\)](#) para alocar nós de dispositivos - device nodes - transparentemente para o usuário, executar MAKEDEV nestas versões é inútil).

19.8. Rede Digital de Serviços Integrados - RDSI (ISDN)

Uma boa fonte de informações sobre a tecnologia e o hardware ISDN é a [página sobre ISDN de Dan Kegel](#).

Um simples e rápido mapa para o ISDN está a seguir:

- Se você vive na Europa, pode querer investigar a seção de placas ISDN.
- Se você estiver planejando usar ISDN primariamente para se conectar à Internet através de um Provedor de forma discada não permanente, você pode olhar os Adaptadores de Terminais. Assim você terá maior flexibilidade e menos problemas se você trocar de provedores.
- Se você estiver conectando duas Redes Locais ou à Internet com uma conexão ISDN dedicada, você pode considerar a opção de um roteador/ponte dedicado.

O custo é um fator significativo na determinação de qual solução você vai escolher. As seguintes opções são listadas da menos para a mais cara.

19.8.1. Placas ISDN

Contribuído por Hellmuth Michaelis.

A implementação ISDN do FreeBSD suporta somente o padrão DSS1/Q.931 (ou Euro-ISDN) usando placas passivas. A partir do FreeBSD 4.4, algumas placas ativas são suportadas nas quais o firmware também suporta outros protocolos de sinalização; isto também inclui a primeira placa ISDN de Acesso Primário (Primary Rate - PRI) suportada.

O **Isdn4bsd** lhe permite conectar a outros roteadores ISDN usando IP sobre HDLC ou PPP síncrono: tanto usando um kernel contendo PPP com `isppp` - um driver modificado a partir do `sppp` - ou usando o comando de usuário `ppp(8)`. Usando o comando `ppp(8)`, é possível agregar dois ou mais canais B ISDN. Uma aplicação de secretária eletrônica telefônica também está disponível, bem como muitos utilitários, como um modem de 300 Baud em software.

Um crescente número de placas ISDN para PC são suportadas pelo FreeBSD e os relatos mostram que ele é usado com sucesso por toda a Europa e em muitas outras partes do mundo.

As placas ISDN passivas suportadas são na maioria as que possuem os chipsets Infineon (anteriormente Siemens) ISAC/HSCX/IPAC, mas também placas ISDN com chips da Cologne Chip (somente barramento ISA), placas PCI com chips Winbond W6692, algumas placas com combinações do chipset Tiger300/320/ISAC e algumas placas com chipsets de fabricantes específicos como as placas AVM Fritz!Card PCI V.1.0 e a AVM Fritz!Card PnP.

Atualmente as placas ISDN ativas suportadas são as AVM B1 (ISA e PCI) e as AVM T1 PCI PRI.

Para documentação sobre **isdn4bsd**, veja o diretório `/usr/share/examples/isdn/` em seu sistema FreeBSD ou em [homepage of isdn4bsd](#) que também tem ponteiros para dicas, erratas e muito mais documentação como o [isdn4bsd handbook](#).

No caso de você estar interessado em adicionar suporte para um protocolo ISDN diferente, uma placa PC ISDN atualmente não suportada ou de alguma forma melhorar o **isdn4bsd**, por favor, entre em contato com Hellmuth Michaelis <hm@FreeBSD.org>.

Para questões referentes à instalação, configuração e resolução de problemas com **isdn4bsd**, está disponível uma lista de discussão [freebsd-isdn](#).

19.8.2. Adaptadores de Terminal ISDN

Adaptadores de Terminal (AT), são para o ISDN o que os modems são para as linhas telefônicas normais.

A maioria dos ATs usam o conjunto de comandos padrão dos modems Hayes, e podem ser usados como substitutos de modems.

Um AT vai operar basicamente da mesma forma que um modem exceto pelas velocidades de conexão e vazão serão muito mais rápidas que seu antigo modem. Você vai precisar configurar o [PPP](#) exatamente da mesma forma que configura um modem. Certifique-se de configurar a velocidade serial para a maior possível.

A principal vantagem de usar um AT para conectar a um Provedor Internet é que você pode usar PPP Dinâmico. Enquanto o espaço de endereçamento IP se torna cada vez mais escasso, a maioria dos provedores não desejam mais te fornecer um endereço IP estático. A maioria dos roteadores dedicados não são capazes de suportar alocação dinâmica de IP.

ATs contam completamente com o daemon PPP que você está executando para suas características e estabilidade de conexão. Isto permite a você facilmente fazer atualização de usar um modem para ISDN em uma máquina FreeBSD, se você já possui o PPP configurado. Entretanto, quaisquer problemas que você já experimentou com o programa PPP irão persistir.

Se você quer estabilidade máxima, use a opção de kernel [PPP](#), não o [ijjPPP](#) de contexto de usuário.

É sabido que os seguintes ATs funcionam com o FreeBSD.

- Motorola BitSurfer e Bitsurfer Pro
- Adtran

A maioria dos outros ATs provavelmente também vão funcionar, uma vez que os fornecedores de ATs tentam garantir que seu produto pode aceitar a maior parte do conjunto de comandos padrão AT dos modems.

O verdadeiro problema com os ATs externos é que, como os modems, você vai precisar de uma boa placa serial em seu computador.

Você deve ler o tutorial [FreeBSD Serial Hardware](#) para uma compreensão detalhada dos dispositivos seriais e as diferenças entre portas seriais assíncronas e síncronas.

Um AT executando em uma porta serial de um PC padrão (assíncrona) limita você a 115.2 Kbps, mesmo que você tenha uma conexão de 128 Kbps. Para utilizar completamente os 128 Kbps que o ISDN é capaz, você precisa mover o AT para uma placa serial síncrona.

Não faça a tolice de comprar um AT interno pensando que você evitou o assunto síncrono/assíncrono. Os ATs internos simplesmente possuem um chip de porta serial padrão PC embutidos. Tudo que isto fará é poupá-lo de comprar outro cabo serial e encontrar outra tomada elétrica disponível.

Uma placa síncrona com um AT é pelo menos tão rápida quanto um roteador dedicado, e com uma simples caixa 386 com FreeBSD gerenciando-a, provavelmente mais flexível.

A escolha de AT/síncrono ou roteador dedicado é um amplo assunto religioso. Houve algumas debates sobre isto nas listas de discussão. Eu sugiro que você busque os [arquivos](#) para a discussão completa.

19.8.3. Roteadores/Pontes Dedicados

Pontes ou roteadores ISDN não são nada específicos ao FreeBSD ou qualquer outro sistema operacional. Para uma descrição mais completa da tecnologia de roteamento e ligação, por favor consulte um livro de referência em Redes de Comunicação de Dados.

No contexto desta página, os termos roteador e ponte serão usados de forma intercambiável.

Enquanto o custo de roteadores/pontes ISDN de poucos recursos diminui, tornam-se cada vez mais escolhas populares. Um roteador ISDN é uma pequena caixa que se conecta diretamente em sua rede local Ethernet e gerencia sua própria conexão a outro roteador/ponte. Possui embutido software para comunicação via PPP e outros protocolos populares.

Um roteador vai lhe permitir vazão muito mais rápida que um AT padrão, uma vez que estará usando uma conexão ISDN totalmente síncrona.

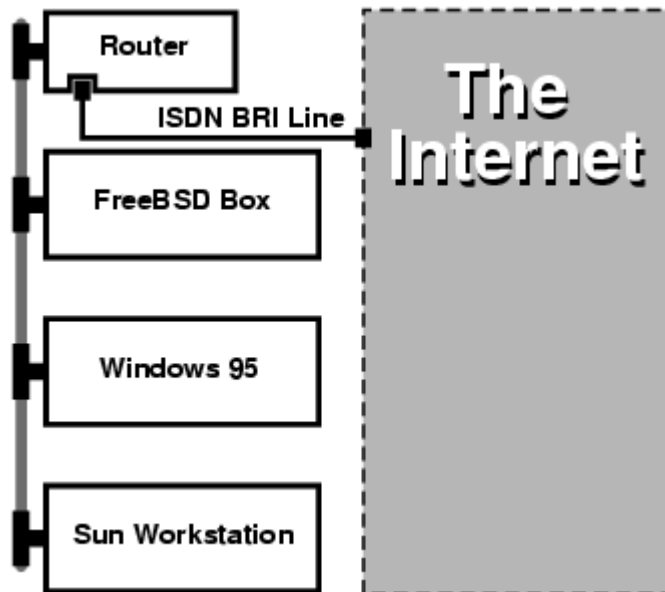
O principal problema com roteadores e pontes ISDN é que a interoperabilidade entre fabricantes ainda pode ser um problema. Se você estiver planejando se conectar a um provedor Internet, você deve conversar suas necessidades com eles.

Se você está planejando conectar dois segmentos de Redes Locais, como sua Rede Local doméstica e a do escritório, esta é a solução mais simples e de mais baixo custo de manutenção. Uma vez que você está adquirindo o equipamento para as duas pontas da conexão, você pode estar seguro de que o enlace irá funcionar.

Por exemplo, para conectar um computador doméstico ou rede de escritório filial à rede do escritório principal, a seguinte configuração pode ser usada.

Exemplo 19-2. Escritório Filial ou Rede Doméstica

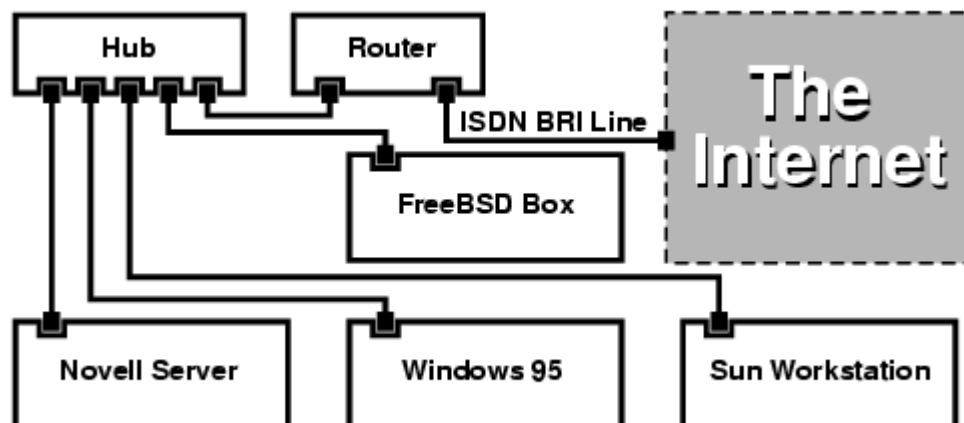
A rede usa uma topologia em barra com Ethernet 10 base 2 ("thinnet"). Conecte o roteador ao cabo de rede com transceptor AUI/10BT, se necessário.



Se seu escritório doméstico/filial é somente um computador, você pode usar um cabo de par trançado crossover para conectá-lo diretamente ao roteador dedicado.

Exemplo 19-3. Escritório Principal ou Outra Rede Local

A rede usa uma topologia em estrela, com Ethernet 10 base T ("Par Trançado").



Uma grande vantagem da maioria dos roteadores/pontes é que eles lhe permitem ter 2 conexões PPP *separadas e independentes* a 2 sítios separados ao *mesmo* tempo. Isto não é suportado na maioria dos ATs, exceto em modelos específicos (geralmente caros) que possuem duas portas seriais. Não confundir isto com agregação de canais, MPP etc.

Isto pode ser uma característica muito útil se, por exemplo, você possuir uma conexão ISDN dedicada em seu escritório e desejar se conectar através dela, mas não quer ter mais uma linha ISDN no trabalho. Um roteador no local do escritório pode gerenciar uma conexão dedicada de canal B (64 Kbps) para a Internet e usar o outro canal B para uma conexão de dados separada. O segundo canal B pode ser usado para receber ou

realizar acesso discado ou agregação dinâmica (MPP etc.) com o primeiro canal B para obter mais largura de banda.

Uma ponte Ethernet também vai lhe permitir transmitir mais do que tráfego IP. Você também pode transmitir IPX/SPX ou quaisquer outros protocolos que você usa.

19.9. NIS/YP

Escrito por Bill Swingle. Aumentado por Eric Ogren e Udo Erdelhoff.

19.9.1. O Que É?

O NIS, que quer dizer Serviços de Informações de Rede (Network Information Services), foi desenvolvido pela Sun Microsystems para centralizar a administração de sistemas UNIX® (originalmente SunOS™). Ele tornou-se, essencialmente, um padrão da indústria; todos os maiores sistemas tipo UNIX (Solaris™, HP-UX, AIX®, Linux, NetBSD, OpenBSD, FreeBSD etc) suportam NIS.

O NIS era anteriormente conhecido por Páginas Amarelas (Yellow Pages), mas por causa de questões de marcas registradas, a Sun trocou o nome. O antigo termo (e yp) ainda é muitas vezes visto e usado.

É um sistema cliente/servidor baseado em RPC que permite a um grupo de máquinas dentro de um domínio NIS, compartilhar um conjunto comum de arquivos de configuração. Isto permite que um administrador de sistemas configure sistemas clientes NIS com um mínimo de dados de configurações a partir de um ponto central.

É similar ai sistema de domínio do Windows NT®; apesar da implementação interna de ambos não ser nada parecida, a funcionalidade básica pode ser comparada.

19.9.2. Termos/Processos Que Você Deve Saber

Existem diversos termos e vários processos de usuário importantes que você vai encontrar quando tentar implementar NIS no FreeBSD, esteja você tentando criar um servidor NIS ou atuando como um cliente NIS:

Termo	Descrição
Nome de Domínio NIS	Um servidor mestre NIS e todos os seus clientes (incluindo seus servidores escravos) possuem um nome de domínio NIS. Similar a um nome de domínio Windows NT, o nome de domínio NIS não tem nada a ver com DNS.
portmap	Precisa estar rondando para ativar o RPC (Remote Procedure Call, um protocolo de rede usado pelo NIS). Se o <code>portmap</code> não estiver rodando, será

Termo	Descrição
	impossível rodar um servidor NIS, ou atuar como um cliente NIS.
ypbind	``Associa'' um cliente NIS a seu servidor. Vai receber o nome de domínio NIS do sistema e, usando RPC, conectará ao servidor. O <code>ypbind</code> é o centro da comunicação cliente-servidor em um ambiente NIS; se o <code>ypbind</code> morrer em uma máquina cliente, ela não será capaz de acessar o servidor NIS.
ypserv	Deve ser rodado somente em servidores NIS; este é o próprio processo servidor NIS. Se o ypserv(8) morrer, então o servidor não será mais capaz de responder a solicitações NIS (felizmente, existe um servidor escravo que pode assumir em seu lugar). Existem algumas implementações de NIS (mas não a do FreeBSD), que não tentam reconectar a outro servidor se o servidor anteriormente usado morre. Frequentemente, a única coisa que ajuda neste caso é reiniciar o processo servidor (ou mesmo o servidor inteiro) ou o processo <code>ypbind</code> no cliente.
rpc.yppasswdd	Outro processo que deve rodar somente nos servidores mestre NIS; este é um daemon que permitirá aos clientes NIS mudarem suas senhas NIS. Se este daemon não estiver rodando, os usuários terão que acessar (fazer login) no servidor mestre NIS e mudar suas senhas nele.

19.9.3. Como Funciona?

Existem três tipos de sistemas em um ambiente NIS: servidores mestre, servidores escravos e clientes. Servidores atuam como repositórios centrais para informações de configurações de sistemas. Servidores mestres guardam a cópia autoritativa desta informação, enquanto servidores escravos espelham estas informações para redundância. Clientes contam com os servidores para lhes fornecerem estas informações.

Informações em muitos arquivos podem ser compartilhados desta forma. Os arquivos `master.passwd`, `group` e `hosts` são normalmente compartilhados via NIS. Quando um processo em um cliente precisa de informação que normalmente seria encontrada nestes arquivos localmente, em vez disso, ele faz uma pergunta ao servidor NIS ao qual está conectado.

19.9.3.1. Tipos de Máquinas

- Um *servidor NIS mestre*. Este servidor, de forma análoga a um controlador primário de domínio Windows NT, mantém os arquivos usados por todos os clientes NIS. Os arquivos `passwd`, `group`, e diversos outros usados pelos clientes NIS residem no servidor NIS mestre.

Nota: É possível para uma máquina ser um servidor NIS mestre para mais de um domínio NIS. Entretanto, isto não será abordado nesta introdução, que assume um ambiente NIS de escala relativamente pequena.

- *Servidores NIS escravos.* Similar aos controladores de domínio secundários do Windows NT, os servidores NIS escravos mantêm cópias dos arquivos de dados do mestre. Os servidores NIS escravos fornecem redundância, que é necessária em ambientes importantes. Eles também ajudam a balancear a carga no servidor mestre: os clientes NIS sempre se conectam ao servidor NIS do qual primeiro receberem uma resposta, e isto inclui as respostas do servidor escravo.
- *Clientes NIS.* Clientes NIS, como a maioria das estações de trabalho Windows NT se autenticam contra o servidor NIS (ou o controlador de domínio Windows NT no caso de estações de trabalho Windows NT) para obterem acesso (log on).

19.9.4. Usando NIS/YP

Esta seção vai tratar de configurar um ambiente NIS de exemplo.

Nota: Esta seção assume que você está rodando o FreeBSD 3.3 ou mais recente. As instruções aqui mostradas *provavelmente* funcionarão para qualquer versão do FreeBSD maior que 3.0, mas não há garantias de que isto seja verdadeiro.

19.9.4.1. Planejamento

Vamos assumir que você é o administrador de um pequeno laboratório universitário. Neste laboratório, que consiste de 15 máquinas FreeBSD, atualmente não há ponto centralizado de administração; cada máquina têm seu próprio `/etc/passwd` e `/etc/master.passwd`. Estes arquivos são mantidos em sincronia entre si através de intervenções manuais; atualmente, quando você adiciona um usuário no lab, você precisa rodar `adduser` em todas as 15 máquinas. Claramente, isto precisa mudar, então você decidiu converter o lab para usar NIS, usando duas das máquinas como servidores.

Então, a configuração do lab agora parece com algo como:

Nome da máquina	Endereço IP	Papel da máquina
ellington	10.0.0.2	mestre NIS
coltrane	10.0.0.3	escravo NIS
basie	10.0.0.4	Estação de trabalho da faculdade
bird	10.0.0.5	Máquina cliente
cli[1-11]	10.0.0.[6-17]	Outras máquinas clientes

Se você está configurando um esquema NIS pela primeira vez, é uma boa idéia pensar muito bem como você quer fazer. Não importa qual o tamanho da sua rede, existem algumas poucas decisões que precisam ser tomadas.

19.9.4.1.1. Escolhendo um Nome de Domínio NIS

Isto pode não ser o "nome de domínio" com o qual você está acostumado. Ele é mais corretamente chamado de "nome de domínio NIS". Quando um cliente anuncia suas solicitações por informações via broadcasts, ele inclui o nome do domínio NIS ao qual pertence. Assim é como múltiplos servidores em uma rede podem saber qual deve responder qual solicitação. Pense no nome de domínio NIS como o nome de um grupo de sistemas relacionados de alguma forma.

Algumas organizações escolhem usar seu nome de domínio Internet para ser seu nome de domínio NIS. Isto não é recomendado porque pode confundir ao se tentar resolver problemas de rede. O nome de domínio NIS deve ser único dentro de sua rede e é útil que descreva o grupo de máquinas que representa. Por exemplo, o departamento de Arte da Acme Ltda. pode estar no domínio NIS "acme-arte". Para este exemplo, assuma que você escolheu o nome *domínio-teste*.

Entretanto, alguns sistemas operacionais (notadamente o SunOS) usam seu nome de domínio NIS como seu nome de domínio Internet. Se uma ou mais máquinas em sua rede possuem esta restrição, você *precisa* usar o nome de domínio Internet como seu nome de domínio NIS.

19.9.4.1.2. Requerimentos Físicos do Servidor

Existem diversas coisas a serem mantidas em mente quando se for escolher uma máquina para usar como servidor NIS. Uma das infelicidades do NIS é o nível de dependência que os clientes têm do servidor. Se um cliente não puder fazer contato com o servidor de seu domínio NIS, muito frequentemente a máquina se torna inutilizada. A falta de informações de usuários e grupos causa congelamento na maioria dos sistemas. Com isto em mente, você deve se certificar de escolher uma máquina que não estará sujeita a ser reinicializada regularmente, ou uma que pode ser usada para desenvolvimento. O servidor NIS idealmente deve ser uma máquina isolada, cujo único propósito na vida é ser um servidor NIS. Se você tem uma rede que não é pesadamente utilizada, é aceitável colocar o servidor NIS em uma máquina rodando outros serviços, apenas tenha em mente que se o servidor NIS se tornar indisponível, afetará adversamente *todos* os clientes NIS.

19.9.4.2. Servidores NIS

As cópias canônicas de todas as informações NIS são armazenadas em uma única máquina chamada servidor NIS mestre. As bases de dados usadas para armazenar as informações são chamadas de mapas NIS. No FreeBSD, estes mapas são armazenados em `/var/yp/[domínio]` onde `[domínio]` é o nome de domínio NIS sendo servido. Um único servidor NIS pode suportar diversos domínios de uma vez, então é possível ter diversos domínios destes diretórios, um para cada domínio suportado. Cada domínio terá seu próprio conjunto independente de mapas.

Os servidores mestre e escravo NIS tratam de todas as solicitações NIS com o daemon `yppserv`. O `yppserv` é responsável por receber solicitações entrantes dos clientes NIS, traduzindo o domínio solicitado e mapeando o nome a um caminho para o arquivo de banco de dados correspondente e transmitindo dados da base de volta para o cliente.

19.9.4.2.1. Configurando Um Servidor NIS Mestre

Configurar um servidor NIS mestre pode ser relativamente simples, dependendo de suas necessidades. O FreeBSD possui suporte nativo para NIS. Tudo que você precisa é adicionar as seguintes linhas ao `/etc/rc.conf`, e o FreeBSD irá fazer o resto para você.

1. `nisdomainname="dominio-teste"`
2. Esta linha vai configurar o nome de domínio NIS para *dominio-teste* na configuração de rede (p.ex. depois de uma inicialização).
3. `nis_server_enable="YES"`
4. Isto dirá ao FreeBSD para iniciar os processos servidores NIS quando a rede for a próxima a ser ativada.
5. `nis_yppasswdd_enable="YES"`
6. Isto vai ativar o daemon `rpc.yppasswdd` que, como mencionado acima, permitirá aos usuários mudarem suas senhas NIS a partir de uma máquina cliente.

Nota: Dependendo de sua configuração NIS, você pode precisar incluir entradas adicionais. Veja a [seção sobre servidores NIS que também são clientes NIS](#), abaixo, para detalhes.

Agora, tudo que você precisa fazer é rodar o comando `/etc/netstart` como superusuário. Ele irá configurar tudo para você, usando os valores que você definiu no `/etc/rc.conf`.

19.9.4.2.2. Inicializando os Mapas NIS

Os *mapas NIS* são arquivos de bancos de dados, mantidos no diretório `/var/yp`. Eles são gerados de arquivos de configuração no diretório `/etc` do servidor NIS mestre, com uma exceção: o arquivo `/etc/master.passwd`. Isto é por uma boa razão; você não quer propagar senhas de sua conta `root` e de outras contas administrativas para todos os servidores no domínio NIS. Então, antes de inicializar os mapas NIS, você deve:

```
# cp /etc/master.passwd /var/yp/master.passwd
# cd /var/yp
# vi master.passwd
```

Você deve remover todas as entradas referentes a contas de sistema (`bin`, `tty`, `kmem`, `games` etc), bem como quaisquer contas que você não quer que sejam propagadas aos clientes NIS (por exemplo, `root` e qualquer outra conta com UID 0 (superusuário)).

Nota: Tenha certeza de que o arquivo `/var/yp/master.passwd` não pode ser acessado por grupo (`group`) ou pelo mundo (`world`) (modo 600)! Use o comando `chmod`, se necessário.

Quando você tiver terminado, é hora de inicializar os mapas NIS! O FreeBSD inclui um script chamado `ypinit` para fazer isto por você (veja sua página de manual para mais informações). Note que este script está disponível na maioria dos Sistemas Operacionais UNIX, mas não em todos. No UNIX Digital UNIX/Compaq Tru64 é chamado `ypsetup`. Porque estamos gerando mapas para um mestre NIS, estamos passando a opção `-m` ao `ypinit`. Para gerar estes mapas NIS, assumindo que você já executou os passos acima, rode:

```
ellington# ypinit -m dominio-teste
Server Type: MASTER Domain: dominio-teste
Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
Do you want this procedure to quit on non-fatal errors? [y/n: n] n
Ok, please remember to go back and redo manually whatever fails.
If you don't, something might not work.
At this point, we have to construct a list of this domains YP servers.
rod.darktech.org is already known as master server.
Please continue to add any slave servers, one per line.  When you are
done with the list, type a <control D>.
master server   : ellington
next host to add: coltrane
next host to add: ^D
The current list of NIS servers looks like this:
ellington
coltrane
Is this correct? [y/n: y] y
```

[..output from map generation..]

```
NIS Map update completed.
ellington has been setup as an YP master server without any errors.
```

O `ypinit` deve ter criado `/var/yp/Makefile` do `/var/yp/Makefile.dist`. Quando criado, este arquivo assume que você está operando em um ambiente de servidor único, somente com máquinas FreeBSD. Uma vez que o *dominio-teste* também tem um servidor escravo, você precisa editar o `/var/yp/Makefile`:

```
ellington# vi /var/yp/Makefile
```

Você deve comentar a linha que diz

```
NOPUSH = "True"
```

(se ainda não estiver comentada).

19.9.4.2.3. Configurando um Servidor NIS Escravo

configurando um servidor NIS escravo é ainda mais simples que configurar o mestre. Acesse o servidor escravo e edite o arquivo `/etc/rc.conf` como fez anteriormente. A única diferença é que agora precisamos usar a opção `-s` quando rodarmos o `ypinit`. A opção `-s` requer que o nome do mestre NIS também lhe seja passada, de forma que nossa linha de comando fique parecida com esta:

```
coltrane# ypinit -s ellington dominio-teste
```

Server Type: SLAVE Domain: dominio-teste Master: ellington

Creating an YP server will require that you answer a few questions. Questions will all be asked at the beginning of the procedure.

Do you want this procedure to quit on non-fatal errors? [y/n: n] n

Ok, please remember to go back and redo manually whatever fails. If you don't, something might not work. There will be no further questions. The remainder of the procedure should take a few minutes, to copy the databases from ellington.

Transferring netgroup...

ypxfr: Exiting: Map successfully transferred

Transferring netgroup.byuser...

ypxfr: Exiting: Map successfully transferred

Transferring netgroup.byhost...

ypxfr: Exiting: Map successfully transferred

Transferring master.passwd.byuid...

ypxfr: Exiting: Map successfully transferred

Transferring passwd.byuid...

ypxfr: Exiting: Map successfully transferred

Transferring passwd.byname...

ypxfr: Exiting: Map successfully transferred

Transferring group.bygid...

ypxfr: Exiting: Map successfully transferred

Transferring group.byname...

ypxfr: Exiting: Map successfully transferred

Transferring services.byname...

ypxfr: Exiting: Map successfully transferred

Transferring rpc.bynumber...

ypxfr: Exiting: Map successfully transferred

Transferring rpc.byname...

ypxfr: Exiting: Map successfully transferred

Transferring protocols.byname...

ypxfr: Exiting: Map successfully transferred

Transferring master.passwd.byname...

ypxfr: Exiting: Map successfully transferred

Transferring networks.byname...

ypxfr: Exiting: Map successfully transferred

Transferring networks.byaddr...

ypxfr: Exiting: Map successfully transferred

Transferring netid.byname...

ypxfr: Exiting: Map successfully transferred

Transferring hosts.byaddr...

ypxfr: Exiting: Map successfully transferred

Transferring protocols.bynumber...

ypxfr: Exiting: Map successfully transferred

Transferring ypservers...

ypxfr: Exiting: Map successfully transferred

Transferring hosts.byname...

ypxfr: Exiting: Map successfully transferred

coltrane has been setup as an YP slave server without any errors. Don't forget to update map ypservers on ellington.

Agora você deve ter um diretório chamado `/var/yp/dominio-teste`. Cópias dos mapas do servidor mestre NIS devem estar neste diretório. Você vai precisar estar certo de que estes sejam mantidos atualizados. As seguintes entradas no `/etc/crontab` dos servidores escravos devem resolver:

```
20      *      *      *      *      root    /usr/libexec/ypxfr
passwd.byname
21      *      *      *      *      root    /usr/libexec/ypxfr
passwd.byuid
```

Estas duas linhas forçam o escravo a sincronizar seus mapas com os mapas do servidor mestre. Apesar destas entradas não serem obrigatórias, uma vez que o servidor mestre tenta garantir que quaisquer mudanças aos seus mapas NIS sejam comunicadas aos seus escravos e porque a informação de senhas é vital para os sistemas dependentes do servidor, é uma boa idéia forçar as atualizações. Isto é mais importantes em redes com alta utilização, onde atualizações de mapas podem nem sempre ser completadas.

Agora, rode o comando `/etc/netstart` também no servidor escravo, o qual reinicia o servidor NIS.

19.9.4.3. clientes NIS

Um cliente NIS estabelece o que é chamado de uma associação (ou ligação) a um servidor NIS específico, usando o daemon `ypbind`. O `ypbind` verifica o domínio padrão do sistema (como configurado pelo comando `domainname`), e começa a fazer difusão (envio de broadcasts) de solicitações RPC na rede local. Estas solicitações especificam o nome do domínio para o qual o `ypbind` está tentando estabelecer uma associação. Se um servidor que foi configurado para atender às solicitações receber um dos broadcasts, ele responderá ao `ypbind`, que vai gravar o endereço do servidor. Se existirem diversos servidores disponíveis (um mestre e diversos escravos, por exemplo), o `ypbind` vai usar o endereço do primeiro que lhe responder. Deste ponto em diante, o sistema cliente vai direcionar todas as suas solicitações NIS para aquele servidor. O `ypbind` irá ocasionalmente fazer ``ping" no servidor para certificar-se de que ele está ativo e operando. Se houver falha na recepção de uma resposta a um dos pings dentro de um período razoável de tempo, o `ypbind` marcará o domínio como desassociado (unbound) e vai começar novamente a fazer difusão na esperança de localizar outro servidor.

19.9.4.3.1. Configurando Um Cliente NIS

Configurando uma máquina FreeBSD para ser um cliente NIS é bastante simples.

1. Edite o arquivo `/etc/rc.conf` e adicione as seguintes linhas para configurar o nome de domínio NIS e inicie o `ypbind` durante a inicialização da rede:
2. `nisdomainname="dominio-teste"`
3. `nis_client_enable="YES"`
4. Para importar todas as possíveis entradas de senhas do servidor NIS, remova todas as contas de usuários de seu arquivo `/etc/master.passwd` e use o `vipw` para adicionar a seguinte linha ao final do arquivo:
5. `+:::~::~:`

Nota: Esta linha irá proporcionar uma conta a qualquer um com uma conta válida nos mapas do servidor NIS. Existem muitas formas de configurar seu cliente NIS alterando esta linha. Veja a [seção netgroups](#) abaixo para mais informações. Para leitura mais detalhada veja o livro da O'Reilly sobre *Gerenciando NFS e NIS (Managing NFS and NIS)*.

Nota: Você deve manter pelo menos uma conta local (p.ex. não importada via NIS) em seu `/etc/master.passwd` e esta conta também deve ser membro do grupo `wheel`. Se acontecer algo errado com o NIS, esta conta pode ser usada para acessar remotamente, tornar-se `root`, e fazer manutenções.

6. Para importar todos as possíveis entradas de grupos do servidor NIS, adicione esta linha ao seu arquivo `/etc/group`:
7. `+:*:::`

Após completar estes passos, você deve ser capaz de rodar `ypcat passwd` e ver o mapa de senhas do servidor NIS.

19.9.5. Segurança do NIS

Em geral, qualquer usuário remoto pode emitir uma RPC ao [ypserv\(8\)](#) e recuperar os conteúdos dos mapas NIS, desde que os usuários remotos saibam o nome do domínio. Para evitar tais transações não-autorizadas, o [ypserv\(8\)](#) suporta uma característica chamada `securenets` que pode ser usada para restringir acesso a um conjunto de sistemas. Na inicialização, o [ypserv\(8\)](#) irá tentar carregar as informações de `securenets` de um arquivo chamado `/var/yp/securenets`.

Nota: Este caminho varia dependendo do caminho especificado com a opção `-p`. Este arquivo contém entradas que consistem na especificação da rede e de uma máscara de rede separadas por um espaço em branco. Linhas iniciando com ``#` são consideradas comentários. Um arquivo `securenets` de amostra se parece com este:

```
# permite conexoes do sistema local -- obrigatorio
127.0.0.1      255.255.255.255
# permite conexões de qualquer sistema
# na rede 192.168.128.0
192.168.128.0 255.255.255.0
# permite conexões de qualquer sistema
# entre 10.0.0.0 e 10.0.15.255
# isto inclui as máquinas no laboratório de testes
10.0.0.0      255.255.240.0
```

Se o [ypserv\(8\)](#) receber uma solicitação de um endereço que se enquadra (`match`) a uma destas regras, ele irá processar a solicitação normalmente. Se o endereço falhar em se enquadrar a uma das regras, a solicitação será ignorada e uma mensagem de alerta será registrada. Se o arquivo `/var/yp/securenets` não existir, o `ypserv` permitirá conexões vindas de qualquer sistema.

O programa `ypserv` também tem suporte ao pacote **tcpwrapper** de Wietse Venema. Isto permite ao administrador utilizar os arquivos de configuração do **tcpwrapper** para controle de acesso, ao invés do `/var/yp/securenets`.

Nota: Enquanto ambos os mecanismos de controle fornecem alguma segurança, eles, como o teste de porta privilegiada, estão vulneráveis a ataques de "`falsificação de IP (IP spoofing)". Todo o tráfego relativo a NIS deve ser bloqueado em seu firewall.

Servidores usando `/var/yp/securenets` podem falhar em atender clientes NIS legítimos com implementações arcaicas de TCP/IP. Algumas destas implementações configuram todos os bits de sistemas (hosts) para zero quando fazem difusão (broadcasts) e/ou falham em observar a máscara de subrede quando calculam o endereço de broadcast. Enquanto alguns destes problemas podem ser consertados mudando a configuração do cliente, outros problemas podem forçar a aposentadoria dos sistemas clientes em questão ou o abandono do `/var/yp/securenets`.

Usando `/var/yp/securenets` em um servidor com tal implementação arcaica de TCP/IP é realmente uma má idéia e levará à perda de funcionalidade NIS para grandes partes de sua rede.

O uso do pacote **tcpwrapper** aumenta a latência no seu servidor NIS. O retardo (delay) adicional pode ser longo o suficiente para causar expiração em programas clientes, especialmente em rede congestionadas ou com servidores NIS lentos. Se um ou mais de seus sistemas clientes sofrerem destes sintomas, você deve converter os sistemas em questão em servidores NIS escravos e forçá-los a associarem-se a si próprios.

19.9.6. Impedindo Alguns Usuários de Fazer Log On

Em nosso lab, há uma máquina `basie` que é suposta para ser uma estação de trabalho somente de professores. Não queremos retirar esta máquina do domínio NIS, ainda assim, o arquivo `passwd` no servidor NIS mestre contem contas para ambos professores e alunos. O que podemos fazer?

Há uma forma de impedir usuários específicos de acessarem uma máquina, mesmo que estejam presentes na base de dados NIS. Para isto, tudo o que você precisa fazer é adicionar `-nome_usuario` ao final do arquivo `/etc/master.passwd` na máquina cliente, onde `nome_usuario` é o nome de usuário do usuário que você deseja impedir de acessar o sistema. Isto deve ser preferivelmente feito usando o `vipw`, uma vez que o `vipw` irá verificar suas alterações no `/etc/master.passwd`, bem como automaticamente reconstruir a base de dados de senhas quando você terminar a edição. Por exemplo, se nós quisermos impedir o usuário `bill` de acessar `basie` faremos:

```
basie# vipw
[adicionar -bill ao final, e sair]
vipw: rebuilding the database...
vipw: done
```

```
basie# cat /etc/master.passwd
```

```
root:[password]:0:0::0:0:The super-user:/root:/bin/csh
toor:[password]:0:0::0:0:The other super-user:/root:/bin/sh
daemon:*:1:1::0:0:Owner of many system processes:/root:/sbin/nologin
operator:*:2:5::0:0:System &:/:/sbin/nologin
bin:*:3:7::0:0:Binaries Commands and Source,,:/:/sbin/nologin
tty:*:4:65533::0:0:Tty Sandbox:/:/sbin/nologin
kmem:*:5:65533::0:0:KMem Sandbox:/:/sbin/nologin
games:*:7:13::0:0:Games pseudo-user:/usr/games:/sbin/nologin
news:*:8:8::0:0:News Subsystem:/:/sbin/nologin
man:*:9:9::0:0:Mister Man Pages:/usr/share/man:/sbin/nologin
bind:*:53:53::0:0:Bind Sandbox:/:/sbin/nologin
```

```

uucp:*:66:66::0:0:UUCP pseudo-
user:/var/spool/uucppublic:/usr/libexec/uucp/uucico
xten:*:67:67::0:0:X-10 daemon:/usr/local/xten:/sbin/nologin
pop:*:68:6::0:0:Post Office Owner:/nonexistent:/sbin/nologin
nobody:*:65534:65534::0:0:Unprivileged user:/nonexistent:/sbin/nologin
+:::
-bill

```

basie#

19.9.7. Usando Netgroups

Contribuído por Udo Erdelhoff.

O método mostrado na seção anterior funciona razoavelmente bem se você precisa de regras especiais para uma pequena quantidade de usuários e/ou máquinas. Em redes maiores, você *vai* esquecer de bloquear alguns usuários de acessarem máquinas sensíveis, ou pode até ter que modificar cada máquina separadamente, perdendo assim, o maior benefício do NIS, a administração *centralizada*.

A solução dos desenvolvedores do NIS para este problema é chamada de *netgroups*. Seu propósito e semântica podem ser comparados aos grupos normais usados pelos sistemas de arquivos UNIX. As principais diferenças são uma falta de id numérico e a capacidade de definir um netgroup incluindo tanto contas de usuários e outros netgroups.

Os netgroups foram desenvolvidos para lidar com redes grandes e complexas, com centenas de usuários e máquinas. Por um lado, isto é uma Coisa Boa, se você é forçado a lidar com tal situação. Por outro lado, esta complexidade torna quase impossível explicar netgroups com exemplos realmente simples. O exemplo usado no restante desta seção demonstra este problema.

Vamos assumir que sua introdução bem sucedida ao NIS em laboratório chamou a atenção de seus superiores. Seu próximo trabalho será estender seu domínio NIS para cobrir algumas outras máquinas no campus. As duas tabelas contêm os nomes dos novos usuários, bem como novas máquinas e breve descrições delas.

Nome(s) de Usuário(s)	Descrição
alpha, beta	Empregados normais do departamento de TI
charlie, delta	Os novos aprendizes do departamento de TI
echo, foxtrott, golf, ...	Funcionários comuns
able, baker, ...	Os atuais estagiários
Nomes das Máquina(s)	Descrição
war, death, famine, pollution	Seus servidores mais importantes. Somente os funcionários de TI têm permissão para acessar estas máquinas.

Nome(s) de Usuário(s)	Descrição
pride, greed, envy, wrath, lust, sloth	Servidores menos importantes. Todos os membros do departamento de TI têm permissão de acesso nestes servidores.
one, two, three, four, ...	Estações de trabalho ordinárias. Somente os funcionários <i>reais</i> têm permissão para usar estas máquinas.
trashcan	Uma máquina muito velha, sem qualquer dado crítico. Mesmo o residente pode acessar esta máquina.

Se você tentasse implementar estas restrições separadamente bloqueando cada usuário, você teria que adicionar uma linha *-usuário* no `passwd` de cada sistema, para cada usuário que não está permitido acessar àquele sistema. Se você esquecer apenas uma entrada, você pode estar em apuros. Pode ser viável fazer isto corretamente durante a configuração inicial, entretanto, você eventualmente *vai* esquecer de adicionar linhas para novos usuários durante as operações diárias. Apesar de tudo, Murphy era um otimista.

Lidar com esta situação com netgroups oferece diversas vantagens. Cada usuário não precisa ser tratado separadamente; você designa um usuário para um ou mais netgroups e permite ou proíbe logins para todos os membros do netgroup. Se você adicionar uma nova máquina, você terá somente que definir as restrições de acesso para os netgroups. Se um novo usuário for adicionado, você terá somente que adicionar o usuário a um ou mais netgroups. Estas mudanças são independentes umas das outras; nada mais de "para cada combinação de usuário e máquina, faça..." se a sua configuração de NIS é cuidadosamente planejada, você somente terá que modificar exatamente um arquivo de configuração central para conceder ou negar acesso às máquinas.

O primeiro passo é a inicialização do mapa NIS do netgroup. O [ypinit\(8\)](#) do FreeBSD não cria este mapa por padrão, mas sua implementação de NIS vai suportá-lo quando for criado. Para criar um mapa vazio, digite

```
ellington# vi /var/yp/netgroup
```

e comece a adicionar conteúdo. Para nosso exemplo, precisamos de pelo menos quatro netgroups: Funcionários de TI, aprendizes de TI, funcionários normais e estagiários.

```
IT_EMP ( ,alpha,dominio-teste) ( ,beta,dominio)
IT_APP ( ,charlie,dominio-teste) ( ,delta,dominio-teste)
USERS ( ,echo,dominio-teste) ( ,foxtrott,dominio-teste) \
( ,golf,dominio-teste)
INTERNS ( ,able,dominio-teste) ( ,baker,dominio-teste)
```

IT_EMP, *IT_APP* etc. são nomes de netgroups. Cada grupo entre colchetes adiciona um ou mais contas de usuário a ele. Os três campos dentro de um grupo são:

1. O nome do(s) sistema(s) onde os seguintes itens são válidos. Se você não especificar um nome de sistema, a entrada é válida em todos os sistemas (hosts).

Se você especificar um sistema, você vai entrar em um mundo de escuridão, horror e total confusão.

2. O nome da conta que pertence a este netgroup.
3. O domínio NIS para a conta. Você pode importar contas de outros domínios NIS para seu netgroup se você é um dos camaradas sem sorte, com mais de um domínio NIS.

Cada um destes campos pode conter curingas. Veja [netgroup\(5\)](#) para detalhes.

Nota: Nomes de netgroup mais longos que 8 caracteres não devem ser usados, especialmente se você tem máquinas rodando outros sistemas operacionais em seu domínio NIS. Os nomes são sensíveis à diferenças entre maiúsculas e minúsculas; usar maiúsculas para nomes de netgroups é uma forma fácil de distinguir entre usuário, máquina e nomes de netgroups.

Alguns clientes NIS (diferentes do FreeBSD) não conseguem lidar com netgroups com grande número de entradas. Por exemplo, algumas antigas versões do SunOS começam a causar problemas se um netgroup contém mais de 15 *entradas*. Você pode contornar este limite criando vários sub-netgroups com 15 ou menos usuários e um netgroup real que consiste dos sub-netgroups:

```
BIGGRP1 (,joe1,dominio) (,joe2,dominio) (,joe3,dominio) [...]  
BIGGRP2 (,joe16,dominio) (,joe17,dominio) [...]  
BIGGRP3 (,joe31,dominio) (,joe32,dominio)  
BIGGROUP BIGGRP1 BIGGRP2 BIGGRP3
```

Você pode repetir este processo se precisar de mais que 225 usuários dentro de um único netgroup.

Ativar e distribuir seu novo mapa NIS é fácil:

```
ellington# cd /var/yp  
ellington# make
```

Isto vai gerar os três mapas NIS `netgroup`, `netgroup.byhost` e `netgroup.byuser`. Use [ypcat\(1\)](#) para verificar se seus novos mapas NIS estão disponíveis:

```
ellington% ypcat -k netgroup  
ellington% ypcat -k netgroup.byhost  
ellington% ypcat -k netgroup.byuser
```

A saída do primeiro comando deve parecer-se com o conteúdo do `/var/yp/netgroup`. O segundo comando não produzirá saída, se você não especificou netgroups específicos para sistemas (hosts). O terceiro comando pode ser usado para obter a lista de netgroups para um usuário.

A configuração do cliente é bem simples. Para configurar o servidor `war`, você somente terá que iniciar o [yipw\(8\)](#) e substituir a linha

```
+:::~:::
```

por

```
+@IT_EMP:::::::::
```

Agora, somente o dado para usuários definidos no netgroup *IT_EMP* é importado na base de dados de *war* e somente a estes usuários é permitido o acesso (login).

Infelizmente, esta limitação também se aplica à função `~` do shell e todas as rotinas convertendo entre nomes de usuários e IDs numéricos de usuários. Em outras palavras, `cd ~usuário` não vai funcionar, `ls -l` vai mostrar o id numérico ao invés do nome do usuário, e `find . -user joe -print` irá falhar com ```No such user"`. Para consertar isto, você vai precisar importar todas as entradas de usuários *sem permitir que eles acessem seus servidores*.

Isto pode ser feito adicionando outra linha ao `/etc/master.passwd`. Esta linha deve conter:

```
+:::::::::/sbin/nologin, significando ``Importe todas as entradas mas substitua o shell por /sbin/nologin nas entradas importadas." Você pode substituir qualquer campo na entrada passwd colocando um valor padrão em seu arquivo /etc/master.passwd.
```

Atenção Certifique-se que a linha `+:::::::::/sbin/nologin` está após `+@IT_EMP:::::::::`. Caso contrário, todas as contas de usuários importadas do NIS terão `/sbin/nologin` como seu shell de acesso.

Após esta mudança, você terá somente que modificar um mapa NIS se um novo funcionário juntar-se ao departamento de TI. Você pode usar uma abordagem similar para os servidores menos importantes substituindo a antiga `+:::::::::` em sua versão local do `/etc/master.passwd` por algo como isto:

```
+@IT_EMP:::::::::
+@IT_APP:::::::::
+:::::::::/sbin/nologin
```

As linhas correspondentes para operação normal de estações de trabalho podem ser:

```
+@IT_EMP:::::::::
+@USERS:::::::::
+:::::::::/sbin/nologin
```

E tudo estaria bem até que aconteça uma mudança na política algumas mudanças depois: o departamento de TI começa contratando estagiários. Os estagiários em TI são permitidos de acessarem as estações de trabalho normais e os servidores menos importantes; e os aprendizes de TI são permitidos acessar os servidores principais. Você adiciona um novo netgroup, *IT_INTERN*, adiciona os novos estagiários a este netgroup e começa a mudar a configuração em cada uma e em todas as máquinas... Como diz o velho ditado: ```Erros no planejamento centralizado levam à bagunça geral"`.

A capacidade do NIS de criar netgroups de outros netgroups pode ser usada para evitar situações como esta. Uma possibilidade é a criação de netgroups baseados em atividades. Por exemplo, você pode criar um netgroup chamado *BIGSRV* para definir as restrições de acesso para os servidores importantes, outro netgroup chamado *SMALLSRV* para os servidores menos importantes e um terceiro netgroup chamado

USERBOX para as estações de trabalho normais. Cada um desses netgroups contém os netgroups que são permitidos acessarem estas máquinas. As novas entradas para seu mapa NIS do netgroup deve ser parecer com este:

```
BIGSRV    IT_EMP IT_APP
SMALLSRV  IT_EMP IT_APP ITINTERN
USERBOX   IT_EMP ITINTERN USERS
```

Este método de definir restrições para acesso funcionam razoavelmente bem se você puder definir grupos de máquinas com restrições idênticas. Infelizmente, esta é a exceção e não a regra. Na maior parte do tempo, você vai precisar da capacidade de definir acesso (login) em cada máquina.

Definições de netgroup específico por máquina, são a outra possibilidade para tratar com a mudança de política delineada acima. Neste cenário, o `/etc/master.passwd` de cada caixa contém duas linhas começando por ```+`". A primeira delas adiciona um netgroup com contas com permissão de acesso nesta máquina, a segunda adiciona todas as demais contas com `/sbin/nologin` como shell. É uma boa idéia usar a versão em maiúsculas do nome da máquina como nome do netgroup. Em outras palavras, as linhas devem ficar parecidas com estas:

```
+@BOXNAME:::::::::
+:::::::::/sbin/nologin
```

Uma vez que você tenha completado esta tarefa para todas as suas máquinas, você não terá que modificar as versões locais do `/etc/master.passwd` nunca mais. Todas as alterações subsequentes podem ser tratadas modificando o mapa NIS. Eis um exemplo de um possível mapa de netgroup para este cenário com alguns brindes adicionais.

```
# Primeiro, defina grupos de usuários
IT_EMP    (,alpha,dominio-teste)    (,beta,dominio-teste)
IT_APP    (,charlie,dominio-teste)  (,delta,dominio-teste)
DEPT1     (,echo,dominio-teste)     (,foxtrott,dominio-teste)
DEPT2     (,golf,dominio-teste)     (,hotel,dominio-teste)
DEPT3     (,india,dominio-teste)    (,juliet,dominio-teste)
ITINTERN  (,kilo,dominio-teste)     (,lima,dominio-teste)
D_INTERNS (,able,dominio-teste)     (,baker,dominio-teste)
#
# Agora, defina alguns grupos baseados em funções
USERS     DEPT1    DEPT2        DEPT3
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP    ITINTERN
USERBOX   IT_EMP  ITINTERN  USERS
#
# Adicione grupos para tarefas especiais
# Permite acesso echo e golf acessarem nossa máquina de antivírus
SECURITY IT_EMP  (,echo,dominio-teste) (,golf,dominio-teste)
#
# netgroups baseados em máquinas
# Nossos principais servidores
WAR       BIGSRV
FAMINE    BIGSRV
# Usuário india precisa de acesso a este servidor
POLLUTION BIGSRV  (,india,dominio-teste)
```

```

#
# Este e' realmente importante e precisa de mais restrições de acesso
DEATH      IT_EMP
#
# A maquina antivírus mencionada acima
ONE        SECURITY
#
# Restrinja a máquina a um único usuário
TWO        (,hotel,dominio-teste)
# [...more groups to follow]

```

Se você estiver usando algum tipo de banco de dados para gerenciar suas contas de usuário, você deve ser capaz de criar a primeira parte do mapa com as ferramentas de relatórios de seu banco de dados. Desta forma, novos usuários terão, automaticamente, acesso às máquinas.

Uma última palavra de precaução: Pode nem sempre ser recomendável usar netgroups baseados em máquinas. Se você estiver implantando um par, uma dúzia ou mesmo centenas de máquinas idênticas para laboratórios de estudantes, você deve usar netgroups baseados em funções ao invés de netgroups baseados em máquinas para manter o tamanho dos mapas NIS dentro de limites razoáveis.

19.9.8. Coisas Importantes para Lembrar-se

Existem ainda um par de coisas que você vai precisar fazer de forma diferente, agora que você está em um ambiente NIS.

- Cada vez que você desejar adicionar um usuário ao lab, você precisa adicioná-lo *somente* no servidor NIS mestre, e *você precisa lembrar-se de reconstruir os mapas NIS*. Se você esquecer de fazer isto, o novo usuário não será capaz de acessar em nenhum lugar, exceto no mestre NIS. Por exemplo, se precisássemos adicionar um novo usuário ``jsmith" ao lab, poderíamos fazer:
 - # pw useradd jsmith
 - # cd /var/yp
 - # make test-domain

Você também pode rodar `adduser jsmith` ao invés do `pw useradd jsmith`.

- *Mantenha as contas de administração fora dos mapas NIS*. Você não quer propagar contas administrativas e senhas para as máquinas que vão ter usuários que não devem ter acesso a estas contas.
- *Mantenha o mestre e o escravo NIS seguros, e minimize seus tempos de paradas*. Se alguém invadir ou simplesmente desligar estas máquinas, eles terão efetivamente deixaram muitas pessoas sem a capacidade de acessar o lab.

Este é a principal fraqueza de qualquer sistema de administração centralizada. Se você não proteger seus servidores NIS, você terá muitos usuários zangados!

19.9.9. Compatibilidade com NIS v1

O **ypserv** do FreeBSD tem algum suporte para atender clientes NIS v1. A implementação NIS do FreeBSD usa somente o protocolo NIS v2, entretanto, outras implementações incluem suporte para a v1 do protocolo, para manter compatibilidade com sistemas antigos. Os daemons **ypbind** fornecidos com estes sistemas tentarão estabelecer uma associação (ligação) com um servidor NIS v1 mesmo que eles nunca precisem dele (e eles podem insistir em fazer difusão na busca por um, mesmo depois de terem recebido uma resposta de um servidor v2). Note que enquanto o suporte para chamadas normais de clientes é fornecida, esta versão do ypserv não trata requisições de transferência de mapas v1; conseqüentemente, ele não pode ser usado como um mestre ou escravo em conjunto com servidores NIS mais antigos que suportam somente o protocolo v1. Felizmente, não devem mais haver tais servidores em uso nos dias atuais.

19.9.10. Servidores NIS Que Também São Clientes NIS

Cuidados devem ser tomados quando rodar o ypserv em um domínio multi-servidor onde as máquinas servidoras também são clientes NIS. Geralmente é uma boa idéia forçar os servidores a associarem-se a si próprios ao invés de permitir que façam difusão (broadcast) de solicitações para associações e possivelmente associem-se mutuamente. Estranhos modos de falhas podem resultar se um servidor sair do ar e outros forem dependentes dele. Eventualmente todos os clientes vão ter seu acesso expirado e tentarão se associar a outros servidores, mas o retardo envolvido pode ser considerável e o modo de falha ainda está presente, uma vez que os servidores podem associar-se mutuamente de novo.

Você pode forçar um sistema a associar-se a um servidor em particular rodando o ypbinding com o parâmetro *-S*. Se você não quiser fazer isto manualmente cada vez que inicializar o seu servidor NIS, você pode adicionar as seguintes linhas ao seu `/etc/rc.conf`:

```
nis_client_enable="YES"      # run client stuff as well
nis_client_flags="-S NIS domain,server"
```

Veja [ypbind\(8\)](#) para maiores informações.

19.9.11. Formatos de Senhas

Uma das questões mais comuns que as pessoas enfrentam quando tentam implementar o NIS, é a compatibilidade de formatos de senhas. Se o seu servidor NIS está usando senhas criptografadas com DES, ele somente suportará clientes que também estejam usando DES. Por exemplo, se você tem clientes NIS Solaris em sua rede, então você quase certamente precisa usar senhas criptografadas com DES.

Para verificar qual o formato seus servidores e clientes estão usando, olhe no `/etc/login.conf`. Se o sistema (host) está configurado para usar senhas criptografadas com DES, então a classe *padrão* (*default*) vai conter uma entrada como esta:

```
default:\
    :passwd_format=des:\
```



```
:copyright=/etc/COPYRIGHT:\
[Entradas adicionais omitidas]
```

Outros possíveis valores para a capacidade *passwd_format*, incluem *blf* e *md5* (para senhas criptografadas com Blowfish e MD5, respectivamente).

Se você fez mudanças no `/etc/login.conf`, você também vai precisar reconstruir a base de dados de capacidade de login, a qual é feita executando o seguinte comando como `root`:

```
# cap_mkdb /etc/login.conf
```

Nota: O formato de senhas já presentes no `/etc/master.passwd` não será atualizado até que um usuário modifique sua senha pela primeira vez *depois* da reconstrução da base de dados de capacidade.

Em seguida, para garantir que senhas estão criptografadas com o formato que você escolheu, você também deve verificar que o *crypt_default* no `/etc/auth.conf` dá precedência para o formato de senha da sua escolha. Para fazer isto, coloque o formato que você escolheu em primeiro na lista. Por exemplo, quando usar senhas criptografadas com DES, a entrada deve ser:

```
crypt_default = des blf md5
```

Tendo seguido os passos acima em cada um dos servidores e clientes NIS baseados em FreeBSD, você pode estar certo de que todos eles concordam em qual formato de senha é usado na sua rede. Se você tem problemas com autenticação em um cliente NIS, este é um bom lugar para começar a procurar por possíveis problemas. Lembre-se: se você quer implementar um servidor NIS em uma rede heterogênea, você provavelmente terá que usar DES em todos os seus sistemas, porque é o padrão de menor denominador comum.

19.10. DHCP

Escrito por Greg Sutter.

19.10.1. O Que É DHCP?

DHCP, o Protocolo de Configuração Dinâmica de Servidor (Dynamic Host Configuration Protocol, Descreve os meios pelo qual um sistema pode se conectar a uma rede e obter a informação necessária para comunicação naquela rede. O FreeBSD usa a implementação de DHCP da ISC (Internet Systems Consortium), então toda informação dependente de implementação aqui disponível é para ser utilizada com a distribuição da ISC.

19.10.2. O Que Esta Seção Cobre

Esta seção descreve os componentes cliente e servidor do sistema DHCP da ISC. O programa cliente, `dhclient`, já vem integrado no FreeBSD e a parte servidor está disponível do port [net/isc-dhcp3](#). As páginas de manual [dhclient\(8\)](#), [dhcp-options\(5\)](#), e [dhclient.conf\(5\)](#), adicionalmente às referências disponíveis abaixo, são recursos úteis.

19.10.3. Como Funciona

Quando o `dhclient`, o cliente DHCP, é executado na máquina cliente, ele começa a transmissão por difusão de requisições de informações de configuração. Por padrão, estas solicitações estão na porta UDP 68. O servidor responde na UDP 67, dando ao cliente um endereço IP e outras informações de rede, como a máscara de rede, roteador e servidores DNS. Toda esta informação é fornecida na forma de um "arrendamento (lease)" e é válido somente por um determinado período (configurado pelo mantenedor do servidor DHCP). Desta forma, endereços IP atribuídos a clientes que não estão mais conectados à rede, podem ser reaproveitados automaticamente.

Clientes DHCP podem obter uma grande quantidade de informações do servidor. Uma lista completa pode ser encontrada em [dhcp-options\(5\)](#).

19.10.4. Integração com o FreeBSD

O FreeBSD integra totalmente o cliente DHCP da ISC, o `dhclient`. O suporte ao cliente DHCP é fornecido tanto no instalador quanto no sistema base, evitando a necessidade de conhecimento detalhado das configurações em qualquer rede que tenha um servidor DHCP. O `dhclient` foi incluído em todas as distribuições do FreeBSD, desde a versão 3.2

O DHCP é suportado pelo `sysinstall`. Quando configurar uma interface de rede através do `sysinstall`, a primeira pergunta feita é, "Do you want to try DHCP configuration of this interface? ("Você deseja tentar configuração DHCP nesta interface?")"

Respondendo afirmativamente, será executado o `dhclient`, e se funcionar, as informações da configuração de rede serão preenchidas automaticamente.

Existem duas coisas que você precisa fazer para ter seu sistema usando DHCP na inicialização:

- Certifique-se de que o dispositivo `bpf` está compilado no seu kernel. Para fazer isso, adicione o *pseudo-device* `bpf` no seu arquivo de configuração do kernel e reconstrua o kernel. Para maiores informações sobre construção de kernels, veja [Capítulo 9](#).

O dispositivo `bpf` já é parte do kernel `GENERIC` fornecido com o FreeBSD, desta forma, se você não tem um kernel adaptado, você não precisa criar um para ter o DHCP funcionando.

Nota: Para os que são particularmente preocupados com segurança, você deve estar ciente de que o `bpf` também é o dispositivo que permite o funcionamento correto de analisadores de pacotes (embora ainda precisem ser executados pelo usuário `root`). O `bpf` é obrigatório para usar DHCP, mas se você é muito sensível com relação a segurança, você provavelmente não deve adicionar o `bpf` ao seu kernel na expectativa de que em algum momento no futuro você vai utilizar DHCP.

- Edite o seu `/etc/rc.conf` para incluir o seguinte:
- `ifconfig_fxp0="DHCP"`

Nota: Certifique-se de substituir `fxp0` pela designação da interface que você deseja configurar automaticamente, como descrito em [Seção 6.8](#).

Se você está usando um local diferente para o `dhclient`, ou se você deseja passar parâmetros adicionais para o comando, inclua também o seguinte (editando, se necessário):

```
dhcp_program="/sbin/dhclient"  
dhcp_flags=""
```

O servidor DHCP, `dhcpd`, é incluído como parte do port [net/isc-dhcp3](#) na coleção de ports. Este port contém a distribuição completa do DHCP da ISC, contendo o cliente, servidor, agente de relay e documentação.

19.10.5. Arquivos

- `/etc/dhclient.conf`

O `dhclient` requer um arquivo de configuração `/etc/dhclient.conf`. Tipicamente o arquivo contém somente comentários e os padrões mais razoáveis definidos. Este arquivo de configuração é descrito pela página de manual [dhclient.conf\(5\)](#)

- `/sbin/dhclient`

O `dhclient` é compilado estaticamente e reside em `/sbin`. A página de manual [dhclient\(8\)](#) fornece mais informações sobre o `dhclient`.

- `/sbin/dhclient-script`

O `dhclient-script` é o script de configuração do cliente DHCP específico do FreeBSD. É descrito em [dhclient-script\(8\)](#), mas você não deve precisar de qualquer modificação de usuário para seu funcionamento.

- `/var/db/dhclient.leases`

O cliente DHCP mantém uma base de dados de arrendamentos neste arquivo, que é gravado como um registro (log). [dhclient.leases\(5\)](#) oferece uma descrição um pouco mais longa.

19.10.6. Leitura Complementar

O protocolo DHCP é descrito totalmente na [RFC 2131](#). Um recurso de informações também foi montado em [dhcp.org](#).

19.10.7. Instalando e Configurando um Servidor DHCP

19.10.7.1. O Que Esta Seção Cobre

Esta seção fornece informações sobre como configurar um sistema FreeBSD para atuar como servidor DHCP usando a implementação da ISC da suíte DHCP.

A parte servidora da suíte não é fornecida no FreeBSD, então você vai precisar instalar o port [net/isc-dhcp3](#) para ativar este serviço. Veja [Capítulo 4](#) para mais informações sobre uso da coleção de ports.

19.10.7.2. Instalação do Servidor DHCP

Para configurar seu sistema servidor FreeBSD como um servidor DHCP, você vai precisar garantir que o dispositivo [bpf\(4\)](#) está compilado no seu kernel. Para isto, adicione o *pseudo-device* `bpf` ao seu arquivo de configuração do kernel e reconstrua o kernel. Para mais informações sobre construção de kernels, veja [Capítulo 9](#).

O dispositivo `bpf` já é parte do kernel `GENERIC`, fornecido com o FreeBSD, assim você não precisa criar um kernel adaptado para ter o DHCP funcionando.

Nota: Aqueles particularmente preocupados com segurança, devem observar que o `bpf` também é o dispositivo que permite aos analisadores de pacotes executarem seu trabalho (apesar de tais programas precisarem de acesso privilegiado). O `bpf` é exigido para uso do DHCP, mas se você é muito preocupado com questões de segurança, você provavelmente não deve incluir o `bpf` em seu kernel simplesmente por supor que vai usar DHCP em algum momento no futuro.

A próxima coisa que você vai precisar fazer é editar o exemplo `dhcpd.conf` que foi incluído pelo port [net/isc-dhcp3](#). Por padrão, este será o arquivo

```
/usr/local/etc/dhcpd.conf.sample, e você deve copiar este arquivo para  
/usr/local/etc/dhcpd.conf antes de modificá-lo.
```

19.10.7.3. Configurando o Servidor DHCP

O arquivo `dhcpd.conf` é composto de declarações relativas a subredes e servidores, e é, talvez, melhor explicado usando um exemplo:

```
option domain-name "exemplo.com"; ❶
option domain-name-servers 192.168.4.100; ❷
option subnet-mask 255.255.255.0; ❸

default-lease-time 3600; ❹
max-lease-time 86400; ❺
ddns-update-style none; ❻

subnet 192.168.4.0 netmask 255.255.255.0 {
    range 192.168.4.129 192.168.4.254; ❼
    option routers 192.168.4.1; ❽
}

host mailhost {
    hardware ethernet 02:03:04:05:06:07; ❾
    fixed-address mailhost.exemplo.com; ❿
}
```

❶

Esta opção especifica o domínio que será fornecido aos clientes como o principal domínio de busca. Veja [resolv.conf\(5\)](#) para mais informações sobre o que isso significa.

❷

Esta opção especifica uma lista separada por vírgulas de servidores DNS que o cliente deve utilizar.

❸

A máscara de rede que será fornecida aos clientes.

❹

Um cliente pode requerer um período de tempo específico válido para um arrendamento. Senão o servidor deverá fazer um arrendamento com este prazo de expiração (em segundos).

❺

Este é o maior período de tempo permitido que o servidor deverá permitir um arrendamento. Se um cliente solicitar um período maior, um arrendamento será efetuado, entretanto, será válido somente por *max-lease-time* segundos.

❻

Esta opção especifica se o servidor DHCP deverá tentar atualizar o DNS quando um arrendamento é aceito ou devolvido. Na implementação da ISC, esta opção é *obrigatória*.

7

Isto estipula quais endereços IP devem ser usados no conjunto reservado para alocação aos clientes. Endereços IP dentro da faixa inclusive os discriminados são distribuídos aos clientes.

8

Declara o gateway padrão que será fornecido aos clientes.

9

O endereço MAC de um sistema (de forma que o servidor DHCP possa reconhecê-lo quando dele receber uma solicitação).

10

Especifica que o sistema sempre deve receber o mesmo endereço IP. Note que um hostname funciona aqui, desde que o servidor DHCP seja capaz de resolver o nome via DNS antes de responder à solicitação com as informações do arrendamento.

Quando você terminar de escrever o seu `dhcpd.conf` você pode dar prosseguimento, iniciando o servidor com o seguinte comando:

```
# /usr/local/etc/rc.d/isc-dhcpd.sh start
```

Se futuramente você precisar realizar alterações na configuração de seu servidor, é importante notar que enviar um sinal `SIGHUP` para o **dhcpd** não irá recarregar as configurações, como na maioria dos daemons. Você precisa enviar um sinal `SIGTERM` para parar o processo, e então, reiniciá-lo com o comando acima.

19.10.7.4. Arquivos

- `/usr/local/sbin/dhcpd`

O **dhcpd** é compilado estaticamente e reside em `/usr/local/sbin`. A página de manual [dhcpd\(8\)](#) instalada com o port fornece mais informações sobre o **dhcpd**.

- `/usr/local/etc/dhcpd.conf`

O **dhcpd** requer um arquivo de configuração, `/usr/local/etc/dhcpd.conf` antes de começar a prestar serviços para os clientes. O arquivo precisa conter todas as informações que devem ser fornecidas aos clientes sendo atendidos, juntamente com informações sobre a operação do servidor. Este arquivo de configuração é descrito pela página de manual [dhcpd.conf\(5\)](#), instalada pelo port.

- `/var/db/dhcpd.leases`

O servidor DHCP mantém uma base de dados de arrendamentos que distribuiu neste arquivo, o qual é escrito como um arquivo de registro (log). A página de

manual [dhcpcd.leases\(5\)](#), instalada pelo port fornece uma descrição um pouco mais longa.

- `/usr/local/sbin/dhcrelay`

O **dhcrelay** é usado em ambientes avançados onde um servidor DHCP repassa uma solicitação de um cliente para outro servidor DHCP em uma rede separada. A página de manual [dhcrelay\(8\)](#) fornecida com o port contém mais detalhes.

19.11. DNS

Contribuído por Chern Lee.

19.11.1. Visão Geral

O FreeBSD usa, por padrão, uma versão do BIND (Berkeley Internet Name Domain), que é a implementação mais comum do protocolo DNS. O DNS é o protocolo pelo qual nomes são mapeados para endereços IP e vice-versa. Por exemplo, uma pergunta por `www.FreeBSD.org` receberá uma resposta com o endereço IP do servidor web do Projeto FreeBSD, ao passo que, uma pergunta por `ftp.FreeBSD.org` retornará o endereço IP da máquina FTP correspondente. Da mesma forma, o oposto pode ocorrer. Uma pergunta pelo endereço IP pode resolver seu nome de sistema. Não é necessário rodar um servidor de nomes para efetuar pesquisas DNS em um sistema.

O DNS é coordenado através da Internet através de um sistema um tanto complexo de servidores de nome raiz autoritativos, e outros servidores de nome de menor escala que abrigam e fazem cache de informações de domínios individuais.

Este documento refere-se ao BIND 8.x, como ele é a versão estável usada no FreeBSD. O BIND 9.x no FreeBSD pode ser instalado através do port [net/bind9](#).

As RFC1034 e RFC1035 ditam o protocolo DNS.

Atualmente, o BIND é mantido pelo [Internet Software Consortium \(www.isc.org\)](#).

19.11.2. Terminologia

Para compreender este documento, alguns termos relativos ao DNS precisam ser entendidos.

Termo	Definição
DNS Direto	Mapeamento de nomes para endereços IP
Origem	Refere-se ao domínio abrangido em um determinado arquivo de zona

Termo	Definição
named , BIND, servidor de nome	Nomes comuns para o pacote do servidor de nomes BIND, incluso no FreeBSD
Resolver	Um processo do sistema através do qual uma máquina pesquisa num servidor de nomes por informações sobre zonas
DNS reverso	O oposto do DNS direto; mapeamento de endereços IP para nomes
Zona Raiz (Root Zone)	O início da hierarquia de zona da Internet. Todas as zonas recaem sob a zona raiz, similar a como todos os arquivos em um sistema de arquivos localizam-se abaixo do diretório raiz.
Zona	Um domínio individual, subdomínio, ou parte de um DNS administrado por uma mesma autoridade

Exemplos de zonas:

- `.` é a zona raiz
- `org.` é a zona sob a zona raiz
- `exemplo.org` é a zona sob a zona `org.`
- `foo.exemplo.org.` é um subdomínio, uma zona sob a zona `exemplo.org.`
- `1.2.3.in-addr.arpa` é a zona referenciando todos os endereços IP que recaem sob o espaço IP `3.2.1.*`

Como se pode ver, a parte mais específica de um nome de sistema aparece à sua esquerda. Por exemplo, `exemplo.org` é mais específico que `org.`, como `org.` é mais específico que a zona raiz. O esboço de cada parte de um nome de sistema é muito parecido com um sistema de arquivos: o diretório `/dev` fica sob o raiz, e assim por diante.

19.11.3. Razões para Rodar um Servidor de Nome

Servidores de nome normalmente aparecem em duas formas: como servidor de nome autoritativo e, como um servidor de nome para fazer cache.

Um servidor de nome autoritativo é necessário quando:

- alguém quer servir informação DNS para o mundo, respondendo de forma autoritativa a pesquisas.
- um domínio, como `exemplo.org`, é registrado e endereços IP precisam ser designados a nomes de sistemas sob ele.
- um bloco de endereços IP requerem entradas DNS reversas (IP para nomes de sistema).

- um servidor de nome de segurança, chamado de escravo, precisa responder a pesquisas quando o primário está fora do ar ou inacessível.

Um servidor de nome para fazer cache é necessário quando:

- um servidor DNS local pode fazer cache e responder mais rapidamente do que pesquisar em um servidor de nome externo.
- uma redução no tráfego de rede de uma forma geral é desejada (o tráfego DNS foi contabilizado como 5% ou mais do tráfego total da Internet).

Quando alguém pesquisa por `www.FreeBSD.org`, o resolver normalmente pesquisa o servidor de nomes do provedor de nível superior e recupera a resposta. Com um servidor DNS local fazendo cache, a pesquisa precisa ser feita uma vez para o mundo exterior. Cada pesquisa adicional não vai precisar buscar fora da rede local, uma vez que há uma cópia local da informação no cache.

19.11.4. Como Funciona

No FreeBSD, o daemon BIND é chamado **named** por razões óbvias.

Arquivo	Descrição
named	o daemon do BIND
<code>ndc</code>	programa de controle do daemon de nome (name daemon control)
<code>/etc/namedb</code>	diretório onde reside a informação de zona do BIND
<code>/etc/namedb/named.conf</code>	arquivo de configuração do daemon

Arquivos de zona normalmente pertencem ao diretório `/etc/namedb`, e contém a informação de zona DNS fornecida pelo servidor de nomes.

19.11.5. Iniciando o BIND

Uma vez que o BIND é instalado por padrão, configurá-lo é relativamente simples.

Para assegurar que o daemon `named` é executado durante a inicialização, coloque as seguintes modificações no `/etc/rc.conf`:

```
named_enable="YES"
```

Para iniciar o daemon manualmente (após configurá-lo)

```
# ndc start
```

19.11.6. Arquivos de Configuração

19.11.6.1. Usando `make-localhost`

Certifique-se de fazer:

```
# cd /etc/namedb
# sh make-localhost
```

para criar adequadamente o arquivo de zona DNS reverso local no `/etc/namedb/localhost.rev`.

19.11.6.2. `/etc/namedb/named.conf`

```
// $FreeBSD$
//
// Consulte a página de manual do named(8) para detalhes. Se você
// alguma vez for configurar um servidor primário, certifique-se de
// que compreendeu os detalhes cabeludos de como o DNS está
// funcionando. Mesmo com simples enganos, você pode quebrar a
// conectividade de grupos atingidos, ou causar enormes quantidades de
// tráfego inútil na Internet.

options {
    directory "/etc/namedb";

// Adicionalmente à cláusula "forwarders", você pode forçar seu
// servidor de nomes a nunca iniciar pesquisas por conta própria, mas
// sempre consultar somente seus despachantes, ativando a linha
// abaixo:
//
//     forward only;

// Se você tem um servidor DNS por perto em seu provedor de acesso,
// entre seu endereço IP aqui, e ative a linha abaixo. Isto fará com
// que você se beneficie de seu cache, reduzindo o tráfego DNS na
// Internet em geral.

/*
    forwarders {
        127.0.0.1;
    };
*/
```

Como diz o comentário, para se beneficiar de um cache de acesso, *despachantes* (*forwarders*) podem ser ativados aqui. Em circunstâncias normais, um servidor de nome irá pesquisar a Internet de forma recursiva em determinados servidores de nomes até que encontre a resposta que procura. Com este recurso ativado, ele terá que consultar o servidor de nome do provedor de acesso (ou o servidor de nome configurado) primeiro, tirando vantagem de seu cache. Se o servidor de nome em questão estiver com acessos de alta velocidade, um servidor de nomes rápido, ativar este recurso pode valer a pena.

Atenção `127.0.0.1` não vai funcionar aqui. Mude este endereço IP para um servidor de nome localizado em seu provedor de acesso.


```
// Nunca esqueça de incluir a zona de busca reversa (IN-ADDR.ARPA)!
// (Estes são os primeiros bytes do respectivo endereço IP, em ordem
// invertida, com ".IN-ADDR.ARPA" anexado.)
//
// Entretanto, antes de iniciar a configurar a zona primária, é bom
// ter
// compreendido completamente como o DNS e o BIND trabalham. Algumas
// vezes, existem ciladas ocultas. Em comparação, é mais fácil
// configurar um secundário.
//
// NB: Não ative cegamente os exemplos abaixo. :-) Ao invés, use
// nomes e endereços reais.
//
// NOTA!!! O FreeBSD roda bind em uma sandbox (veja named_flags no
// rc.conf). O diretório contendo as zonas secundárias devem possuir
// acesso de gravação permitido para o bind. É sugerida a seguinte
// seqüência:
//
//     mkdir /etc/namedb/s
//     chown bind:bind /etc/namedb/s
//     chmod 750 /etc/namedb/s
```

Para mais informações sobre como rodar o BIND numa sandbox, veja [Rodando o named em uma sandbox](#).

```
/*
zone "exemplo.com" {
    type slave;
    file "s/exemplo.com.bak";
    masters {
        192.168.1.1;
    };
};

zone "0.168.192.in-addr.arpa" {
    type slave;
    file "s/0.168.192.in-addr.arpa.bak";
    masters {
        192.168.1.1;
    };
};
*/
```

No `named.conf` existem exemplos de entradas secundárias para as zonas direta e reversa.

Para cada nova zona servida, uma nova entrada de zona precisa ser adicionada ao `named.conf`

Por exemplo, a entrada mais simples para a zona `exemplo.org` pode ficar assim:

```
zone "exemplo.org" {
    type master;
    file "exemplo.org";
};
```

A zona é primária, como indicado pela declaração `type`, mantendo a informação da zona no `/etc/namedb/exemplo.org` indicado pela declaração `file`.

```
zone "exemplo.org" {
    type slave;
    file "exemplo.org";
};
```

No caso do secundário, a informação da zona é transferida do servidor de nomes primário de uma zona em particular, e salvo no arquivo especificado. Se e quando o servidor primário morrer, ou estiver fora de alcance, o servidor de nomes secundário terá a informação da zona transferida e será capaz de servi-la.

19.11.6.3. Arquivos de Zona

Um exemplo de arquivo de zona primário para `exemplo.org` (existente no `/etc/namedb/example.org`) segue abaixo:

```
$TTL 3600

exemplo.org. IN SOA ns1.exemplo.org. admin.exemplo.org. (
    5      ; Serial
    10800 ; Refresh
    3600  ; Retry
    604800 ; Expire
    86400 ) ; Minimum TTL

; Servidores DNS
@           IN NS      ns1.exemplo.org.
@           IN NS      ns2.exemplo.org.

; Nomes de Máquinas
localhost  IN A       127.0.0.1
ns1        IN A       3.2.1.2
ns2        IN A       3.2.1.3
mail       IN A       3.2.1.10
@          IN A       3.2.1.30

; Apelidos (aliases)
www        IN CNAME   @

; Registro MX (MX Record)
@          IN MX     10 mail.exemplo.org.
```

Note que cada nome de sistema que termina com um ``." é um nome exato, ao passo que tudo sem um ``." no final é referenciado à origem. Por exemplo, `www` é traduzido para `www + origem`. Em nosso arquivo de zona fictício, nossa origem é `exemplo.org.`, então `www` será traduzido para `www.exemplo.org.`

O formato de um arquivo de zona é como segue:

```
nomedoregistro      IN tipodoregistro  valor
```

Os registros DNS mais freqüentemente usados:

SOA

início da zona de autoridade

NS

um servidor de nome autoritativo

A

Um endereço de sistema (host address)

CNAME

o nome canônico para um apelido (alias)

MX

servidor de correio (mail exchanger)

PTR

um ponteiro de nome de domínio (usado em DNS reverso)

```
exemplo.org. IN SOA ns1.exemplo.org. admin.exemplo.org. (  
    5      ; Serial  
    10800  ; Atualizar após 3 horas  
    3600   ; Tentar novamente após 1 hora  
    604800 ; Expirar após 1 semana  
    86400  ) ; TTL mínimo de 1 dia  
exemplo.org.
```

o nome de domínio, que também é a origem para este arquivo de zona

ns1.exemplo.org.

o servidor de nome primário/autoritativo para esta zona

admin.exemplo.org.

o endereço de correio eletrônico da pessoa responsável por esta zona, com a @ trocada. (<admin@exemplo.org> é substituído por *admin.exemplo.org*)

5

o número de série do arquivo. Deve ser incrementado toda vez que o arquivo de zona é alterado. Hoje em dia, muitos administradores preferem o formato *aaaammddii* para o número de série. 2001041002 quer dizer que a última modificação foi feita em 10/04/2001, e os dígitos 02 ao final, significam que foi a segunda vez que o arquivo de zona foi alterado nesse dia.

```
@      IN NS      ns1.exemplo.org.
```

Esta é uma entrada *NS*. Cada servidor de nome que responderá autoritativamente pela zona deve ter uma destas entradas. A @ como aqui mostrada poderia ter sido exemplo.org. A @ é traduzida pela origem.

```
localhost    IN A      127.0.0.1
ns1          IN A      3.2.1.2
ns2          IN A      3.2.1.3
mail         IN A      3.2.1.10
@            IN A      3.2.1.30
```

O registro *A* indica nomes de máquinas. Como mostrado acima, ns1.exemplo.org resolveria para 3.2.1.2. De novo, o símbolo da origem, @, é usado aqui, significando que o exemplo.org resolveria para 3.2.1.30.

```
www          IN CNAME  @
```

O registro de nome canônico é geralmente usado para dar apelidos (aliases) para uma máquina. No exemplo, www é apelidado como a máquina endereçada para a origem, ou exemplo.org (3.2.1.30). *CNAMEs* podem ser usados para fornecer apelidos a nomes de sistemas, ou efetuar rotatividade (round-robin) de um nome de sistema entre várias máquinas.

```
@            IN MX     10      mail.exemplo.org.
```

O registro *MX* indica quais servidores de correio são responsáveis por tratar correspondência de entrada para a zona. mail.exemplo.org é o nome do servidor de correio, e 10 a prioridade daquele servidor de correio.

Alguém pode ter vários servidores de correio, com prioridades de 3, 2 e 1. Um servidor de correio tentando fazer uma entrega para exemplo.org tentaria primeiro o *MX* de maior prioridade, depois o de segunda maior e assim por diante, até que o correio seja entregue corretamente.

Para arquivos de zona in-addr.arpa (DNS reverso), o mesmo formato é usado, exceto pelo fato de conter entradas *PTR* ao invés de *A* ou *CNAME*.

```
$TTL 3600
```

```
1.2.3.in-addr.arpa.  IN SOA ns1.exemplo.org.  admin.exemplo.org. (
                    5      ; Serial
                    10800   ; Refresh
                    3600    ; Retry
                    604800  ; Expire
                    3600   ) ; Minimum
```

```
@            IN NS     ns1.exemplo.org.
@            IN NS     ns2.exemplo.org.
```

```
2            IN PTR     ns1.exemplo.org.
3            IN PTR     ns2.exemplo.org.
10           IN PTR     mail.exemplo.org.
30           IN PTR     exemplo.org.
```

Este arquivo fornece o endereço IP adequado aos mapeamentos de nomes de sistemas de nosso domínio fictício acima.

19.11.7. Servidor Cache de Nomes

Um servidor de nome para cache é um tipo de servidor de nome que não é autoritativo autoritativo para nenhuma zona. Ele simplesmente efetua buscas por conta própria e se recorda delas para uso futuro. Para configurar um, apenas configure o servidor de nome normalmente, omitindo quaisquer inclusões de zonas.

19.11.8. Rodando o `named` Restrito a Um Diretório (Sandbox)

Para segurança adicional, você pode querer rodar o [named\(8\)](#) com um usuário não privilegiado e configurá-lo para fazer [chroot\(8\)](#) dentro de um diretório restrito (sandbox). Isto torna tudo fora da sandbox inacessível para o daemon **named**. Caso o **named** seja atacado, isto ajudará a reduzir os danos que podem ser causados. Por padrão, o FreeBSD tem um usuário e um grupo chamados `bind`, planejado para este uso.

Nota: Diversas pessoas recomendariam que ao invés de configurar o **named** para fazer `chroot`, você deveria rodá-lo dentro de um [jail\(8\)](#). Esta seção não tenta abordar esta situação.

Uma vez que o **named** não será capaz de acessar nada fora da sandbox, (como bibliotecas compartilhadas, sockets de logs e assim por diante), há uma série de passos que devem ser seguidos para permitir o correto funcionamento do **named**. Na seguinte lista de verificação, é assumido que o caminho para a sandbox é `/etc/namedb` e que você não fez modificações prévias ao conteúdo deste diretório. Execute os seguintes passos como usuário `root`.

- Crie todos os diretórios que o **named** espera encontrar:
- `# cd /etc/namedb`
- `# mkdir -p bin dev etc var/tmp var/run master slave`
- `# chown bind:bind slave var/*` ❶

O **named** precisa de acesso para gravação somente nestes diretórios, então é tudo que vamos lhe fornecer.

- Rearrume e crie uma zona básica e arquivos de configuração:
- `# cp /etc/localtime etc` ❶
- `# mv named.conf etc && ln -sf etc/named.conf`
- `# mv named.root master`
- `# sh make-localhost && mv localhost.rev localhost-v6.rev master`
- `# cat > master/named.localhost`
- `$ORIGIN localhost.`
- `$TTL 6h`
- `@ IN SOA localhost. postmaster.localhost. (`
- `1 ; serial`
- `3600 ; refresh`

- `1800 ; retry`
- `604800 ; expiration`
- `3600) ; minimum`
- `IN NS localhost.`
- `IN A 127.0.0.1`
- `^D`



Isto permite ao **named** registrar a hora correta ao [syslogd\(8\)](#)

- Se você estiver rodando a versão de FreeBSD anterior à 4.9-RELEASE, construa uma cópia estaticamente compilada do **named-xfer** e a copie na **sandbox**:
 - `# cd /usr/src/lib/libisc`
 - `# make cleandir && make cleandir && make depend && make all`
 - `# cd /usr/src/lib/libbind`
 - `# make cleandir && make cleandir && make depend && make all`
 - `# cd /usr/src/libexec/named-xfer`
 - `# make cleandir && make cleandir && make depend && make NOSHARED=yes all`
 - `# cp named-xfer /etc/namedb/bin && chmod 555 /etc/namedb/bin/named-xfer`

Após a instalação do seu **named-xfer** compilado estaticamente, uma faxina se faz necessária, para evitar deixar cópias envelhecidas de bibliotecas ou programas na sua árvore de fontes:

```
# cd /usr/src/lib/libisc
# make cleandir
# cd /usr/src/lib/libbind
# make cleandir
# cd /usr/src/libexec/named-xfer
# make cleandir
```



Já houveram relatos de que este passo pode ocasionalmente falhar. Se isto acontecer com você, então use o comando:

```
# cd /usr/src && make cleandir && make cleandir
```

and delete your `/usr/obj` tree:

```
# rm -fr /usr/obj && mkdir /usr/obj
```

Isto limpará quaisquer ``restos" de sua árvore de fontes, e, tentar novamente os passos acima deverá funcionar.

Se você está rodando o FreeBSD versão 4.9-RELEASE ou posterior, então a cópia do **named-xfer** em `/usr/libexec` é compilada estaticamente por padrão, e você pode simplesmente usar [cp\(1\)](#) para copiá-la dentro da sua **sandbox**.

- Crie um `dev/null` o qual o **named** possa ver e escrever:
- `# cd /etc/namedb/dev && mknod null c 2 2`


```

    allow-transfer { localhost; };
    notify no;
};
zone "." IN {
    type hint;
    file "master/named.root";
};
zone "private.example.net" in {
    type master;
    file "master/private.example.net.db";
    allow-transfer { 192.168.10.0/24; };
};
zone "10.168.192.in-addr.arpa" in {
    type slave;
    masters { 192.168.10.2; };
    file "slave/192.168.10.db"; ❹
};

```

❶

A declaração *directory* é especificada como /, uma vez que todos os arquivos que o **named** precisa estão dentro deste diretório (lembre-se de que isto equivale ao /etc/namedb de um usuário "normal").

❷

Especifica o caminho completo para o arquivo binário `named-xfer` (do sistema de referência do **named**). Isto é necessário, uma vez que o **named** é compilado para, por padrão, buscar o `named-xfer` no `/usr/libexec`.

❸

Especifica o nome do arquivo (relativo à declaração *directory* acima) onde o **named** pode encontrar o arquivo de zona para esta zona.

❹

Especifica o nome do arquivo (relativo à declaração *directory* acima, onde o **named** deve gravar uma cópia do arquivo de zona para esta zona após transferir a mesma com sucesso do servidor mestre. Por isto que precisamos mudar a posse do diretório `slave` para `bind` nos passos de configuração acima.

Após completar os passos acima, reinicie seu servidor ou o [syslogd\(8\)](#) e inicie o [named\(8\)](#), certificando-se de usar as novas opções especificadas em `syslogd_flags` e `named_flags`. Você agora deve estar rodando uma cópia restrita do **named**!

19.11.9. Segurança

Apesar do BIND ser a implementação mais comum do DNS, sempre há a questão da segurança. Furos possíveis e exploráveis de segurança são encontrados de vez em quando.

É uma boa idéia assinar [CERT](#) e [freebsd-security-notifications](#) para se manter atualizado sobre as atuais questões de segurança na Internet e do FreeBSD.

Dica: Se um problema surgir, manter os fontes atualizados e ter uma compilação recente do named não é má idéia.

19.11.10. Leitura adicional

Páginas de manual do BIND/named: [ndc\(8\)](#) [named\(8\)](#) [named.conf\(5\)](#)

- [Página Oficial do ISC Bind](#)
 - [BIND FAQ](#)
 - [O'Reilly DNS and BIND 4th Edition](#)
 - [RFC1034 - Domain Names - Concepts and Facilities](#)
 - [RFC1035 - Domain Names - Implementation and Specification](#)
-

19.12. NTP

Contribuído por Tom Hukins.

19.12.1. Visão Geral

Ao longo do tempo, o relógio de um computador tende a variar. Enquanto o tempo passa, o relógio do computador se torna menos preciso. O NTP (Network Time Protocol) é uma forma de assegurar que o seu relógio está correto.

Muitos serviços Internet dependem, ou muito se beneficiam, da precisão do relógio do computador estar correto. Por exemplo, um servidor Web pode receber solicitações para enviar um arquivo se ele foi modificado desde uma certa hora. Serviços como o [cron\(8\)](#) executam comandos em uma hora específica. Se o relógio está errado, estes comandos podem não ser executados no momento esperado.

O FreeBSD é disponibilizado com o servidor [ntpd\(8\)](#) que pode ser usado para consultar outros servidores NTP para ajustar o relógio em sua máquina ou fornecer serviços de hora para outros.

19.12.2. Escolhendo os Servidores NTP Adequados

Para sincronizar seu relógio, você vai precisar encontrar um ou mais servidores NTP para usar. Seu administrador de rede ou Provedor de Serviços Internet podem ter configurado um servidor NTP para este propósito--verifique sua documentação para confirmar se este é o caso. Existe uma [lista de servidores NTP publicamente acessíveis](#) os quais podem ser usados para encontrar um servidor NTP próximo a você. Certifique-se de estar ciente da política de qualquer servidor escolhido e, se necessário, solicite permissão para uso.

Escolher alguns servidores NTP não conectados é uma boa idéia no caso de algum dos servidores que você estiver usando se tornar inalcançável ou seu relógio não for

confiável. [ntpd\(8\)](#) usa as respostas recebidas de outros servidores de forma inteligente--ele vai favorecer os servidores mais confiáveis em detrimento dos menos confiáveis.

19.12.3. Configurando Sua Máquina

19.12.3.1. Configuração Básica

Se você somente deseja sincronizar seu relógio quando a máquina for inicializada, você pode usar [ntpdate\(8\)](#). Isto pode ser adequado para algumas máquinas desktop as quais são frequentemente reinicializadas e necessitam somente de sincronizações esparsas, mas a maioria das máquinas deveria executar [ntpd\(8\)](#).

Usando [ntpdate\(8\)](#) na inicialização também é uma boa idéia para máquinas que rodam [ntpd\(8\)](#). O programa [ntpd\(8\)](#) altera o relógio gradualmente, ao passo que o [ntpdate\(8\)](#) ajusta o relógio, qualquer que seja a diferença entre a hora atual do relógio e a hora correta.

Para ativar o [ntpdate\(8\)](#) durante a inicialização, adicione `ntpdate_enable="YES"` ao `/etc/rc.conf`. Você também vai precisar especificar todos os servidores com os quais deseja sincronizar e quaisquer parâmetros a serem passados ao [ntpdate\(8\)](#) em `ntpdate_flags`.

19.12.3.2. Configuração Geral

O NTP é configurado pelo arquivo `/etc/ntp.conf` no formato descrito em [ntp.conf\(5\)](#). Aqui está um exemplo simples:

```
server ntplocal.exemplo.com prefer
server timeserver.example.org
server ntp2a.example.net
```

```
driftfile /var/db/ntp.drift
```

A opção `server` especifica quais servidores devem ser usados, com um servidor por linha. Se um servidor é especificado com o argumento `prefer`, como em `ntplocal.exemplo.com`, este servidor é preferido sobre os demais servidores. Uma resposta de um servidor preferido será descartada se ela diferir muito das respostas dos demais servidores, caso contrário será usada desconsiderando as outras respostas. O argumento `prefer` normalmente é usado para servidores NTP que são reconhecidamente de alta precisão, como os que possuem hardware especial para monitoramento da hora.

A opção `driftfile` especifica qual arquivo é usado para armazenar a compensação de frequência do relógio do sistema. O programa [ntpd\(8\)](#) usa isto para compensar automaticamente as variações naturais do relógio, permitindo que seja mantida uma configuração correta mesmo que seja cortada a conexão de todas as fontes externas de hora durante um período de tempo.

A opção `driftfile` especifica qual arquivo é usado para armazenar informações sobre as respostas recebidas anteriormente dos servidores NTP que você estiver usando. Este

arquivo contém informações internas para o NTP. Ele não deve ser alterado por nenhum outro processo.

19.12.3.3. Controlando acesso ao Seu Servidor

Por padrão, seu servidor NTP será acessível a todos os sistemas na Internet. A opção *restrict* em `/etc/ntp.conf` permite que você controle quais máquinas podem acessar seu servidor.

Se você quer negar acesso a todas as máquinas ao seu servidor NTP, adicione a seguinte linha ao `/etc/ntp.conf`:

```
restrict default ignore
```

Se você quer permitir que somente máquinas em sua rede sincronizem seus relógios com seu servidor, mas assegurar que não lhes seja permitido configurar o servidor ou usada como pares (peers) de sincronismo, adicione

```
restrict 192.168.1.0 mask 255.255.255.0 notrust nomodify notrap
```

onde `192.168.1.0` é um endereço IP em sua rede e `255.255.255.0` é a máscara de subrede de sua rede.

`/etc/ntp.conf` pode conter múltiplas opções *restrict*. Para mais detalhes, veja a subseção *Access Control Support* de [ntp.conf\(5\)](#).

19.12.4. Executando o Servidor NTP

Para assegurar que o servidor NTP é iniciado durante a inicialização, adicione a linha `xntpd_enable="YES"`. Se você deseja passar parâmetros adicionais ao [ntp\(8\)](#), edite o parâmetro `xntpd_flags` no `/etc/rc.conf`.

Para iniciar o servidor sem reinicializar a máquina, rode `ntpd` certificando de especificar quaisquer parâmetros adicionais de `xntpd_flags` em `/etc/rc.conf`. Por exemplo:

```
# ntpd -p /var/run/ntpd.pid
```

Nota: No FreeBSD 5.X, várias opções foram renomeadas no `/etc/rc.conf`. Assim, você vai ter que trocar cada ocorrência de `xntpd` por `ntpd` nas opções acima.

19.12.5. Usando o ntpd com uma Conexão Internet Temporária

O programa [ntpd\(8\)](#) não precisa de uma conexão permanente à Internet para funcionar corretamente. Entretanto, se você tem uma conexão temporária que está configurada para discar por demanda, é uma boa idéia evitar que o tráfego NTP de disparar uma conexão discada ou de manter a conexão ativa. Se você estiver usando o usuário PPP, você pode usar diretrizes *filter* em `/etc/ppp/ppp.conf`. Por exemplo:

```
set filter dial 0 deny udp src eq 123
# Prevent NTP traffic from initiating dial out
set filter dial 1 permit 0 0
set filter alive 0 deny udp src eq 123
# Prevent incoming NTP traffic from keeping the connection open
set filter alive 1 deny udp dst eq 123
# Prevent outgoing NTP traffic from keeping the connection open
set filter alive 2 permit 0/0 0/0
```

Para maiores detalhes, veja a seção *FILTRAGEM DE PACOTES* em [ppp\(8\)](#) e os exemplos em `/usr/share/examples/ppp/`.

Nota: Alguns provedores de acesso Internet bloqueiam portas de números baixos impedindo o NTP de funcionar, uma vez que as respostas nunca alcançam sua máquina.

19.12.6. Mais Informações

Documentação para o servidor NTP podem ser encontradas em `/usr/share/doc/ntp/` em formato HTML.

19.13. NAT - Tradução de Endereço de Rede

Contribuído por Chern Lee.

19.13.1. Visão Geral

O daemon de Tradução de Endereço de Rede do FreeBSD, geralmente conhecido como [natd\(8\)](#) é o daemon que aceita pacotes IP entrantes não modificados, altera a origem para a máquina local e re-injeta estes pacotes de volta ao fluxo de pacotes IP que saem. [natd\(8\)](#) faz isso mudando o endereço IP de origem e a porta, de forma que, quando o dado é recebido de volta, é capaz de determinar sua localização original e repassá-lo ao seu solicitante original.

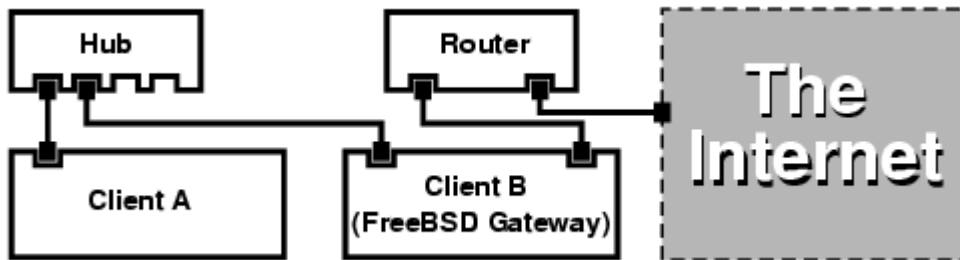
O uso mais comum do NAT é executar o que é mais geralmente conhecido como Compartilhamento de Conexão Internet.

19.13.2. Estrutura

Dada a redução do espaço de endereçamento IP no IPv4, e o elevado número de usuários em linhas de alta velocidade para consumidores, como cabo ou DSL, as pessoas estão progressivamente precisando de uma solução de Compartilhamento de Conexão Internet. A capacidade de conectar vários computadores em linha (online) através de uma conexão e endereço IP torna o [natd\(8\)](#) uma escolha razoável.

Freqüentemente, um usuário tem uma máquina conectada a uma linha DSL ou cabo com um endereço IP e deseja usar este computador conectado para oferecer acesso Internet a diversos mais através de uma rede local.

Para isto, a máquina FreeBSD na Internet precisa atuar como um gateway. Esta máquina precisa ter duas placas de rede --uma para conectar ao roteador Internet, a outra para conectar a uma rede local. Todas as máquinas na rede local são conectadas através de um concentrador ou comutador.



Uma configuração como esta é geralmente usada para compartilhar uma conexão Internet. Uma das máquinas da LAN está conectada à Internet. As demais máquinas acessam a Internet através daquela máquina "gateway".

19.13.3. Configuração

As seguintes opções precisam estar no arquivo de configuração do kernel:

```
options IPFIREWALL
options IPDIVERT
```

Adicionalmente, por escolha, o seguinte também pode ser apropriado:

```
options IPFIREWALL_DEFAULT_TO_ACCEPT
options IPFIREWALL_VERBOSE
```

As seguintes linhas precisam estar no `/etc/rc.conf`:

```
gateway_enable="YES"
firewall_enable="YES"
firewall_type="OPEN"
natd_enable="YES"
natd_interface="fxp0"
natd_flags=""
```

`gateway_enable="YES"` Configura a máquina para atuar como um gateway. Executando `sysctl net.inet.ip.forwarding=1` causaria o mesmo efeito.

`firewall_enable="YES"` Ativa as regras do firewall em `/etc/rc.firewall` durante a inicialização.

`firewall_type="OPEN"` Isto especifica um conjunto predefinido de regras do firewall, permitindo a entrada de qualquer coisa. Veja `/etc/rc.firewall`

para tipos adicionais.

<code>natd_interface="fxp0"</code>	Indica a qual interface repassar pacotes (a interface conectada à Internet).
<code>natd_flags=""</code>	Qualquer das opções de configuração passadas ao natd(8) durante a inicialização.

Tendo as opções anteriores definidas em `/etc/rc.conf` seria executado `natd -interface fxp0` durante a inicialização. Ele também pode ser executado manualmente.

Cada máquina e interface na rede local deveria ter números de endereços IP atribuídos no espaço de rede privado como definido por [RFC 1918](#) e ter como o gateway padrão, o endereço IP interno da máquina **natd**.

Por exemplo, o cliente A e B na rede local possuem endereços IP `192.168.0.2` e `192.168.0.3`, enquanto a interface de rede local da máquina executando o `natd` possui o endereço IP `192.168.0.1`. O gateway padrão dos clientes A e B precisa ser o da máquina com `natd`, `192.168.0.1`. A interface externa, ou Internet, da máquina com `natd` não requer nenhuma modificação especial para o funcionamento do [natd\(8\)](#).

19.13.4. Redirecionamento de Porta

A desvantagem do [natd\(8\)](#) é que os clientes da rede local não são diretamente acessíveis a partir da Internet. Os clientes dentro da rede local podem fazer conexões saindo para o mundo, mas não podem receber conexões vindas de fora. Isto apresenta um problema ao se tentar executar serviços Internet em uma das máquinas cliente da rede local. Uma forma simples de contornar isto é redirecionando certas portas Internet na máquina com **natd** para um cliente da rede local.

Por exemplo, um servidor IRC roda em um cliente A, e um servidor web em um cliente B. Para que isto funcione corretamente, as conexões recebidas nas portas 6667 (IRC) e 80 (web) precisam ser redirecionadas para as respectivas máquinas.

Um `-redirect_port` precisa ser passado ao [natd\(8\)](#) com as opções adequadas. A sintaxe é a seguinte:

```
-redirect_port proto targetIP:targetPORT[-targetPORT]
                [aliasIP:]aliasPORT[-aliasPORT]
                [remoteIP[:remotePORT[-remotePORT]]]
```

No exemplo acima, o argumento deve ser:

```
-redirect_port tcp 192.168.0.2:6667 6667
-redirect_port tcp 192.168.0.3:80 80
```

Isto irá direcionar as portas `tcp` corretas para as máquinas clientes da rede local.

Um argumento *-redirect_port* pode ser usado para indicar faixas de portas além de portas individuais. Por exemplo, *tcp 192.168.0.2:2000-3000 2000-3000* redirecionaria todas as conexões recebidas nas portas 2000 a 3000 para portas 2000 a 3000 no cliente A.

Estas opções podem ser usadas executando diretamente o [natd\(8\)](#), ou inseridas na opção *natd_flags=""* em */etc/rc.conf*.

Para mais opções de configuração, consulte [natd\(8\)](#).

19.13.5. Redirecionamento de Endereço

Redirecionamento de endereço é útil se vários endereços IP estão disponíveis, mesmo que eles precisem estar em uma máquina. Com isto, o [natd\(8\)](#) pode designar cada cliente da rede local com seu próprio endereço externo. O [natd\(8\)](#) então reescreve os pacotes de saída dos clientes da rede local com o endereço IP externo apropriado e redireciona todo o tráfego de entrada para aquele endereço IP de volta ao cliente da rede local específico. Isto também é conhecido como NAT estático. Por exemplo, os endereços IP 128.1.1.1, 128.1.1.2 e 128.1.1.3 pertencem à máquina gateway com **natd**. O 128.1.1.1 pode ser usado como o endereço IP externo da máquina gateway, enquanto 128.1.1.2 e 128.1.1.3 são repassados de volta aos clientes A e B da rede local.

A sintaxe de *-redirect_address* é a seguinte:

```
-redirect_address IPlocal IPPúblico  
IPlocal    O endereço IP interno do cliente da rede local.
```

```
IPPúblico  O endereço IP externo correspondente ao cliente da rede local.
```

No exemplo, este argumento seria escrito:

```
-redirect_address 192.168.0.2 128.1.1.2  
-redirect_address 192.168.0.3 128.1.1.3
```

Como *-redirect_port*, estes argumentos também são inseridos na opção *natd_flags=""* do */etc/rc.conf*. Com redirecionamento de endereço, não há necessidade de redirecionamento de porta, uma vez que todos os dados recebidos em um endereço IP em particular são redirecionados.

Os endereços IP externos na máquina com **natd** precisam ser ativos como secundários na interface externa. Veja como fazer em [rc.conf\(5\)](#).

19.14. O "Super-Servidor" inetd

Contribuído por Chern Lee.

19.14.1. Visão Geral

O [inetd\(8\)](#) é chamado de "Super-Servidor Internet" porque ele gerencia conexões para diversos daemons. O **inetd** atua como um servidor gerenciador para outros daemons. Quando uma conexão é recebida pelo **inetd**, ele determina para qual daemon a conexão é destinada e executa o daemon correspondente e a ele delega o socket. Executar uma instância do **inetd** reduz a carga no sistema de forma geral, comparado a se executar cada daemon individualmente.

Primariamente, o **inetd** é usado para executar outros daemons, mas diversos protocolos simples são tratados diretamente, como **chargen**, **auth** e **daytime**.

Esta seção vai abordar as configurações básicas do **inetd** através de seus parâmetros de linha de comando e seu arquivo de configuração `/etc/inetd.conf`.

19.14.2. Configurações

O **inetd** é inicializado através do sistema `/etc/rc.conf`. A opção `inetd_enable` é ajustada para "NO", por padrão, mas é muitas vezes ativada pelo **sysinstall** com o perfil de segurança médio. Colocando:

```
inetd_enable="YES"
or
```

```
inetd_enable="NO"
no /etc/rc.conf pode ativar ou desativar a execução do inetd na inicialização do sistema.
```

Adicionalmente, parâmetros de linha de comando podem ser passados para o **inetd** através da opção `inetd_flags`.

19.14.3. Opções de Linha de Comando

Sinopse do **inetd**:

```
inetd [-d] [-l] [-w] [-W] [-c máximo] [-C taxa] [-a endereço | nome-do-sistema] [-p nome do arquivo] [-R taxa] [arquivo de configuração]
```

-d

Ativa a depuração.

-l

Ativa o registro das conexões bem sucedidas.

-w

Ativa o encapsulamento TCP para serviços externos (ativado por padrão).

-W

Ativa o encapsulamento do TCP para serviços internos que são embutidos no **inetd** (ativado por padrão).

-c máximo

Especifica o número máximo padrão de invocações simultâneas de cada serviço; o padrão é ilimitado. Pode ser anulado em cada serviço com o parâmetro *max-child*.

-C taxa

Especifica o número máximo padrão de vezes que um serviço pode ser invocado a partir de um único endereço IP em um minuto; o padrão é ilimitado. Pode ser cancelado, por serviço, com o parâmetro *max-conexões-por-ip-por-minuto*.

-R taxa

Especifica o número máximo de vezes que um serviço pode ser invocado em um minuto; o padrão é 256. Uma taxa de 0 permite um número ilimitado de invocações.

-a

Especifica um único endereço IP para ser unido. Alternativamente, um nome de sistema pode ser especificado, e neste caso, o endereço IPv4 ou IPv6 que corresponder àquele nome é usado. Habitualmente um nome de sistema é especificado quando o **inetd** é executado dentro de um [jail\(8\)](#), e neste caso, o nome do sistema corresponde ao ambiente do [jail\(8\)](#).

Quando o é usado o nome do sistema e união com ambos endereços IPv4 e IPv6 é desejada, uma entrada com o tipo apropriado de protocolo para cada união é exigida para cada serviço, no `/etc/inetd.conf`. Por exemplo, um serviço baseado em TCP precisaria de duas entradas, uma usando `tcp4` para o protocolo e outra usando `tcp6`.

-p

Especifica um arquivo alternativo para armazenar o ID do processo.

Estas opções podem ser passadas ao **inetd** usando a opção *inetd_flags* no `/etc/rc.conf`. Por padrão, o *inetd_flags* é ajustado para ```-wW"`, que ativa o encapsulamento TCP para os serviços **inetd** internos e externos. Para usuários novatos,

estes parâmetros geralmente não precisam ser modificados ou mesmo incluídos no `/etc/rc.conf`.

Nota: Um serviço externo é um daemon fora do **inetd**, que é invocado quando uma conexão é recebida para ele. Por outro lado, um serviço interno é um que o **inetd** possui os meios de fornecer por si próprio.

19.14.4. `inetd.conf`

A configuração do **inetd** é controlada através do arquivo `/etc/inetd.conf`.

Quando uma modificação é feita no `/etc/inetd.conf`, o **inetd** pode ser forçado a reler seu arquivo de configuração enviando um sinal HangUP ao processo **inetd**, como mostrado:

Exemplo 19-4. Enviando um Sinal HangUP ao `inetd`

```
# kill -HUP `cat /var/run/inetd.pid`
```

Cada linha do arquivo de configuração especifica um daemon individual. Comentários no arquivo são precedidos por um ``#"`. O formato do `/etc/inetd.conf` é o seguinte:

```
nome-do-serviço
tipo-de-socket
protocolo
{wait|nowait}[/max-de-filhos[/max-de-conexões-por-ip-por-minuto]]
usuário[:grupo][[/classe-de-login]]
programa-servidor
argumentos-do-programa-servidor
```

Uma entrada de exemplo para o daemon **ftpd** usando IPv4:

```
ftp      stream tcp      nowait root    /usr/libexec/ftpd      ftpd -l
nome-do-serviço
```

Este é o nome do serviço do daemon em questão. Ele precisa corresponder ao serviço listado no `/etc/services`. Isto determina qual porta o **inetd** precisa escutar. Se um novo serviço está sendo criado, ele precisa ser colocado primeiro no `/etc/services`.

tipo-de-socket

Pode ser *stream*, *dgram*, *raw*, ou *seqpacket*. *stream* precisa ser usado para daemons TCP baseados em conexão, enquanto *dgram* é usado para daemons usando o protocolo de transporte UDP.

protocolo

Um dos seguintes:

Protocolo	Descrição
tcp, tcp4	TCP IPv4
udp, udp4	UDP IPv4
tcp6	TCP IPv6
udp6	UDP IPv6
tcp46	Ambos TCP IPv4 e v6
udp46	Ambos UDP IPv4 e v6

{wait|nowait}[/max-de-filhos[/max-de-conexões-por-ip-por-minuto]]

wait/nowait indica se o daemon invocado pelo **inetd** é capaz de lidar com seu próprio socket ou não. Sockets do tipo *dgram* precisam usar a opção *wait*, enquanto daemons de sockets tipo *stream*, os quais geralmente são multi-threaded, devem usar a opção *nowait*. *wait* geralmente repassa múltiplos sockets para um único daemon, enquanto *nowait* dispara um daemon filho para cada novo socket.

O número máximo de daemons filho que o **inetd** pode disparar pode ser ajustado usando a opção *max-de-filhos*. Se um limite de dez instâncias de um daemon em particular é necessária, um */10* deve ser colocado após a opção *nowait*.

Além de *max-de-filhos*, outra opção pode ser ativada para limitar o número máximo de conexões a partir de um único lugar para um daemon em particular. O *max-conexões-por-ip-por-minuto* faz justamente isto. Um valor de dez nesta opção limitaria qualquer endereço IP a se conectar a determinado serviço a dez tentativas por minuto. Isto é útil para evitar consumo de recursos intencional ou não e ataques de Negação de Serviço (Denial of service - DoS) para a máquina.

Neste campo, *wait* ou *max-de-filhos* e *max-conexões-por-ip-por-minuto* são opcionais.

Um daemon multi-threaded do tipo *stream* sem qualquer limites *max-de-filhos* ou *max-conexões-por-ip-por-minuto* seria simplesmente: *nowait*.

O mesmo daemon com um limite máximo de dez daemons seria: *nowait/10*

Adicionalmente, a mesma configuração com um limite de vinte conexões por endereço IP por minuto e um limite total máximo de dez daemons filhos seria: *nowait/10/20*.

Estas opções são todas utilizadas pela configuração padrão do daemon **fingerd**, como visto aqui:

```
finger stream tcp      nowait/3/10 nobody /usr/libexec/fingerd
fingerd -s
usuário
```

O usuário é o nome do usuário através do qual o daemon específico deve ser executado. Mais frequentemente, daemons rodam com o usuário `root`. Por propósito de segurança, é comum encontrar alguns servidores rodando como o usuário `daemon`, ou o usuário com menos privilégios, `nobody`.

programa-servidor

O caminho completo do daemon a ser executado quando uma conexão é recebida. Se o daemon é um serviço a ser fornecido internamente pelo **inetd**, então *internal* deve ser utilizado.

argumentos-do-programa-servidor

Este funciona em conjunto com *program-servidor* especificando os argumentos, iniciando com o `argv[0]`, passado ao daemon no momento da invocação. Se **meudaemon -d** é a linha de comando, *meudaemon -d* seria o valor de *argumentos do programa servidor*. Novamente, se o daemon é um serviço interno, use *internal* aqui.

19.14.5. Segurança

Dependendo do perfil de segurança selecionado durante a instalação, muitos daemons do **inetd** podem ser ativados por padrão. Se não há necessidade aparente para um daemon em particular, desative-o! Coloque um `##` na frente do daemon em questão e envie um [sinal de hangup para o inetd](#). Alguns daemons como o **fingerd**, podem ser totalmente indesejáveis, porque eles fornecem muita informação para um atacante.

Alguns daemons não são conscientes com a segurança e possuem longos ou inexistentes tempos de expiração (timeouts) para tentativas de conexão. Isto permite um atacante enviar conexões lentamente para um daemon em particular, saturando os recursos disponíveis. Pode ser uma boa idéia colocar limitações *ip-por-minuto* e *max-filhos* em certos daemons.

Por padrão, o encapsulamento TCP é ativado. Consulte a página de manual [hosts_access\(5\)](#) para mais informações sobre como colocar restrições TCP em vários daemons invocados pelo **inetd**.

19.14.6. Miscelânea

daytime, **time**, **echo**, **discard**, **chargen**, e **auth** são todos serviços fornecidos internamente pelo **inetd**.

O serviço **auth** fornece serviços de identidade (`ident`, `identd`) de rede, e é configurável até certo ponto.

Consulte a página de manual [inetd\(8\)](#) para informações mais aprofundadas.

19.15. PLIP - IP em Linha Paralela

PLIP nos permite usar TCP/IP entre portas paralelas. É útil em máquinas sem placas de rede, ou para instalar em laptops. Nesta seção vamos examinar:

- Criando um cabo paralelo (laplink).
- Conectando dois computadores com PLIP.

19.15.1. Criando um Cabo Paralelo

Você pode comprar um cabo paralelo na maioria das lojas de suprimentos de informática. Se você não puder fazer isso, ou você apenas quer saber como um cabo é feito, a tabela a seguir mostra como fazer um a partir de um cabo paralelo para impressora.

Tabela 19-1. Conectando um Cabo Paralelo para Conexão em Rede

A-name	A-End	B-End	Descr.	Post/Bit
DATA0 -ERROR	2 15	15 2	Data	0/0x01 1/0x08
DATA1 +SLCT	3 13	13 3	Data	0/0x02 1/0x10
DATA2 +PE	4 12	12 4	Data	0/0x04 1/0x20
DATA3 -ACK	5 10	10 5	Strobe	0/0x08 1/0x40
DATA4 BUSY	6 11	11 6	Data	0/0x10 1/0x80
GND	18-25	18-25	GND	-

19.15.2. Configurando o PLIP

Primeiro, você precisa obter um cabo laplink. Depois, certifique-se de que ambos os computadores possuem um kernel com suporte do gerenciador de dispositivos [lpt\(4\)](#):

```
# grep lp /var/run/dmesg.boot
lpt0: <Printer> on ppbus0
lpt0: Interrupt-driven port
```

A porta paralela precisa ser uma porta gerenciada por interrupções. No FreeBSD 4.X, você deve ter uma linha similar à seguinte em seu arquivo de configuração do kernel:


```
device ppc0 at isa? irq 7
```

No FreeBSD 5.X, o arquivo `/boot/device.hints` deve conter as seguintes linhas:

```
hint.ppc.0.at="isa"  
hint.ppc.0.irq="7"
```

Então verifique se o arquivo de configuração do kernel tem uma linha *device plip* ou se o módulo de kernel `plip.ko` está carregado. Em ambos os casos, a interface de rede paralela deve aparecer quando você usar diretamente o comando [ifconfig\(8\)](#). No FreeBSD 4.X, seria assim:

```
# ifconfig lp0  
lp0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
```

e no FreeBSD 5.X:

```
# ifconfig plip0  
plip0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
```

Nota: O nome do dispositivo usado para a interface paralela é diferente entre o FreeBSD 4.X (`lpX`) e o FreeBSD 5.X (`plipX`).

Conecte o cabo laplink na interface paralela em ambos os computadores.

Configure os parâmetros da interface de rede nos dois sítios como `root`. Por exemplo, se você quer conectar o sistema `host1` executando FreeBSD 4.X com `host2` executando FreeBSD 5.X:

```
host1 <-----> host2  
IP Address 10.0.0.1 10.0.0.2
```

Configure a interface no `host1` com o comando:

```
# ifconfig lp0 10.0.0.1 10.0.0.2
```

Configure a interface no `host2` fazendo:

```
# ifconfig plip0 10.0.0.2 10.0.0.1
```

Você agora deve ter uma conexão funcionando. Por favor leia as páginas de manual [lp\(4\)](#) e [lpt\(4\)](#) para mais detalhes.

Você deve também adicionar ambas os sistemas ao `/etc/hosts`:

```
127.0.0.1 localhost.my.domain localhost  
10.0.0.1 host1.my.domain host1  
10.0.0.2 host2.my.domain
```

Para confirmar que a conexão funciona, vá a cada sistema e faça um ping para o outro. Por exemplo, no `host1`:

```

# ifconfig lp0
lp0: flags=8851<UP,POINTOPOINT,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet 10.0.0.1 --> 10.0.0.2 netmask 0xff000000
# netstat -r
Routing tables

Internet:
Destination Gateway    Flags    Refs      Use      Netif  Expire
host2        host1      UH        0          0        lp0
# ping -c 4 host2
PING host2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: icmp_seq=0 ttl=255 time=2.774 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=255 time=2.530 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=255 time=2.556 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=255 time=2.714 ms

--- host2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.530/2.643/2.774/0.103 ms

```

19.16. IPv6

Originalmente Escrito por Aaron Kaplan. Reestruturado e Adicionado por Tom Rhodes.

IPv6 (também conhecido como IPng "IP next generation") é a nova versão do bem conhecido protocolo IP (também conhecido como IPv4). Como outros sistemas *BSD atuais, o FreeBSD inclui a implementação de referência KAME. Então o seu sistema FreeBSD vem com tudo o que você vai precisar para experiências com IPv6. Esta seção enfoca na configuração e ativação do IPv6.

No início dos anos 90, as pessoas tomaram conhecimento do rápido esgotamento do espaço de endereçamento do IPv4. Dada a taxa de expansão da Internet, haviam duas preocupações principais:

- Esgotamento de endereços. Hoje isto não é uma preocupação tão grande, uma vez que espaços de endereços privados (10.0.0.0/8, 192.168.0.0/24, etc.) e a Tradução de Endereços de Rede (Network Address Translation - NAT) estão sendo empregados.
- Entrada nas tabelas de roteadores estavam ficando grandes demais. Hoje isto ainda é uma preocupação.

IPv6 trata destas e muitas outras questões:

- Espaço de endereçamento de 128 bits. Em outras palavras, existem teóricos 340,282,366,920,938,463,463,374,607,431,768,211,456 endereços disponíveis. Isto significa que são aproximadamente $6.67 * 10^{27}$ endereços IPv6 por metro quadrado em nosso planeta.
- Roteadores armazenarão somente endereços agregados de redes em suas tabelas de roteamento, reduzindo assim, o espaço médio de uma tabela de roteamento para 8192 entradas.

Também existem muitas outras características úteis do IPv6 como:

- Autoconfiguração de endereço (RFC2462)
- Endereços anycast ("um dentre muitos")
- Endereços multicast obrigatórios
- IPsec (IP security)
- Estrutura simplificada de cabeçalho
- IP Móvel
- Mecanismos de transição IPv4-para-IPv6

Para maiores informações veja:

- IPv6 overview at Sun.com
- IPv6.org
- KAME.net
- 6bone.net

19.16.1. Fundamentos em Endereços IPv6

Existem diferentes tipos de endereços IPv6: Unicast, Anycast e Multicast.

Endereços Unicast são os endereços conhecidos. Um pacote enviado a um endereço unicast chega exatamente à interface pertencente ao endereço.

Endereços anycast são sintaticamente indistinguíveis de endereços unicast, mas eles endereçam um grupo de interfaces. Um pacote destinado para um endereço anycast vai chegar na interface mais próxima (em métrica de roteador). Endereços anycast somente podem ser usados por roteadores.

Endereços multicast identificam um grupo de interfaces. Um pacote destinado a um endereço multicast vai chegar em todas as interfaces pertencentes ao grupo multicast.

Nota: O endereço de difusão (broadcast) IPv4 (normalmente xxx.xxx.xxx.255) é expresso por endereços multicast em IPv6.

Endereços IPv6 Reservados:

ipv6-address	prefixlength(Bits)	description	Notes
::	128 Bits	não especificado	cf. 0.0.0.0 em endereço IPv4
::1	128 Bits	endereço loopback	cf. 127.0.0.1 em IPv4
::00:xx:xx:xx:xx	96 Bits	IPv4 embutido	Os 32 bits mais baixos são os endereços IPv4. Também chamado ``Endereço IPv4 compatível com IPv6``
::ff:xx:xx:xx:xx	96 Bits	IPv4 mapeado	Os 32 bits mais baixos são os IPv6 address endereços IPv4. Para sistemas que não suportam IPv6

fe80:: - feb::	10 Bits	link-local	cf. endereço loopback
em			
	IPv4		
fec0:: - fef::	10 Bits	locais ao sítio	
ff::	8 Bits	multicast	
001 (base 2)	3 Bits	unicast globais	Todos os endereços
unicast globais			são atribuídos deste
			conjunto. Os 3 primeiros Bits
			são ``001''.

19.16.2. Lendo Endereços IPv6

A forma canônica é representada como: x:x:x:x:x:x:x:x, cada ``x" sendo um valor hexadecimal de 16 Bits. Por exemplo FEBC:A574:382B:23C1:AA49:4592:4EFE:9982

Freqüentemente um endereço terá longas subséries compostas por zeros, então cada subsérie pode ser abreviada por ``:". Por exemplo fe80::1 corresponde à forma canônica fe80:0000:0000:0000:0000:0000:0000:0001

Uma terceira forma é escrever a última parte de 32 Bits na forma conhecida (decimal), estilo IPv4, com pontos ``." como separadores. Por exemplo, 2002::10.0.0.1 corresponde à representação canônica (hexadecimal)

2002:0000:0000:0000:0000:0000:0a00:0001 que por sua vez equivale a escrever 2002::a00:1

Agora, os leitores devem ser capazes de compreender o seguinte:

```
# ifconfig
r10: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu
1500
    inet 10.0.0.10 netmask 0xfffff00 broadcast 10.0.0.255
    inet6 fe80::200:21ff:fe03:8e1%r10 prefixlen 64 scopeid 0x1
    ether 00:00:21:03:08:e1
    media: Ethernet autoselect (100baseTX )
    status: active
```

fe80::200:21ff:fe03:8e1%r10 é um endereço autoconfigurado tipo enlace-local. Ele inclui o MAC Ethernet misturado como parte da autoconfiguração.

Para maiores informações sobre a estrutura dos endereços IPv6, veja a RFC2373.

19.16.3. Conectando-se

Atualmente existem quatro formas de se conectar a outros sistemas e redes IPv6:

- Associar-se ao 6bone experimental
- Obtendo uma rede IPv6 de seu provedor de conectividade. Procure seu provedor Internet para instruções.
- Tunelamento via 6-para-4
- Usar o port freenet6 se estiver em uma conexão discada.

Aqui vamos falar sobre como se conectar ao 6bone uma vez que atualmente parece ser a forma mais popular.

Primeiro, veja a página do 6bone e encontre a conexão para o 6bone mais próxima de você. Escreva para a pessoa responsável e com um pouco de sorte você receberá instruções sobre como configurar sua conexão. Geralmente isto envolve preparar um túnel GRE (gif).

Aqui está um exemplo típico de como preparar um túnel [gif\(4\)](#):

```
# ifconfig gif0 create
# ifconfig gif0
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
# ifconfig gif0 tunnel MEU_END_IPv4 END_IPv4_DELE
# ifconfig gif0 inet6 alias
MEU_END_IPv6_ATRIBUIDO_COMO_PONTO_FINAL_DO_TÚNEL
```

Substitua as palavras em maiúsculas pela informação recebida do nó 6bone de nível superior.

Isto estabelece o túnel. Verifique se o túnel está funcionando usando [ping6\(8\)](#)
ff02::1%gif0. Você deve receber duas respostas do ping.

Nota: Em caso de você estar intrigado pelo endereço ff02::1%gif0, isto é um endereço multicast. %gif0 indica que o endereço multicast na interface de rede gif0 deve ser usado. Uma vez que nós usamos ping para um endereço multicast, o outro ponto final do túnel deve responder também).

Agora, configurar a rota para sua conexão ao 6bone deve ser bem clara:

```
# route add -inet6 default -interface gif0
# ping6 -n MINHA_CONEXÃO
# traceroute6 www.jp.FreeBSD.org
(3ffe:505:2008:1:2a0:24ff:fe57:e561) from 3ffe:8060:100::40:2, 30 hops
max, 12 byte packets
 1 atnet-meta6 14.147 ms 15.499 ms 24.319 ms
 2 6bone-gw2-ATNET-NT.ipv6.tilab.com 103.408 ms 95.072 ms *
 3 3ffe:1831:0:ffff::4 138.645 ms 134.437 ms 144.257 ms
 4 3ffe:1810:0:6:290:27ff:fe79:7677 282.975 ms 278.666 ms 292.811
ms
 5 3ffe:1800:0:ff00::4 400.131 ms 396.324 ms 394.769 ms
 6 3ffe:1800:0:3:290:27ff:fe14:cdee 394.712 ms 397.19 ms 394.102
ms
```

Esta saída vai ser diferente de máquina para máquina. Agora você deve ser capaz de alcançar o sítio IPv6 www.kame.net e ver a tartaruga dançante -- se você tiver um navegador pronto para IPv6, como o [www/mozilla](http://www.mozilla).

19.16.4. DNS no Mundo IPv6

Existem duas novas formas de registros DNS para IPv6:

- Registros AAAA,

- Registros A6

Usar registros AAAA é bem direto. Atribua seu nome de sistema ao novo endereço IPv6 que você recém obteve adicionando:

```
MEUNOMEDESISTEMA      AAAA      MEUENDIPv6
```

ao seu arquivo DNS de zona primária. No caso de você não servir suas próprias zonas DNS peça ao seu provedor DNS. Versões atuais do **bind** (versões 8.3 e 9) suportam registros AAAA.

Capítulo 20. Correio eletrônico

Índice

- 20.1. [Sinótese](#)
- 20.2. [Usando o correio eletrônico](#)
- 20.3. [Configuração do **sendmail**](#)
- 20.4. [Trocando seu agente de transferência de mensagens](#)
- 20.5. [Resolução de problemas](#)
- 20.6. [Tópicos avançados](#)
- 20.7. [SMTP com UUCP](#)
- 20.8. [Usando o correio com uma conexão discada](#)
- 20.9. [Autenticação SMTP](#)

Trabalho original por Bill Lloyd. Reescrito por Jim Mock.

20.1. Sinótese

“Correio eletrônico”, mais conhecido como email, é uma das formas mais usadas de comunicação atualmente. Este capítulo descreve uma introdução básica da execução de um servidor de correio eletrônico no FreeBSD. No entanto, esta não é uma referência completa e de fato várias considerações importantes são omitidas. Para uma abrangência mais completa sobre esse assunto, leia a lista de referências de excelentes livros listados em [Apêndice B](#).

Após ler este capítulo você saberá:

- Quais componentes de software estão envolvidos no envio e recebimento do correio eletrônico.
- Onde a configuração básica do **sendmail** está no localizada no FreeBSD.
- Como bloquear *spammer* de usar ilegalmente seu servidor de correio como relay.
- Como instalar e configurar um servidor de correio alternativo em seu sistema, substituindo o **sendmail**.
- Como resolver problemas comuns no seu servidor de correio eletrônico.
- Como usar SMTP com UUCP.
- Como usar o correio com uma conexão dialup.

- Como configurar uma autenticação SMTP para melhorar a segurança.

Antes de ler este capítulo você deve ter:

- Configurado corretamente sua conexão de rede ([Capítulo 19](#)).
 - Configurado corretamente as informações do DNS para seu host de correio ([Capítulo 19](#)).
 - Saber como instalar software adicional de terceiros ([Capítulo 4](#)).
-

20.2. Usando o correio eletrônico

Existe cinco principais partes envolvidas na troca de email. Elas são: [O programa do usuário](#), [O daemon do servidor](#), [DNS](#), [o POP ou o daemon do IMAP](#), e é claro, [o próprio servidor de email](#).

20.2.1. O programa de usuário

Isto inclui programas de linha de comando como **mutt**, **pine**, **elm**, e **mail**, e programas GUI como **balsa**, **xfmail** para falar de alguns, e muito mais ``sofisticados" como o navegador WWW. Estes programas simplesmente passam as transações de email para o ["servidor de email"](#) local, chamando qualquer um dos [daemons do servidor](#) disponível ou entregando através de TCP.

20.2.2. Daemon do servidor de email

Geralmente é o **sendmail** (por padrão no FreeBSD) ou um dos outros daemons de servidores de emails como **qmail**, **postfix**, ou **exim**. Existem outros porém estes são os mais usados.

O servidor daemon normalmente tem duas funções--ele olha depois de receber um mail que chegou e entrega a mensagem. Ele não permite que você conecte a ele via POP ou IMAP para ler sua mensagem. Você precisa de um [daemon](#) adicional para isso.

Esteja ciente de que algumas versões antigas do **sendmail** tem sérios problemas de segurança, entretanto se você manter rodando uma versão sempre atual, você não deverá ter problemas. É uma boa idéia sempre manter atualizado qualquer aplicativo que você rode.

20.2.3. Email e DNS

O sistema de nome de domínios (DNS) e seu daemon `named` tem um grande papel na entrega de email. Em ordem para entregar a mensagem do seu site para outro, o servidor daemon irá consultar no DNS para determinar o host que irá receber o destino da mensagem.

Isso funciona da mesma forma quando é enviado uma mensagem para você. O DNS contém o banco de dados com mapeamento de nomes para o endereço IP, e o nome do host para o servidor de email. O endereço IP é especificado no registro A. O registro MX (Mail eXchanger) especifica o servidor de email que irá receber a mensagem para você. Se você não tiver um registro MX para seu hostname, sua mensagem será entregue diretamente para seu host.

20.2.4. Recebendo mensagem

O recebimento de mensagens para o seu domínio é feito pelo servidor de correio. Ele irá coletar a mensagem enviada para você e armazená-la para leitura ou pega-la. Para puxar a mensagem armazenada no servidor, você irá precisar conectar no servidor de email. Isto é feito usando POP ou IMAP. Se você quer ler mensagens diretamente no servidor de email, então o POP ou IMAP não é necessário.

Se você quer rodar um servidor POP ou IMAP, há duas coisas que você deve fazer:

1. Pegue o daemon do POP ou IMAP da [coleção de ports](#) e instale em seu sistema.
2. Modifique o `/etc/inetd.conf` para carregar o servidor POP ou IMAP.

20.2.5. O servidor de correio

O servidor de correio é o nome dado ao servidor que é responsável pela entrega e recebimento de mensagens do seu host, e possivelmente de sua rede.

20.3. Configuração do sendmail

Contribuído por Christopher Shumway.

[sendmail\(8\)](#) é o agente de transferência de mensagens (MTA) padrão no FreeBSD. A função do **sendmail** é aceitar mensagens a partir de um agente de usuário de mensagens (MUA) e entregá-la ao mailer apropriado como definido pelo arquivo de configuração. **sendmail** pode também aceitar conexões de rede e entregar as mensagens em caixas de mensagens locais ou entregá-las para outro programa.

O **sendmail** usa os seguintes arquivos de configuração:

Nome do arquivo	Função
<code>/etc/mail/access</code>	sendmail acessa o arquivo da base de dados
<code>/etc/mail/aliases</code>	Apelidos de caixa de mensagem
<code>/etc/mail/local-host-names</code>	Lista dos hosts que o sendmail aceita mensagens

Nome do arquivo	Função
/etc/mail/mailer.conf	Arquivo de configuração do Mailer
/etc/mail/mailertable	Tabela de entrega do mailer
/etc/mail/sendmail.cf	O arquivo de configuração principal do sendmail
/etc/mail/virtusertable	Usuários virtuais e tabelas de domínios

20.3.1. /etc/mail/access

A banco de dados de acesso que define quais host(s) ou endereço IP tem acesso local ao servidor de email e qual tipo de acesso eles tem. Os hosts podem ser listados com *OK*, *REJECT*, *RELAY* ou simplesmente passar para o **sendmail** segurar erros de rotina dado pelo encarregado do envio de mensagens. Os hosts que são listados como *OK*, são por padrão permitidos enviar mensagens para este host assim como enviar mensagens através da maquina local para destinatários finais. Os hosts que são listados com *REJECT* são rejeitados por qualquer conexão de email no servidor. Os hosts que tem a opção *RELAY* para seu hostname, são permitidos para enviar mensagens para qualquer destino através desse servidor de correio.

Exemplo 20-1. Configurando o banco de dados de acesso do sendmail

```
cyberspammer.com      550 Nós não aceitamos mensagens de spammers
FREE.STEALTH.MAILER@  550 Nós não aceitamos mensagens de
spammers
another.source.of.spam REJECT
okay.cyberspammer.com OK
128.32                RELAY
```

Neste exemplo nós temos cinco entradas. Remetentes de mensagens que são encontrados do lado esquerdo da tabela são afetados pela ação do lado direito da tabela. Os primeiros dois exemplos dão um erro de código para a rotina do **sendmail** A mensagem é emitida para o host remoto quando é encontrado o email do lado esquerdo da tabela. A próxima entrada, rejeita mensagens vindo de um host específico na internet, *another.source.of.spam*. A próxima entrada aceita conexões de mensagens vindo do host *okay.cyberspammer.com*, que é mais exata do que a linha acima *cyberspammer.com*. Definições mais específicas sobreescrevem definições menos exatas. A última entrada permite *relaying* de correio eletrônico vindo dos hosts com endereço IP que começam com *128.32*. Estes hosts estão aptos a enviar mensagens através desse servidor de correio que são destinados para outros servidores de correio.

Quando este arquivo é atualizado, você precisa executar o comando `make` no diretório `/etc/mail/` para atualizar a base de dados.

20.3.2. /etc/mail/aliases

A base de dados de apelidos contém a lista de caixas de mensagens virtuais que é expandido a outros usuário(s), arquivos, programas ou outros apelidos. Aqui estão alguns exemplos que podem ser usado no /etc/mail/aliases:

Exemplo 20-2. Apelidos de email

```
root: localuser
ftp-bugs: joe,eric,paul
bit.bucket: /dev/null
procmail: "|/usr/local/bin/procmail"
```

O formato do arquivo é simples; O nome da caixa de mensagens(e-mail) é indicado do lado esquerdo dos dois pontos e depois do lado direito é o destino da mensagens para onde serão direcionadas. O primeiro exemplo simplesmente direciona a caixa de correio de mensagens que chegam para o usuário `root` para a caixa de correio do usuário `localuser`, que é olhada novamente na base de dados de apelidos. Se nada for encontrado, então a mensagem é entregue, ao usuário local `localuser`. O próximo exemplo mostra uma lista de emails. Mensagens para a caixa de correio `ftp-bugs` é direcionada para três caixas de mensagem local, `joe`, `eric`, e `paul`. Note que uma caixa de mensagens remota pode ser especificada como *usuário@exemplo.com.br*. O próximo exemplo mostra escrevendo uma mensagem para um arquivo, neste caso `/dev/null`. O último exemplo mostra enviando mensagens para um programa, neste caso a mensagem é escrita para uma entrada padrão do `/usr/local/bin/procmail` através do UNIX® pipe.

Quando este arquivo é atualizado, você precisa executar o comando `make` no `/etc/mail/` para atualizar a base de dados.

20.3.3. /etc/mail/local-host-names

Esta é a lista dos hostnames do [sendmail\(8\)](#) que são aceitos como nomes de host locais. Coloque qualquer domínio ou host do **sendmail** que está poderá receber mensagens. Por exemplo, se este servidor aceitar mensagens para este domínio `exemplo.com` e o host `mail.exemplo.com`, o `local-host-names` pode ter algo como isto:

```
example.com
mail.example.com
```

Quando este arquivo é atualizado, O [sendmail\(8\)](#) precisa ser reiniciado para ativar as mudanças.

20.3.4. /etc/mail/sendmail.cf

O **sendmail** tem como arquivo de configuração principal, o `sendmail.cf` que controla o comportamento total do **sendmail**, incluindo tudo de reescrita de endereços de mensagens à impressão de mensagens rejeição aos servidores remotos de correio. Naturalmente, com um papel tão diverso, este arquivo de configuração é completamente

complexo e os detalhes fogem do escopo desta seção. Felizmente, este arquivo de configuração raramente precisa ser modificado por servidores de correio padrão.

O arquivo de configuração principal do **sendmail** pode ser construído a partir dos macros [m4\(1\)](#) que define as características e comportamento do **sendmail**. Por favor veja o `/usr/src/contrib/sendmail/cf/README` para maiores detalhes.

Quando mudanças neste arquivo são feitas, o **sendmail** precisa ser reiniciado para que as mudanças tenham efeito.

20.3.5. `/etc/mail/virtusertable`

O `virtusertable` mapeia os endereços de correio para domínios virtuais e caixas de mensagens para caixas de mensagens reais. Estas caixas de mensagens podem ser locais, remotas, apelidos definidos no `/etc/mail/aliases` ou arquivos.

Exemplo 20-3. Exemplo de mapeamento de domínio virtual de mensagens

```
root@exemplo.com          root
postmaster@exemplo.com    postmaster@noc.exemplo.net
@exemplo.com              joe
```

No exemplo acima, nós temos um mapeamento para o domínio `exemplo.com`. O arquivos é processado de cima pra baixo, ordenado na busca da primeira constatação da entrada. O primeiro item mapeia `root@exemplo.com` para a caixa de mensagens local do usuário `root`. A próxima entrada mapeia `postmaster@exemplo.com` para a caixa de mensagens do usuário `postmaster` do host `noc.exemplo.net`. Finalmente, se nada de `exemplo.com` for encontrado, ele irá usar o último mapeamento, que ao ser enviada uma mensagem para qualquer pessoa para o domínio `exemplo.com` ela será entregue a um determinado usuário. Neste caso ele irá fazer o mapeamento para a caixa de mensagens local do usuário `joe`.

20.4. Trocando seu agente de transferência de mensagens

Escrito por Andrew Boothman. Informações obtidas a partir do email escrito por Gregory Neil Shapiro.

Como mencionado, o FreeBSD vem com o **sendmail** já instalado como seu MTA (Agente de transferência de mensagens). Consequentemente por padrão sua carga está na saída e entrada de mensagens.

Entretanto, por uma série de razões, alguns administradores de sistema querem trocar o seu sistema de MTA. Estas razões variam simplesmente em querer testar outro MTA até a necessidade de funções específicas ou confia em outro pacote responsável pelo envio

de mensagens. Felizmente, por qualquer razão, o FreeBSD faz com que essa mudança seja simples de ser realizada.

20.4.1. Instalando um novo MTA

Você tem uma variedade de escolhas de MTAs disponíveis. Um bom ponto de partida é a [Coleção de Ports do FreeBSD](#) onde você poderá encontrar muitos deles. É claro que você está livre para usar qualquer MTA que você quiser em qualquer localidade, assim como rodá-lo no FreeBSD.

Comece instalando seu novo MTA. Uma vez que ele está instalado, ele lhe dará a oportunidade de decidir se ele cumprirá suas necessidades, e também lhe dá a oportunidade de configurar seu novo aplicativo antes de desabilitar o **sendmail**. Ao se fazer isso, você deve certificar-se que a instalação do novo aplicativo não tentará sobrescrever os binários do sistema como `/usr/bin/sendmail`. Se não, seu novo aplicativo de mensagens será posto fazendo o serviço antes que você tenha configurado-o.

Por favor consulte a documentação do seu MTA para informações sobre como configurar seu aplicativo que você escolheu.

20.4.2. Desabilitando o **sendmail**

O processo de inicialização do **sendmail** mudou significativamente entre o 4.5-RELEASE e 4.6-RELEASE. Portanto, o procedimento usado para desabilitá-lo é subitamente diferente.

20.4.2.1. FreeBSD 4.5-STABLE antes de 2002/4/4 e antes (incluindo o 4.5-RELEASE e antes disso)

Entre com a opção:

```
sendmail_enable="NO"
```

no `/etc/rc.conf`. Isto irá desabilitar o **sendmail** como serviço de chegada de mensagens, mas se o `/etc/mail/mailler.conf` (veja abaixo) não for alterado, o **sendmail** ainda continuará sendo usado para enviar mensagens.

20.4.2.2. FreeBSD 4.5-STABLE depois de 2002/4/4 (Incluindo o 4.6-RELEASE e mais recentes)

Para desabilitar completamente o **sendmail** você deve usar

```
sendmail_enable="NONE"
```

```
no /etc/rc.conf.
```

Atenção Se você desabilitar o **sendmail** como serviço de saída de mensagens, é importante que você o substitua com um sistema totalmente alternativo no serviço de

entrega de mensagens. Se você escolher não fazê-lo, funções do sistema como [periodic\(8\)](#) serão incapaz de entregar seus resultados por e-mail que eles esperariam normalmente. Várias partes do seu sistema pode esperar que tenham sistema **sendmail**-compatível funcional. Se as aplicações continuarem a usar os binários do **sendmail** para tentar enviar mensagens após desabilitar, a mensagem ficará inativa na fila do **sendmail**, e nunca será entregue.

Se você quer desabilitar o serviço de chegada do **sendmail** você deve configurar,

```
sendmail_enable="NO"
```

no `/etc/rc.conf`. Maiores informações sobre as opções de inicialização do **sendmail** está disponível na página de manual [rc.sendmail\(8\)](#).

20.4.3. Executando seu novo MTA na inicialização do sistema

Você pode ter dois métodos para escolher de como rodar seu novo MTA na inicialização do sistema, dependendo da versão do FreeBSD que você está rodando.

20.4.3.1. FreeBSD 4.5-STABLE antes de 2002/4/11 (Incluindo o 4.5-RELEASE e mais recentes)

Adicione um script em `/usr/local/etc/rc.d/` com extensão `.sh` e executável pelo usuário `root`. Este script deve ser aceito os parâmetros `start` e `stop`. Na hora da inicialização, os scripts do sistema irão executar o comando

```
/usr/local/etc/rc.d/supermailer.sh start
```

Que você também pode manualmente inicializar o servidor. Na hora de desligar, os scripts do sistema irão usar a opção `stop`, executando o comando

```
/usr/local/etc/rc.d/supermailer.sh stop
```

Que você também pode manualmente parar o servidor enquanto o sistema está rodando.

20.4.3.2. FreeBSD 4.5-STABLE depois de 2002/4/11 (Incluindo a 4.6-RELEASE e posteriores)

Com versões mais recentes do FreeBSD, você pode usar o método acima ou você pode configurar

```
mta_start_script="nomedoarquivo"
```

no `/etc/rc.conf`, onde *nomedoarquivo* é o nome de algum script que você queira executar na inicialização do sistema para inicializar set MTA.

20.4.4. Substituindo o sendmail como encarregado do envio da correspondência do sistema

O programa **sendmail** é tão unipresente como aplicativo padrão na maioria dos sistemas UNIX® que alguns aplicativos assumem que ele já esteja instalado e configurado. Por esta razão, muitos dos MTAs alternativos fornece sua própria implementação de compatibilidade da interface de linha de comando do **sendmail**; isto facilita "colocá-los" como substitutos do **sendmail**.

Consequentemente, se você está usando um servidor de correio alternativo, você precisará certificar-se de que o aplicativo que tenta executar executará binários padrão do **sendmail** tais como `/usr/bin/sendmail`, executa realmente seu servidor de correio escolhido preferivelmente. Felizmente, o FreeBSD fornece um sistema chamado [mailwrapper\(8\)](#) que faz este trabalho pra você.

Quando o **sendmail** está operando como instalado, você irá encontrar algo como abaixo no `/etc/mail/mailer.conf`:

```
sendmail      /usr/libexec/sendmail/sendmail
send-mail     /usr/libexec/sendmail/sendmail
mailq        /usr/libexec/sendmail/sendmail
newaliases   /usr/libexec/sendmail/sendmail
hoststat     /usr/libexec/sendmail/sendmail
purgestat    /usr/libexec/sendmail/sendmail
```

Isto significa que quando alguns desses comandos (tais como o próprio `sendmail`) estão funcionando, o sistema invoca realmente uma cópia do `sendmail` nomeado `mailwrapper`, que checa o `mailer.conf` e executa `/usr/libexec/sendmail/sendmail` instead. Este sistema faz com que seja fácil de mudar quais binários são atualmente executados quando estas funções padrão do `sendmail` são invocadas.

Consequentemente se você quisesse que `/usr/local/supermailer/bin/sendmail-compat` seja executado em vez do **sendmail**, você pode mudar o `/etc/mail/mailer.conf` para ler:

```
sendmail      /usr/local/supermailer/bin/sendmail-compat
send-mail     /usr/local/supermailer/bin/sendmail-compat
mailq        /usr/local/supermailer/bin/mailq-compat
newaliases   /usr/local/supermailer/bin/newaliases-compat
hoststat     /usr/local/supermailer/bin/hoststat-compat
purgestat    /usr/local/supermailer/bin/purgestat-compat
```

20.4.5. Finalizando

Uma vez que você tenha tudo configurado como você quer, você deve matar o processo do **sendmail** que você não tem mais necessidade e iniciar o processo do seu novo MTA, ou simplesmente reiniciar. Reiniciar o sistema irá lhe dar a oportunidade de assegurar-se que você configurou corretamente seu sistema para iniciar automaticamente seu novo MTA na inicialização do sistema.

20.5. Resolução de problemas

20.5.1. [Porque eu tenho que usar o FQDN para hosts no meu site?](#)

20.5.2. [O **sendmail** diz ``mail loops back to myself''](#)

20.5.3. [Como eu posso rodar um servidor de correio em um host com conexão discada?](#)

20.5.4. [Por que eu recebo o erro ``Relaying Denied'' quando envio mensagens a partir de outros hosts?](#)

20.5.1. Porque eu tenho que usar o FQDN para hosts no meu site?

Você provavelmente encontrará que o host está em um domínio diferente; por exemplo se você tiver em `foo.bar.edu` e você deseja alcançar um host chamado `mumble` no domínio `bar.edu`, você terá que consultar-lhe pelo nome do domínio inteiro, `mumble.bar.edu`, em vez de somente `mumble`.

Tradicionalmente, isto foi permitido pelo resolvidores de nome do BSD. Entretanto a versão atual do **BIND** que vem com o FreeBSD não fornece mais abreviaturas para nomes de domínios não totalmente qualificados com excessão do domínio que você está dentro. Então o host `mumble` não qualificado deve ser encontrado como `mumble.foo.bar.edu`, ou irá ser procurado no domínio raiz.

Isto é diferente do comportamento anterior, onde a busca continuou através do `mumble.bar.edu`, e `mumble.edu`. Veja na RFC 1535 porque isso foi considerado uma má prática, ou mesmo um furo de segurança.

Como uma boa solução, você pode colocar a linha:

```
search foo.bar.edu bar.edu
```

em vez do precedente:

```
domain foo.bar.edu
```

no seu `/etc/resolv.conf`. No entanto, certifique-se que a ordem de busca não vai além do ``limite entre a administração pública e local'', como a RFC 1535 chama isto.

20.5.2. O **sendmail** diz ``mail loops back to myself''

Esta questão é respondida no FAQ do **sendmail** como segue:

Eu estou tendo mensagens de ``erros de configuração local'' como:

```
553 relay.domain.net config error: mail loops back to myself
554 <user@domain.net>... Local configuration error
```

Como eu resolvo este problema?

Você pediu que o correio do domínio (ex., `domain.net`) para ser enviado para um host específico (neste caso, `relay.domain.net`) usando um registro MX, mas a máquina de relay não se reconhece como `domain.net`. Adicione `domain.net` ao `/etc/mail/local-host-names` (se você estiver usando `FEATURE(use_cw_file)`) ou adicione ``Cw `domain.net`'' ao `/etc/mail/sendmail.cf`.

O FAQ do **sendmail** pode ser encontrado em <http://www.sendmail.org/faq/> e é recomendado que você o leia caso queira fazer algum ``ajuste" da configuração do seu correio.

20.5.3. Como eu posso rodar um servidor de correio em um host com conexão discada?

Você quer conectar uma máquina em uma LAN à internet. A máquina FreeBSD será o gateway de correio para a LAN. A conexão PPP não é dedicada.

Há pelo menos duas maneiras de fazer isso. Uma delas é usar o UUCP.

A outra maneira é contratar um servidor internet conectado o tempo todo para fornecer o serviço de MX secundário para seu domínio. Por exemplo, se o domínio da sua empresa é `exemplo.com` e seu provedor de serviços internet configurou `exemplo.net` para fornecer serviço de MX secundário para seu domínio:

```
exemplo.com.      MX      10      exemplo.com.
                  MX      20      exemplo.net.
```

Somente um host deve ser específico como recepiante final (adicione `Cw exemplo.com` em `/etc/mail/sendmail.cf` no `exemplo.com`).

Quando o envio do `sendmail` está tentando entregar a mensagem ele irá tentar conectar pra você (`exemplo.com`) sobre o link do modem. Isto na maioria das vezes provavelmente irá dar tempo excedido porque você está conectado. O programa **sendmail** irá automaticamente entregá-lo ao site MX secundário, ex., seu provedor internet (`exemplo.net`). O site MX secundário irá periodicamente então tentar conectar ao seu host e entregar as mensagens ao host MX primário (`exemplo.com`).

Você pode querer usar algo assim para um script de login:

```
#!/bin/sh
# Me coloque em /usr/local/bin/pppmyisp
( sleep 60 ; /usr/sbin/sendmail -q ) &
/usr/sbin/ppp -direct pppmyisp
```

Se você criar um script de login separado para um usuário você poderia usar o comando `sendmail -qRexemplo.com` preferivelmente no script acima. Isto irá forçar todas as mensagens em sua fila para `exemplo.com` serem processadas imediatamente.

Um refinamento adicional da situação segue abaixo:

Mensagem retirado da [lista de discussão de provedores de acesso baseados em FreeBSD](#).

```
> Nós fornecemos a um cliente um MX secundário. O cliente conecta ao
> nosso serviço diversas vezes ao dia automaticamente para puxar as
mensagens
> do seu MX primário (Nós não chamamos o seu site quando uma mensagem
para seu domínio
> chegou). Nosso sendmail envia para a fila de mensagens a cada 30
minutos. Neste
```



```
> momento ele tem que permanecer 30 minutos conectado para certificar-  
se que todas as mensagens  
> estão indo para o MX primário.  
>  
> Há algum comando que inicie o sendmail para enviar todas as  
mensagens  
> agora? É claro que o usuário não tem privilégios de root em nossa  
máquina.
```

Na seção ``privacy flags'` do `sendmail.cf`, há a definição `Opgoaway,restrictqrun`

Retire o `restrictqrun` para permitir que usuários que não sejam root iniciem o processamento de fila. Você pode querer também rearrumar os MXs. Nós somos o 1o MX para nossos clientes como estes, e nós definimos:

```
# Se nós formos o melhor MX para o host, tente uma vez diretamente  
gerar  
# um erro de configuração local.  
OwTrue
```

Essa maneira que o site remoto irá entregar diretamente pra você, sem ficar tentando conexão com o cliente. Você então envia para seu cliente. Somente funciona para ``hosts'', então você precisa obter o nome da máquina de correio do seu cliente ``cliente.com'' assim como ``hostname.cliente.com'' no DNS. Soemnte coloque um registro A no DNS para ``cliente.com''.

20.5.4. Por que eu recebo o erro ``Relaying Denied" quando envio mensagens a partir de outros hosts?

Na instalação padrão do **sendmail** FreeBSD, é configurado para somente enviar mensagem a partir de hosts em que o MTA está rodando. Por exemplo, se um servidor POP3 está instalado, então usuários estarão aptos a checar mensagens da escola, trabalho, ou outros locais remotos mas eles ainda não estão aptos a enviar mensagens a partir de outras localidades de fora. Tipicamente, em alguns momentos depois da tentativa, um email será enviado do **MAILER-DAEMON** com a mensagem de erro ``5.7 Relaying Denied".

Há diversas maneiras para contornar isso. A solução mais direta é colocar o endereço do seu provedor de serviços internet no arquivo `relay-domains` em `/etc/mail/relay-domains`. Uma maneira rápida de fazer isso pode ser:

```
# echo "seu.provedor.exemplo.com" > /etc/mail/relay-domains
```

Depois de criar ou editar este arquivo você deve reiniciar o **sendmail**. isto funciona muito bem se você for um administrador de sistemas e não deseja enviar mensagens localmente, ou gostaria de usar um cliente gráfico em uma máquina ou para enviar mensagens através de um outro provedor de serviços internet. Isto também é muito útil se você tem somente uma ou duas contas de email configurada. Se houver um grande

número de endereços a adicionar, você pode simplesmente abrir esse arquivo no seu editor de texto preferido, e então adicionar os domínios um por linha:

```
seu.provedor.exemplo.com
outro.provedor.exemplo.net
usuarios-provedor.exemplo.org
www.exemplo.org
```

Agora toda mensagem enviada através do seu sistema, por qualquer um dos hosts nesta lista (desde que o usuário tenha uma conta em seu sistema), será sucedida. Esta é uma maneira muito agradável de permitir aos usuários o envio de mensagens remotamente a partir de seu sistema sem permitir que outras pessoas enviem SPAM através do seu sistema.

20.6. Tópicos avançados

A seguinte seção cobre tópicos mais comuns como configuração de correio e ajuste do correio para seu domínio inteiro.

20.6.1. Configuração básica

Fora da máquina, você deve poder enviar mensagens para hosts externos como você configurou no `/etc/resolv.conf` ou rodou o seu próprio servidor de nomes. Se você gostaria de receber mensagens em seu host entregue ao MTA (ex., **sendmail**) em seu próprio host FreeBSD, há dois métodos:

- Rode seu próprio servidor de nomes e tenham seu próprio domínio. Por exemplo, `FreeBSD.org`
- Puxe as mensagens entregue diretamente a seu host. Isto é feito entregando o correio diretamente ao nome atual do DNS da sua máquina. Por exemplo, `exemplo.FreeBSD.org`.

Indiferente quais dos métodos acima você escolha, a fim de ter o correio entregando diretamente para seu host, deve ter um endereço IP estático (não um endereço dinâmico, como a maioria das conexões discadas). Se você está atrás de um firewall, ele deve permitir tráfego SMTP para você. Se você quiser receber mensagens diretamente em seu host, você precisa certificar-se de qualquer uma dessas duas coisas:

- Certifique-se que o registro MX (com o menor valor) em seu DNS aponte para o endereço IP do seu host.
- Certifique-se que não há entrada MX em seu DNS para seu host.

Qualquer uma das coisas acima irá permitir que você receba mensagens diretamente em seu host.

Tente isto:

```
# hostname
exemplo.FreeBSD.org
# host exemplo.FreeBSD.org
exemplo.FreeBSD.org has address 204.216.27.XX
```

Se isto for o que você ver, mensagens diretamente para [<seulogin@exemplo.FreeBSD.org>](mailto:seulogin@exemplo.FreeBSD.org) deve funcionar sem problemas (presumindo que o **sendmail** está rodando corretamente em `exemplo.FreeBSD.org`).

Se preferivelmente você ver algo assim:

```
# host exemplo.FreeBSD.org
exemplo.FreeBSD.org has address 204.216.27.XX
exemplo.FreeBSD.org mail is handled (pri=10) by hub.FreeBSD.org
```

Toda mensagem enviada para seu host (`exemplo.FreeBSD.org`) terminará coletada na hub sob o mesmo nome de usuário em vez da emissão diretamente ao seu host.

A informação acima é segurada por seu servidor DNS. O registro DNS que contém a informação de roteamento de correio é a entrada *Mail eX*. Se não existir registro MX, as mensagens serão entregues diretamente para o host pelo endereço IP.

A entrada MX para `freefall.FreeBSD.org` está algo assim:

```
freefall      MX  30  mail.crl.net
freefall      MX  40  agora.rdrop.com
freefall      MX  10  freefall.FreeBSD.org
freefall      MX  20  who.cdrom.com
```

Como você pode ver, `freefall` tem várias entradas MX. O menor número MX é o host que recebe mensagens diretamente se estiver disponível; se ele não estiver acessível por alguma razão, os outros (às vezes chamado de "backups MXs") aceitam mensagens temporariamente, e passa quando um host de número menor se torna disponível, eventualmente ao host de menor número.

Sites MX alternativos devem ter conexões de internet separadas de você próprio a fim de ser mais úteis. Seu provedor de internet ou outro local amigável não devem ter problema de fornecer esse serviço para você.

20.6.2. Mensagens de correio para seu domínio

A fim de configurar um "mailhost" (chamado servidor de correio) você precisa ter todo o correio emitidos a varias estações dirigidas a ele. Basicamente, você quer "solicitar" qualquer mensagem para qualquer hostname no seu domínio (neste caso `*.FreeBSD.org`) e desviá-la para seu servidor de correio para que então seus usuários podem receber suas mensagens no servidor de correio principal.

Para facilitar as coisas, uma conta de usuário com o mesmo *nome de usuário* deve existir em ambas as máquinas. Use o [adduser\(8\)](#) para fazer isto.

O host de correio que você estará usando deve ser o entregador de correio designado para cada estação na rede. Isto é feito na sua configuração de DNS, dessa forma:

```
exemplo.FreeBSD.org    A    204.216.27.XX        ; estação
                       MX   10 hub.FreeBSD.org    ; host de correio
```

Isto irá direcionar mensagens para cada estação de trabalho no host de correio não importando onde estão os pontos de registro A. A mensagem é enviada para o host MX.

Você não pode fazer isso você mesmo, ao menos que você esteja rodando um servidor DNS. Se você não tem, ou não pode rodar seu próprio servidor DNS, fale com seu provedor de serviços internet ou com alguém que possa lhe fornecer seu DNS.

Se você estiver hospedando emails virtuais, a seguinte informação virá em boa hora. Por exemplo, nós iremos assumir que você tem um cliente com seu próprio domínio, neste caso `cliente1.org`, e você quer que todas mensagens para `cliente1.org` sejam enviadas para seu host de correio, `mail.meuhost.com`. Sua entrada no DNS deve ficar algo assim:

```
cliente1.org           MX   10   mail.meuhost.com
```

Você *não* precisa de um registro A para `cliente1.org` se você quiser somente gerenciar emails para este domínio.

Nota: Esteja ciente que o ping para o `cliente1.org` não irá funcionar a menos que um registro A exista para ele.

A última coisa que você deve fazer é dizer ao **sendmail** no seu host de correio, quais domínios e/ou hostnames devem aceitar mensagens. Há algumas maneiras diferentes que isto pode ser feito. Qualquer uma das seguintes formas irão funcionar:

- Adicione os hosts no seu arquivo `/etc/mail/local-host-names` se você estiver usando o *FEATURE(use_cw_file)*. Se você estiver usando uma versão do **sendmail** anterior a 8.10, o arquivo é `/etc/sendmail.cw`.
- Adicione a linha *Cwseu.host.com* em seu `/etc/sendmail.cf` ou `/etc/mail/sendmail.cf` se você estiver usando o **sendmail** 8.10 ou mais recente.

20.7. SMTP com UUCP

A configuração do **sendmail** que o FreeBSD traz é voltada para locais que estejam conectados diretamente à internet. Locais que desejam trocar mensagens via UUCP devem instalar outro arquivo de configuração do **sendmail**.

Ajustar o `/etc/mail/sendmail.cf` manualmente é um tópico avançado. O **sendmail** na versão 8 gera os arquivos de configuração pelo [m4\(1\)](#) através de pré-processamento,

onde a configuração atual ocorre em um nível mais alto de abstração. Os arquivos de configuração do [m4\(1\)](#) podem ser encontradas sob `/usr/src/usr.sbin/sendmail/cf`.

Se você não tiver instalado seu sistema com todos os fontes, a configuração do sendmail vai estar quebrado em um arquivo de fonte separado na distribuição. Assumindo que você tem o código fonte do FreeBSD montando em CDROM, faça:

```
# cd /cdrom/src
# cat scontrib.?? | tar xzf - -C /usr/src/contrib/sendmail
```

Isto irá descompactar somente alguns cem kilobytes. O arquivo `README` no diretório `cf` pode servir como uma introdução básica a configuração do m4.

A melhor maneira de suportar a entrega UUCP é usar a característica *mailertable*. Isto cria uma base de dados que o **sendmail** pode usar para tomar decisões de roteamento.

Primeiro, você deve criar seu arquivo `.mc` O diretório

`/usr/src/usr.sbin/sendmail/cf/cf` contém alguns exemplos. Assumindo que você nomeou seu arquivo como `foo.mc`, tudo que você precisa fazer é convertê-lo em um `sendmail.cf` válido:

```
# cd /usr/src/usr.sbin/sendmail/cf/cf
# make foo.cf
# cp foo.cf /etc/mail/sendmail.cf
```

Um típico arquivo `.mc` pode ser dessa forma:

```
VERSIONID(`O número de sua versão') OSTYPE(bsd4.4)

FEATURE(accept_unresolvable_domains)
FEATURE(nocanonify)
FEATURE(mailertable, `hash -o /etc/mail/mailertable')

define(`UUCP_RELAY', seu.uucp.relay)
define(`UUCP_MAX_SIZE', 200000)
define(`confDONT_PROBE_INTERFACES')

MAILER(local)
MAILER(smtp)
MAILER(uucp)

Cw    seu.apelido.host.nome
Cw    seunomeuucp.UUCP
```

As linhas que contém as características *accept_unresolvable_domains*, *nocanonify*, e *confDONT_PROBE_INTERFACES* impedirão todo o uso do DNS durante a entrega de mensagens. A opção *UUCP_RELAY* é necessária suportar entrega UUCP. Coloque simplesmente um hostname da internet lá que esteja apto a gerenciar. Endereços do pseudo-domínio do UUCP; muito provável que, você irá entrar lá no relay de mensagens do seu provedor de serviços internet.

Uma vez que você tenha isto, você precisa do arquivo `/etc/mail/mailertable`. Se você tem somente um link para fora que estará sendo usada para todos os seu emails, a seguinte linha será suficiente:

```
#
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
.
    uucp-dom:seu.uucp.relay
```

Um exemplo mais complexo pode ser como este:

```
#
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
#
horus.interface-business.de    uucp-dom:horus
.interface-business.de        uucp-dom:if-bus
interface-business.de          uucp-dom:if-bus
.heep.sax.de                   smtp8:%1
horus.UUCP                     uucp-dom:horus
if-bus.UUCP                    uucp-dom:if-bus
.                               uucp-dom:
```

As primeiras três linhas gerenciam especialmente casos onde o domínio endereçado de mensagens não pode ser enviado para fora pela rota padrão, mas preferivelmente ser entregue por algum "atalho" vizinho UUCP. As próximas linhas gerencia para o domínio ethernet local que pode ser entregue usando SMTP. Finalmente, os vizinhos UUCP são mencionados na notação do pseudo-domínio .UUCP, para permitir um *uucp-vizinho de !recipiente* sobreescreva as regras padrões. A última linha é sempre somente um ponto único, colocando tudo mais, com a entrega UUCP a um vizinho do UUCP que sirva como gateway de correio universal para o mundo. Todos os nomes de estação atrás *uucp-dom:* palavra chave deve ser um vizinho UUCP válido como você pode verificar usando o comando *uuname*.

Como lembrete, este arquivo necessita ser convertido em uma base de dados DBM antes de ser usado. A linha de comando para realizar isto é colocada como comentário no alto do mailertable. Você sempre tem que executar esse comando cada vez que você modificar seu mailertable.

Sugestão final: se você estiver incerto de algum roteamento particular que o correio deveria funcionar, lembre da opção *-bt* do **sendmail**. Isto inicia o **sendmail** em *modo de teste de endereços*; simplesmente entre com *3,0*, seguido pelo endereço que você deseja testar para roteamento de correio. A última linha fala o agente de correio interno usado, o host de destino que esse agente estará chamando, e o (possivelmente traduzindo) endereço. Saia deste modo digitando **Ctrl+D**.

```
% sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 foo@exemplo.com
canonify      input: foo @ exemplo . com
...
parse        returns: $# uucp-dom $@ seu.relay.uucp $: foo < @ exemplo
. com . >
> ^D
```

20.8. Usando o correio com uma conexão discada

Se você tiver um endereço IP estático, você não deve necessitar fazer nenhum ajuste na configuração padrão. Coloque o nome atribuído a sua estação internet e o **sendmail** fará o resto.

Se você tem um endereço IP dinâmico e usa uma conexão discada para se conectar a internet, você provavelmente tem uma conta de correio no servidor do seu provedor. Vamos assumir que o domínio do seu provedor é `exemplo.net`, e seu nome de usuário é `usuario`, você chamou sua máquina de `bsd.casa`, e seu provedor falou pra você que pode ser usado `relay.exemplo.net` como relay de correio.

A fim de puxar suas mensagens de sua conta de correio no provedor, você deve instalar um agente, puxe as mensagens. O utilitário **fetchmail** é uma boa escolha que suporta vários tipos diferentes de protocolos. Normalmente, seu provedor de internet irá fornecer POP3. Se você está usando PPP no modo usuário, você pode automaticamente baixar seus emails quando a conexão internet estiver estabilizada com a seguinte entrada no seu `/etc/ppp/ppp.linkup`:

```
MYADDR:
!bg su usuario -c fetchmail
```

Se você estiver usando o **sendmail** (como mostrado abaixo) para entregar mensagens para contas não locais, você provavelmente quer mandar o **sendmail** processar sua fila de mensagens assim que sua conexão da internet for estabelecida. Para fazer isso, coloque este comando depois do comando do `fetchmail` em `/etc/ppp/ppp.linkup`.

```
!bg su usuario -c "sendmail -q"
```

Assumindo que você tem uma conta para `usuario` em `bsd.casa`. No diretório `home` do `usuario` em `bsd.casa`, crie o arquivo `.fetchmailrc`:

```
poll exemplo.net protocol pop3 fetchall pass minhasenha
```

Este arquivo não deve ser lido por ninguém exceto o `usuario` que contém a senha *minhasenha*.

A fim de enviar o correio com o correto cabeçalho *from:*, você deve dizer ao **sendmail** para usar `usuario@exemplo.net` em vez de `usuario@bsd.casa`. Você pode também desejar dizer ao **sendmail** para enviar todas as mensagens por `relay.exemplo.net`, permitindo a transmissão mais rápida de mensagens.

O seguinte arquivo `.mc` deve ser suficiente:

```
VERSIONID(`bsd.casa.mc version 1.0')
OSTYPE(bsd4.4)dnl
FEATURE(nouucp)dnl
MAILER(local)dnl
```

```
MAILER(smtp)dnl
Cwlocalhost
Cwbsd.casa
MASQUERADE_AS(`exemplo.net')dnl
FEATURE(allmasquerade)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(nocanonify)dnl
FEATURE(nodns)dnl
define(`SMART_HOST', `relay.exemplo.net')
Dmbsd.casa
define(`confDOMAIN_NAME', `bsd.casa')dnl
define(`confDELIVERY_MODE', `deferred')dnl
```

Consulte a seção anterior para detalhes de como fazer deste arquivo `.mc` um arquivo `sendmail.cf`. Também, não esqueça de reiniciar o **sendmail** depois de atualizar o `sendmail.cf`.

20.9. Autenticação SMTP

Ter uma autenticação SMTP em seu servidor de correio trás vários benefícios. Autenticação SMTP pode adicionar uma outra camada de segurança ao **sendmail**, e trás o benefício de dar aos usuários móveis que alternam seus hosts a possibilidade de usar o mesmo servidor de correio sem a necessidade de reconfigurar seu cliente de correio toda hora.

1. Instale o [security/cyrus-sasl](#) a partir do ports. Você pode achar este port em [security/cyrus-sasl](#). [security/cyrus-sasl](#) tem várias opções na hora da compilação para serem escolhidas, para o método que nós estaremos usando aqui, certifique-se de selecionar a opção *pwcheck*.
2. Depois de instalar [security/cyrus-sasl](#), edite o `/usr/local/lib/sasl/Sendmail.conf` (ou crie-o se não existir) e adicione a seguinte linha:
3. `pwcheck_method: passwd`

Este método irá habilitar o **sendmail** para autenticar na base de dados `passwd` do seu FreeBSD. Isto resolve o problema de criar um novo usuário e senha para cada usuário que necessitar a autenticação SMTP, e mantém o login e senha e email iguais.

4. Agora edite o `/etc/make.conf` e adicione as seguintes linhas:
5. `SENDMAIL_CFLAGS=-I/usr/local/include/sasl1 -DSASL`
6. `SENDMAIL_LDFLAGS=-L/usr/local/lib`
7. `SENDMAIL_LDADD=-lsasl`

Estas linhas irão dar ao **sendmail** a opção de configuração adequada para linkar para o [cyrus-sasl](#) no tempo de compilação. Certifique-se que o [cyrus-sasl](#) está instalado antes de recompilar o **sendmail**.

8. Recompile o **sendmail** executando os seguintes comandos:
9. `# cd /usr/src/usr.sbin/sendmail`


```
10. # make cleandir
11. # make obj
12. # make
13. # make install
```

A compilação do **sendmail** não deve ter qualquer problema se o `/usr/src` não foi alterado bastando ter e as bibliotecas compartilhadas necessárias estarem disponíveis.

14. Depois do **sendmail** ter sido compilado e reinstalado, edite o seu arquivo `/etc/mail/freebsd.mc` (ou qualquer arquivo que você use como arquivo `.mc`). Muitos administradores escolhem usar a saída do [hostname\(1\)](#) como nome do arquivo `.mc`). Adicione estas linhas a ele:

```
15. dnl set SASL options
16. TRUST_AUTH_MECH(`GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN')dnl
17. define(`confAUTH_MECHANISMS', `GSSAPI DIGEST-MD5 CRAM-MD5
LOGIN')dnl
18. define(`confDEF_AUTH_INFO', `/etc/mail/auth-info')dnl
```

Estas opções configuram métodos diferentes disponíveis para o **sendmail** autenticar usuários. Se você gostaria de usar um método diferente do **pwcheck**, por favor veja a documentação inclusa do programa.

19. Finalmente, execute [make\(1\)](#) dentro do `/etc/mail`. Isto irá gerar seu novo arquivo `.mc` e criar um arquivo `.cf` chamado `freebsd.cf` (ou algum outro nome que você usou para seu arquivo `.mc`). Então use o comando `make install restart`, que irá copiar o arquivo para o `sendmail.cf`, e irá reiniciar corretamente o **sendmail**. Para mais informações sobre este processo, você deve consultar o `/etc/mail/Makefile`.

Se tudo ocorreu corretamente, você deve poder inserir sua informação de login em um cliente de emails e enviar uma mensagem de teste. Para uma investigação mais profunda, configure o *LogLevel* do **sendmail** para 13 e observe o `/var/log/maillog` para qualquer erro.

Você pode desejar adicionar as seguintes linhas ao `/etc/rc.conf` para que este serviço esteja disponível depois de cada reinicialização do sistema:

```
sasl_pwcheck_enable="YES"
sasl_pwcheck_program="/usr/local/sbin/pwcheck"
```

Isto irá assegurar que a inicialização do SMTP_AUTH será feita no carregamento do sistema.

Para mais informações, por favor veja a página do **sendmail** a respeito da [autenticação SMTP](#).

Capítulo 21. A Versão Mais Recente

Índice

- 21.1. [Sinopse](#)
- 21.2. [FreeBSD-CURRENT vs. FreeBSD-STABLE](#)
- 21.3. [Sincronizando Seus Fontes](#)
- 21.4. [Usando o `make world`](#)
- 21.5. [Preparando Múltiplas Máquinas](#)

Reestruturado, reorganizado, e partes atualizadas por Jim Mock. Trabalho original por Jordan Hubbard, Poul-Henning Kamp, John Polstra, e Nik Clayton.

21.1. Sinopse

O FreeBSD está constantemente sendo desenvolvido entre as suas versões. Para pessoas que querem estar sempre com as últimas novidades, existem uma variedade de mecanismos que facilitam manter seu sistema sincronizado com os últimos desenvolvimentos. Esteja avisado--as últimas novidades não são para todos! Este capítulo lhe ajudará a decidir se você quer acompanhar o desenvolvimento do sistema, ou permanecer em uma das versões liberadas.

Após ler este capítulo, você saberá:

- As diferenças entre as duas árvores de desenvolvimento: FreeBSD-STABLE e FreeBSD-CURRENT.
- Como manter seu sistema atualizado com **CVSup**, **CVS**, ou **CTM**.
- Como reconstruir e reinstalar toda a base do sistema com `make world`.

Antes de ler este capítulo, você deve:

- Configurar adequadamente sua conexão de rede ([Capítulo 19](#)).
- Saber como instalar programas adicionais de terceiros ([Capítulo 4](#)).

21.2. FreeBSD-CURRENT vs. FreeBSD-STABLE

Existem duas árvores de desenvolvimento para o FreeBSD: FreeBSD-CURRENT e FreeBSD-STABLE. Esta seção explicará um pouco sobre cada uma e descreverá como manter seu sistema atualizado em cada árvore. FreeBSD-CURRENT será discutido primeiro, depois o FreeBSD-STABLE.

21.2.1. Mantendo-se Atualizado com o FreeBSD

Enquanto você lê isto, tenha em mente que o FreeBSD-CURRENT é a "última das últimas" versões de desenvolvimento do FreeBSD. Espera-se que os usuários do

FreeBSD-CURRENT tenham um alto nível de habilidades técnicas, e que possam ser capazes de resolver problemas por si só. Se você é novo no FreeBSD, pense duas vezes antes de instalar esta versão.

21.2.1.1. O que é o FreeBSD-CURRENT?

FreeBSD-CURRENT são os últimos fontes trabalhados do FreeBSD. Isto inclui trabalho em progresso, mudanças experimentais, e mecanismos transacionais que podem ou não estar presentes na próxima versão original do sistema. Enquanto muitos desenvolvedores do FreeBSD compilam o código fonte do FreeBSD-CURRENT diariamente, existem períodos de tempo em que os fontes não são compiláveis. Estes problemas são resolvidos o mais depressa possível, se o FreeBSD-CURRENT causar ou não um desastre ou trazer uma funcionalidade muito desejada, isto pode ser uma questão de que em qual momento você pegou o código fonte!

21.2.1.2. Quem precisa do FreeBSD-CURRENT?

FreeBSD-CURRENT é disponibilizado para 3 grupos primários de interesse:

1. Membros do grupo do FreeBSD que trabalhem ativamente em alguma parte da árvore dos códigos fonte e para quem tem uma grande necessidade de estar extremamente ``atualizado".
2. Membros do grupo do FreeBSD que são testadores ativos, com disposição de gastar tempo resolvendo problemas para certificar-se que o FreeBSD-CURRENT permaneça o mais utilizável possível. Também existem pessoas que desejam dar pequenas sugestões em mudanças e nas direções gerais do FreeBSD, e submeter correções para implementá-las.
3. Aqueles que simplesmente querem observar as coisas, ou usar os fontes com o propósito de referência (e.g. para *leitura*, e não execução). Estas pessoas também fazem comentários ocasionais ou contribuem com código.

21.2.1.3. O que o FreeBSD-CURRENT Não é?

1. Um atalho para ter um gostinho da versão que está para ser liberada só porque você ouviu falar de alguma funcionalidade legal que está lá e você quer ser o primeiro na sua rua a tê-la. Ser o primeiro a ter as novas funcionalidades significa ser o primeiro a ter os novos bugs.
2. Uma maneira rápida de pegar correções de bugs. Qualquer versão do FreeBSD-CURRENT é passível de introduzir novos bugs assim como corrigir os existentes.
3. ``Oficialmente suportado". Nós fazemos o melhor para verdadeiramente ajudar as pessoas em um dos 3 grupos ``legítimos" do FreeBSD-CURRENT, mas nós simplesmente *Não temos tempo* para oferecer suporte técnico. Isto não é porque nós somos pessoas mesquinhas e nojentas que não gostam de ajudar os outros se fossemos nem estariamos fazendo o FreeBSD). Nós simplesmente não podemos responder centenas de mensagens por dia e trabalhar no FreeBSD! Entre melhorar o FreeBSD e responder centenas de mensagens sobre código experimental, os desenvolvedores optaram pela primeira.

21.2.1.4. Usando o FreeBSD-CURRENT

1. Entre nas listas [freebsd-current](#) e [cvs-all](#). Isto não é apenas uma boa idéia, é *essencial*. Se você não está na lista [freebsd-current](#), não verá os comentários que as pessoas estão fazendo a respeito do estado atual do sistema, desta forma você provavelmente vai acabar enfrentando um monte de problemas que outros já encontraram e solucionaram. Ainda mais importante, você perderá boletins importantes que podem ser críticos para que seu sistema continue saudável.

A lista [cvs-all](#) permitirá que você veja o *log* de cada mudança submetida tão logo elas estejam sendo feitas e informações sobre possíveis efeitos colaterais.

Para entrar nestas listas, ou qualquer outra disponível, vá em <http://lists.FreeBSD.org/mailman/listinfo> e clique na lista que você deseja assinar. Instruções para o restante do procedimento estão disponíveis lá também.

2. Pegue os fontes de um dos [sites espelho do FreeBSD](#). . Você pode fazer isso de duas maneiras:
 - a. Use o programa [cvsup](#) com o arquivo `supfile` de nome `standard-supfile` disponível em `/usr/share/examples/cvsup`. Este é o método mais recomendado, uma vez que permite a você pegar toda a coleção uma vez e a partir daí pegar apenas as atualizações. Muitas pessoas executam o `cvsup` a partir do `cron` e mantém seus fontes atualizados automaticamente. Você precisa customizar o arquivo `supfile` de exemplo, e configurar o [cvsup](#) para o seu ambiente.
 - b. Use o aplicativo [CTM](#). Se você possui uma péssima conectividade (conexões muito caras ou apenas endereço eletrônico) o [CTM](#) é uma opção. Entretanto, é um pouco problemático, podendo te dar arquivos corrompidos. Isto leva este aplicativo a ser raramente usado, o que aumenta a chance do mesmo não funcionar por longos períodos de tempo. Recomendamos o uso do [CVSup](#) para qualquer um que tenha um modem de 9600 bps ou uma conexão mais rápida.
3. Se você está pegando os fontes para executar, e não apenas para olhar, então pegue *todo* o FreeBSD-CURRENT, e não apenas algumas partes. A razão para isto é que várias partes dos fontes dependem de atualizações em outros lugares, e tenta compilar apenas um subconjunto é uma quase garantia que você vai ter problemas.

Antes de compilar o FreeBSD-CURRENT, leia o arquivo `Makefile` em `/usr/src` cuidadosamente. Você deve executar o [make world](#) na primeira parte como parte do processo de atualização. Ler o [lista de discussão FreeBSD-CURRENT](#) e o arquivo `/usr/src/UPDATING` manterá você atualizado para outros processos de início que algumas vezes se fazem necessários ao caminharmos para uma próxima versão.

4. Seja ativo! Se você está executando o FreeBSD-CURRENT, queremos saber o que você tem a dizer sobre ele, especialmente se você tem sugestões para melhoras ou conserto para problemas. Sugestões junto com código são recebidas de forma muito entusiasmada!

21.2.2. Permanecendo Estável com o FreeBSD

21.2.2.1. O que é o FreeBSD-STABLE?

FreeBSD-STABLE é a nossa árvore de desenvolvimento vindas de versões principais. Mudanças acontecem nesta árvore de maneira diferente, e com a suposição geral de que foram previamente feitas no FreeBSD-CURRENT em caráter de teste. Esta *ainda* é uma árvore de desenvolvimento, entretanto, e isto pode significar a qualquer momento, os fontes do FreeBSD-STABLE podem ou não ser adequados para um propósito particular. Esta é simplesmente outra faixa de engenharia de desenvolvimento, não uma fonte para usuários finais.

21.2.2.2. Quem Precisa do FreeBSD-STABLE?

Se você está interessado em acompanhar ou contribuir com o processo de desenvolvimento do FreeBSD, então você deve considerar o FreeBSD-STABLE.

Sendo verdadeiro o fato de que consertos de segurança também sejam implementados na árvore do FreeBSD-STABLE você não *precisa* acompanhar o FreeBSD-STABLE para fazer isso. Cada boletim de segurança para o FreeBSD explica como consertar o problema para cada versão afetada [1], e acompanhar uma árvore de desenvolvimento inteira apenas por razões de segurança é passível de trazer para a sua máquina uma série de mudanças indesejáveis também.

Embora nós nos esforcemos para ter certeza de que a árvore FreeBSD-STABLE compile e execute sempre, isto não pode ser garantido. Além disso, enquanto o código é escrito no FreeBSD-CURRENT antes de ser incluído no FreeBSD-STABLE, mais pessoas executam o FreeBSD-STABLE do que o FreeBSD-CURRENT, então é inevitável que algumas vezes serão encontradas falhas no FreeBSD-STABLE que aparentemente não estão no FreeBSD-CURRENT.

Por estas razões, nós *não* recomendamos que você siga cegamente o FreeBSD-STABLE, e é particularmente importante que você não atualize qualquer servidor em produção para o FreeBSD-STABLE sem antes fazer um teste completo no código em seu ambiente de desenvolvimento.

Se você não possui recursos para fazer isso nós recomendamos que você execute a versão ``release" mais recente do FreeBSD, e utilize o mecanismo de atualização binário para se atualizar de ``release" em ``release".

21.2.2.3. Usando o FreeBSD-STABLE

1. Entre na lista [freebsd-stable](#). Isto vai te manter informado sobre as dependências de compilação que podem surgir no FreeBSD-STABLE ou quaisquer outros problemas que necessitem de atenção especial. Os desenvolvedores farão anúncios nesta lista quando estiverem vendo algum conserto ou atualização controversa, oferecendo aos usuários a chance de responder se possuem algo a dizer a respeito desta mudança proposta.

A lista [cvs-all](#) permitirá que você acompanhe a entrada de submissão de *log* para cada mudança na medida que forem sendo feitas e informações pertinentes sobre possíveis efeitos colaterais.

Para entrar nestas listas, ou uma das outras disponíveis, vá em <http://lists.FreeBSD.org/mailman/listinfo> e clique na lista em que deseja se cadastrar. Instruções para o restante do procedimento estão lá.

2. Se você está instalando um novo sistema e quer que ele seja o mais estável possível, você pode simplesmente pegar o último código compilado de <ftp://releng4.FreeBSD.org/pub/FreeBSD/> e instalar como qualquer outra release.

Se você já está executando uma versão anterior do FreeBSD e deseja atualizá-la através dos fontes você pode fazer isso com facilidade a partir dos [sites espelho](#) do FreeBSD. Isto pode ser feito de duas maneiras:

- a. Use o programa [cvsup](#) com o arquivo `supfile` de nome `stable-supfile` do diretório `/usr/share/examples/cvsup`. Este é o método mais recomendado, uma vez que permite que você pegue toda a coleção de fontes uma única vez e, daí em diante, pegar somente o que foi modificado. Muitas pessoas executam o `cvsup` a partir do `cron` para manter seus fontes atualizados automaticamente. Você precisa customizar o arquivo `supfile` de exemplo acima, e configurar o [cvsup](#) para o seu ambiente.
 - b. Use o [CTM](#). Se você não possui uma conexão rápida e barata com a Internet, este é o método que você deve considerar.
3. Essencialmente, se você precisa de acesso rápido sob demanda aos fontes e a banda de comunicação não é problema, use o `cvsup` ou o `ftp`. Se não, use o **CTM**.
 4. Antes de compilar o FreeBSD-STABLE, leia o arquivo `Makefile` em `/usr/src` cuidadosamente. Você deve executar um [make world](#) na primeira vez como parte do processo de atualização. A leitura da lista [lista de discussão FreeBSD-STABLE](#) e do arquivo `/usr/src/UPDATING` manterá você atualizado em outros processos iniciais que algumas vezes podem ser necessários para mover-se para a próxima release.

Notas

- [1] Isto não é completamente verdadeiro. Nós não podemos dar suporte para versões antigas do FreeBSD para sempre, por favor veja <http://www.FreeBSD.org/security/>.

21.3. Sincronizando Seus Fontes

Existem várias maneiras usando uma conexão Internet (ou de email) para ficar atualizado com qualquer área dos fontes do projeto FreeBSD, ou todas as áreas, dependendo do seu interesse. Os serviços primários que oferecemos são [CVS Anônimo](#), [CVSup](#), e [CTM](#).

Atenção Enquanto é possível atualizar apenas partes da sua árvore dos fontes, o único procedimento de atualização com suporte é o de atualizar toda a árvore e recompilar os fontes de nível de usuário (i.e., todos os programas que são executados no espaço do usuário, tais como aqueles em `/bin` e `/sbin`) e fontes do *kernel*. Atualizar apenas parte da sua árvore de fontes, ou apenas os fontes a nível de usuário, frequentemente resultará em problemas. Estes problemas podem variar de erros de compilação a panico no *kernel* ou corrupção de dados.

CVS Anônimo e **CVSup** usam o modelo *pull* de atualização dos fontes. No caso do **CVSup** o usuário (ou um *script cron*) invoca o programa `cvsup`, e ele interage com um servidor `cvsupd` em algum lugar para fazer com que seus arquivos fiquem atualizados. As atualizações que você recebe são as mais recentes e você as obtém, somente quando você as quer. Você pode facilmente restringir suas atualizações à arquivos ou diretórios específicos que são do seu interesse. As atualizações são geradas rapidamente pelo servidor, de acordo com o que você tem e o que você quer ter. **CVS Anônimo** é um pouco mais simples do que o **CVSup**, sendo apenas uma extensão do **CVS** que permite puxar as mudanças diretamente de um repositório CVS. **CVSup** pode fazer isto de maneira bem mais eficiente, mas o **CVS Anônimo** é mais fácil de usar.

O **CTM**, por outro lado, não compara interativamente os fontes que você tem com os que estão no arquivo principal ou com os que foram puxados. Ao invés disso, um *script* que identifica as mudanças nos arquivos desde a última vez que o **CTM** foi usado é executado várias vezes por dia na máquina **CTM** principal, as mudanças são comprimidas, estampadas com um número sequencial e codificadas para transmissão por correio eletrônico (apenas em caracteres ASCII imprimíveis). Uma vez recebidos, estes "deltas **CTM**" podem ser manipulados pelo utilitário [ctm_rmail\(1\)](#) que decodificará automaticamente, verificará e aplicará as mudanças nos fontes do usuário. Este processo é bem mais eficiente do que o **CVSup**, e não sobrecarrega o nosso servidor uma vez que o modelo *push* ao invés do *pull* é utilizado.

Existem alguns contratempos, é claro. Se você inadvertidamente apagar parte dos seus fontes, o **CVSup** detectará e reconstruirá as partes danificadas para você. O **CTM** não fará isto, se você apagar alguma parte da sua árvore de fontes (e não possuir cópia de segurança) então você vai ter que começar do zero (da mais recente "base delta" do CVS) e reconstruir tudo com o **CTM** ou, com o **CVS Anônimo**, simplesmente apagar os bits problemáticos e resincronizar.

21.4. Usando o `make world`

Uma vez sincronizados os fontes locais com uma versão particular do FreeBSD (FreeBSD-STABLE, FreeBSD-CURRENT, e assim por diante) você pode usar a árvore dos fontes para reconstruir o sistema.

Tire uma Cópia de Segurança: Nunca é demais falar como é importante tirar uma cópia de segurança do seu sistema *antes* de fazer isso. Enquanto que reconstruir tudo é (se você seguir estas instruções) uma tarefa fácil, inevitavelmente existirão ocasiões em que você cometerá erros, ou quando erros cometidos por outros na árvore dos fontes fazendo com que seu sistema não inicialize.

Tenha certeza de que você tirou uma cópia de segurança. E tenha um disquete de boot em mãos. Você provavelmente nunca vai precisar utilizá-lo, mas é melhor prevenir do que remediar!

Inscreva-se na Lista de Discussão Certa: O FreeBSD-STABLE e FreeBSD-CURRENT estão, por sua natureza, *em desenvolvimento*. Pessoas que contribuem com o FreeBSD são humanas, e erros acontecem de vez em quando.

Algumas vezes estes erros podem ser um pouquinho nocivos, apenas fazendo com que seu sistema imprima uma mensagem de diagnóstico de alerta. Ou as mudanças podem ser catastróficas, fazendo com que seu sistema não inicialize mais ou destrua seus sistemas de arquivos (ou pior).

Se acontecerem problemas como estes, uma ``nota" é postada na lista apropriada, explicando a natureza do problema e quais sistemas são afetados por ele. E um anúncio ``tudo limpo" é postado quando o problema foi solucionado.

Se você tentar acompanhar o FreeBSD-STABLE ou o FreeBSD-CURRENT e não ler a [lista de discussão FreeBSD-STABLE](#) ou a [lista de discussão FreeBSD-CURRENT](#) respectivamente, então você está caçando problemas.

21.4.1. Leia o Arquivo `/usr/src/UPDATING`

Antes de fazer qualquer coisa, leia o arquivo `/usr/src/UPDATING` (ou o arquivo equivalente sempre que você possuir uma cópia dos códigos fontes). Este arquivo pode conter informações importantes a respeito de problemas que você pode encontrar, ou especifica a ordem na qual você deve executar certos comandos. Se o arquivo `UPDATING` contradizer algo que você ler aqui, ele tem precedência.

Importante: Ler o `UPDATING` não substituí a inscrição e acompanhamento da lista de discussão correta, como mencionado anteriormente. Os dois requisitos são complementares, não exclusivos.

21.4.2. Verifique o `/etc/make.conf`

Examine os arquivos `/etc/defaults/make.conf` e `/etc/make.conf`. O primeiro contém algumas definições padrões - muitas comentadas. Para fazer uso delas quando você reconstruir seu sistema dos fontes, adicione-as ao arquivo `/etc/make.conf`. Tenha em mente que qualquer coisa que você adicione ao arquivo `/etc/make.conf` também é usado sempre que você executar o comando `make`, então é uma boa idéia configurá-lo para alguma coisa sensível para seu sistema.

Um usuário típico provavelmente copiará as linhas `CFLAGS` e `NOPROFILE` encontradas em `/etc/defaults/make.conf` para `/etc/make.conf` e removerá seus comentários.

Examine outras definições (`COPTFLAGS`, `NOPORTDOCS` e assim por diante) e decida se elas são relevantes para você.

21.4.3. Atualize os Arquivos em `/etc`

O diretório `/etc` contém grande parte da informação de configuração do seu sistema, assim como *scripts* que são executados na inicialização do sistema. Alguns destes *scripts* mudam de versão para versão do FreeBSD.

Alguns dos arquivos de configuração também são usados diariamente na execução do sistema. Particularmente, o `/etc/group`.

Existiram ocasiões em que algumas partes da instalação do "make world" esperava encontrar certos nomes de usuários ou grupos no sistema. Ao fazer uma atualização é bem provável que estes usuários ou grupos não existissem. Isto causava problemas na atualização.

Um exemplo recente foi quando o usuário `smmisp` foi adicionado. Os usuários tiveram falhas no processo de instalação quando o [mtree\(8\)](#) tentava criar o arquivo `/var/spool/clientmqueue`.

A solução é examinar o arquivo `/usr/src/etc/group` e comparar com o seu arquivo de grupos. Se existirem grupos no arquivo novo que não existirem no seu então copie-os. Similarmente, você deve renomear quaisquer grupos no `/etc/group` que possuírem o mesmo `GID` mas um nome diferente dos que estão em `/usr/src/etc/group`.

Desde o 4.6-RELEASE você pode executar o [mergemaster\(8\)](#) antes da construção do sistema com a opção `-p`. Isto comparará apenas aqueles arquivos que são essenciais para o sucesso do `buildworld` ou `installworld`. Se a sua versão antiga do `mergemaster` não suportar a opção `-p`, use a versão mais nova na árvore dos fontes ao executá-lo pela primeira vez:

```
# cd /usr/src/usr.sbin/mergemaster
# ./mergemaster.sh -p
```

Dica: Se você está se sentindo particularmente paranóico, você pode checar seu sistema para ver quais arquivos foram possuídos pelo grupo que você está renomeando ou apagando:

```
# find / -group GID -print
```

mostrará todos os arquivos pertencidos pelo grupo *GID* (que pode ser um nome de grupo ou um ID numérico de grupo).

21.4.4. Vá para o Modo Mono Usuário

Você pode querer compilar o sistema no modo mono usuário. Sem mencionar o benefício óbvio de fazer com que as coisas andem mais rápido, a reinstalação do sistema tocará em vários arquivos de sistema importantes, todos os binários padrão de sistema, bibliotecas, arquivos de inclusão e assim por diante. Modificá-los com o sistema no ar (particularmente se você possuir usuários ativos neste período) é caçar problema.

Outro método é compilar o sistema em modo multi-usuário, e então ir para o modo mono usuário para a instalação. Se você quer fazer desta maneira, simplesmente siga os passos a seguir até que a construção do sistema tenha sido completada. Você pode deixar para ir para o modo mono usuário em uma outra hora até que você tenha que fazer o `installkernel` ou o `installworld`.

Como super usuário, você pode executar:

```
# shutdown now
```

de um sistema em execução, que irá para o modo mono usuário.

Alternativamente, reinicialize o sistema, e na linha de comando, entre com a opção `-s`. O sistema iniciará em modo mono usuário. Na linha de comando execute:

```
# fsck -p
# mount -u /
# mount -a -t ufs
# swapon -a
```

Isto verifica todos os sistemas de arquivo, remonta o `/` para leitura/escrita, monta todos os outros sistemas de arquivo UFS referenciados em `/etc/fstab` e liga o swap.

Nota: Se o seu relógio de CMOS está configurado para a hora local e não para GMT (isto é verdadeiro se a saída do comando [date\(1\)](#) não mostrar a zona e a hora corretamente), você pode ter que executar o seguinte comando:

```
# adjkerntz -i
```

Isto certificará que as suas configurações locais de zona/tempo sejam configuradas corretamente -- sem isto, você pode ter alguns problemas mais tarde.

21.4.5. Remova o /usr/obj

Na medida em que partes do sistema vão sendo reconstruídas são colocadas em diretórios que (por padrão) estão abaixo de /usr/obj. A sombra daqueles que estão em /usr/src.

Você pode acelerar o processo do ``make world'', e possivelmente evitar algumas dores de cabeça de dependência removendo este diretório.

Alguns arquivos abaixo de /usr/obj podem ter a opção imutável configurada (veja [chflags\(1\)](#) para mais informações) que deve ser removida antes.

```
# cd /usr/obj
# chflags -R noschg *
# rm -rf *
```

21.4.6. Recompila o Fonte

21.4.6.1. Gravando a Saída

É uma boa idéia gravar a saída que você tem do comando [make\(1\)](#) em outro arquivo. Se alguma coisa der errado você terá uma cópia da mensagem de erro. Enquanto isto não possa te ajudar a diagnosticar o que deu errado, pode ajudar outros se você postar seu problema em uma das listas de discussão do FreeBSD.

A maneira mais fácil de fazer isso é usar o comando [script\(1\)](#), com um parâmetro que especifica o nome do arquivo onde toda a saída será gravada. Você pode fazer isto imediatamente antes de reconstruir todo o sistema, e então digitar `exit` quando o processo terminar.

```
# script /var/tmp/mw.out
Script started, output file is /var/tmp/mw.out
# make TARGET
... compile, compile, compile ...
# exit
Script done, ...
```

Se você fizer isto, *não* grave a saída no diretório /tmp/. Este diretório pode ser apagado na próxima vez que você reinicializar o sistema. Um lugar melhor para gravar é /var/tmp/ (como no exemplo anterior) ou no diretório inicial do root.

21.4.6.2. Compilando a Base do Sistema

Você deve estar no diretório: /usr/src

```
# cd /usr/src
```

(a não ser que, claro, seu código fonte esteja em outro lugar, neste caso vá para este diretório).

Para reconstruir todo o seu sistema use o comando [make\(1\)](#). Este comando lê as instruções do arquivo `Makefile`, que descreve como os programas que compõem o FreeBSD devem ser recompilados, a ordem em que isto deve acontecer, e assim por diante.

O formato geral da linha de comando que você digitará é a seguinte:

```
# make -x -DVARIABLE target
```

Neste exemplo, `-x` é a opção que você deve passar para o [make\(1\)](#). Veja a página de manual do [make\(1\)](#) para exemplos de opções que você pode passar.

`-DVARIABLE` passa uma variável para o `Makefile`. O comportamento do `Makefile` é controlado por estas variáveis. Estas são as mesmas variáveis configuradas no `/etc/make.conf`, e isto é um outro modo de configurá-las.

```
# make -DNOPROFILE target
```

é outra maneira de especificar bibliotecas que não devem ser compiladas, e é o mesmo que a linha

```
NOPROFILE= true # Avoid compiling profiled libraries
```

no arquivo `/etc/make.conf`.

`target` diz ao [make\(1\)](#) o que você quer fazer. Cada `Makefile` define um número diferente de `targets`, e a sua escolha determina o que acontece.

Alguns `targets` são listados no arquivo `Makefile`, mas não foram feitos para que você os execute. Eles são usados pelo processo de compilação nos passos necessários para recompilar o sistema em vários sub passos.

Na maioria das vezes você não vai precisar passar qualquer parâmetro para o [make\(1\)](#), então sua linha de comando deve ficar parecida com:

```
# make target
```

Desde a versão 2.2.5 do FreeBSD (na verdade, primeiro foi criado na árvore FreeBSD-CURRENT, e depois reintroduzida no FreeBSD-STABLE entre a 2.2.2 e 2.2.5) o `target` `world` foi dividido em dois: `buildworld` e `installworld`.

Como implicam os nomes, `buildworld` compila uma nova árvore debaixo de `/usr/obj`, e `installworld` instala esta árvore na máquina.

Isto é muito útil por duas razões. Primeiro, permite que você faça uma recompilação segura, sabendo que os componentes do sistema que estão em execução não serão afetados. A compilação é `self hosted`. Por causa disso, você pode seguramente executar o `buildworld` em uma máquina em execução em modo multi usuário sem medo de efeitos colaterais. Entretanto, ainda é recomendável que você execute o `installworld` em modo mono usuário.

Segundo, permite que você use NFS para atualizar múltiplas máquinas na sua rede. Se você possui três máquinas, A, B e C que você quer atualizar, execute o `make buildworld` e `make installworld` no A. B e C devem montar através de NFS o `/usr/src` e `/usr/obj` do A, e então você pode executar o `make installworld` para instalar o resultado da compilação nas máquinas B e C.

Embora o `target world` ainda exista, você é fortemente encorajado a não usá-lo.

Execute

```
# make buildworld
```

Agora é possível especificar a opção `-j` para o `make`, que fará com que ele gere vários processos simultâneos. Isto é mais útil em máquinas com múltiplas CPUs. Entretanto, uma vez que muito do processo de compilação utiliza mais IO do que CPU, isto também é útil em máquinas com apenas um processador.

Em uma máquina típica de uma CPU você deve executar:

```
# make -j4 buildworld
```

[make\(1\)](#) terá 4 processos rodando ao mesmo tempo. Evidências empíricas postadas nas listas de discussão mostram que isto resulta em benefícios de melhora de performance.

Se você tem uma máquina com vários procesadores e está usando um `kernel` configurado para SMP, tente valores entre 6 e 10 e veja como eles aumentam a velocidade.

Saiba que isto é um tanto quando experimental, e envio de novos códigos para a árvore de fontes pode causar um não funcionamento desta característica. Se a compilação do sistema falhar ao usar este parâmetro tente novamente sem ele antes de reportar problemas.

21.4.6.3. Tempos

Muitos fatores influenciam no tempo de compilação, mas atualmente um Pentium® III com 128 MB de RAM leva aproximadamente duas horas para compilar a árvore FreeBSD-STABLE, sem o uso de atalhos ou truques durante o processo. Uma árvore do FreeBSD-CURRENT levará um pouco mais de tempo.

21.4.7. Compile e Instale um Novo Kernel

Para aproveitar ao máximo seu novo sistema você deve recompilar o `kernel`. Isto é praticamente uma necessidade, uma vez que certas estruturas de memórias mudaram, e programas como o [ps\(1\)](#) e [top\(1\)](#) não funcionarão até que as versões do `kernel` e dos fontes sejam as mesmas.

A maneira mais simples e segura de fazer isto é recompilar e instalar o `kernel` baseado no arquivo `GENERIC`. Enquanto o `GENERIC` não contém todos os dispositivos necessários para o seu sistema, ele deve conter tudo que for necessário para inicializar de volta seu

sistema em modo mono usuário. Isto é um bom teste para ver se o sistema funciona corretamente. Após inicializar a partir do `GENERIC` e verificar que seu sistema funciona então você pode recompilar um novo *kernel* baseado no seu arquivo de configuração.

Se você está atualizando para o FreeBSD 4.0 ou superior então o procedimento antigo de compilação do kernel (como descrito em [Capítulo 9](#)) está defasado. Ao invés desse você deve executar estes comandos *depois* de ter [compilado o sistema com o](#) `buildworld`.

Nota: Se você quer compilar um *kernel* customizado e já possuir um arquivo de configuração, apenas use `KERNCONF=MYKERNEL` desta forma:

```
# cd /usr/src
# make buildkernel KERNCONF=MYKERNEL
# make installkernel KERNCONF=MYKERNEL
```

No FreeBSD 4.2 e anteriores você deve substituir `KERNCONF=` por `KERNEL=`. 4.2-STABLE anterior a 2 de Fevereiro de 2001 não reconhece `KERNCONF=`.

Note que, se você subiu o `kern.securelevel` acima de 1 e configurou o atributo `noschg` ou similar no seu arquivo binário de *kernel*, será necessário ir para o modo mono usuário para usar o `installkernel`. Caso contrário, você deverá ser capaz de executar ambos comandos no modo multi usuário sem problemas. Veja [init\(8\)](#) para detalhes a respeito do `kern.securelevel` e [chflags\(1\)](#) para detalhes sobre as várias opções de parâmetros de arquivos.

Se você está atualizando para uma versão do FreeBSD inferior à 4.0 você deve usar o procedimento antigo de compilação do *kernel*. Entretanto, é recomendado que você use a nova versão do [config\(8\)](#), usando uma linha de comando como esta.

```
# /usr/obj/usr/src/usr.sbin/config/config KERNELNAME
```

21.4.8. Reinicialize no Modo Mono Usuário

Você deve inicializar no modo monousuário para testar o funcionamento do novo kernel. Faça isto seguindo as instruções descritas em [Seção 21.4.4](#).

21.4.9. Instale os Novos Binários do Sistema

Se você estiver compilando uma versão do FreeBSD recente o bastante para ter usado o `make buildworld`, então você deve usá-lo o `installworld` para instalar os novos binários do sistema.

Execute

```
# cd /usr/src
# make installworld
```

Nota: Se você especificou variáveis na linha do comando `make buildworld`, você deve especificar as mesmas variáveis na linha do comando `make installworld`. Isto não é

necessariamente verdadeiro para algumas opções; por exemplo, a opção `-j` nunca deve ser usada com o `installworld`.

Por exemplo, se você executou:

```
# make -DNOPROFILE buildworld
```

você deve instalar o resultado com:

```
# make -DNOPROFILE installworld
```

señão o sistema tentará instalar bibliotecas que não foram compiladas durante o comando `make buildworld`.

21.4.10. Atualize os Arquivos não Atualizados pelo `make world`

Recompilar todo o sistema não atualizará certos diretórios (em particular, `/etc`, `/var` e `/usr`) com arquivos de configuração novos ou modificados.

A maneira mais simples de atualizar estes arquivos é usar o [mergemaster\(8\)](#), embora seja possível fazer isto manualmente se você preferir. Independente da maneira que você escolha, esteja certo de ter uma cópia de segurança do diretório `/etc` caso algo saia errado.

21.4.10.1. `mergemaster`

Contribuição de Tom Rhodes.

O utilitário [mergemaster\(8\)](#) é um *script* Bourne que ajudará você a determinar as diferenças entre seus arquivos de configuração em `/etc`, e os arquivos de configuração na árvore dos fontes em `/usr/src/etc`. Esta é a solução recomendada para manter os arquivos de configuração do sistema atualizados com os que estão na árvore dos fontes.

O `mergemaster` foi integrado ao sistema base do FreeBSD entre o 3.3-RELEASE e 3.4-RELEASE, o que significa que estava presente em todos os -STABLE e -CURRENT deste o 3.3.

Para começar simplesmente digite `mergemaster` na sua linha de comando e observe. `mergemaster` compilará um ambiente temporário de root, a partir de `/`, e colocará lá vários arquivos de configuração do sistema. Neste ponto, os arquivos diferentes serão mostrados pelo formato do [diff\(1\)](#), com o sinal de `+` representado linhas adicionadas ou modificadas, e o sinal `-` representado linhas que foram removidas completamente, ou substituídas por uma nova linha. Veja a página de manual de [diff\(1\)](#) para mais informações sobre a sintaxe do [diff\(1\)](#) e como as diferenças entre os arquivos são mostradas.

[mergemaster\(8\)](#) mostrará a você cada arquivo que possuem diferenças, e neste ponto você terá a opção de apagar o arquivo novo (referenciado como um arquivo temporário), instalar o arquivo temporário em seu estado não modificado, unir o arquivo

temporário com o arquivo atualmente instalado, ou visualizar as diferenças geradas pelo [diff\(1\)](#) novamente.

Escolher pela remoção do arquivo temporário instruirá o [mergemaster\(8\)](#) a manter nosso arquivo atual intacto, e apagar a nova versão. Esta opção não é recomendada, a menos que você não veja razões para modificar o arquivo atual. Você pode obter ajuda a qualquer momento digitando `?` no prompt do [mergemaster\(8\)](#). Se o usuário quiser pular o arquivo, o mesmo será mostrado novamente depois que todos os arquivos tenham sido tratados.

Optar por instalar o arquivo temporário não modificado substituirá o arquivo atual com o novo. Para arquivos não modificados, esta é a melhor opção.

Optar por unir os arquivos mostrará um editor de textos, e o conteúdo dos dois arquivos. Você pode uni-los revisando-os lado a lado na tela, escolhendo partes de cada um para obter um produto final. Quando os arquivos são comparados lado a lado, a tecla `l` selecionará o conteúdo da esquerda e a tecla `r` da sua direita. A saída final será um arquivo formado por ambas as partes, que pode ser instalado. Esta opção normalmente é usada para arquivos que foram modificados pelo usuário.

Optar por visualizar novamente os resultados do [diff\(1\)](#) lhe mostrará as diferenças assim como o [mergemaster\(8\)](#) fez antes de lhe solicitar uma opção.

Depois de terminar os arquivos de sistema o [mergemaster\(8\)](#) lhe mostrará outras opções. Perguntará se você deseja recompilar o arquivo de senhas e/ou executar o [MAKEDEV\(8\)](#) se você estiver executando o FreeBSD anterior a versão 5.0, e terminará com a opção de remover os arquivos temporários que ficaram para trás.

21.4.10.2. Atualização Manual

Se você deseja atualizar manualmente, você não pode simplesmente sobrescrever os arquivos de `/usr/src/etc` para `/etc` e esperar que funcione. Alguns destes arquivos devem ser "instalados" primeiro. Isto porque o diretório `/usr/src/etc` não é uma cópia do que está em seu `/etc`. Além disso, existem arquivos que podem estar em `/etc` e não estar em `/usr/src/etc`.

Se você estiver usando o [mergemaster\(8\)](#) (como recomendado), você pode pular para [próxima seção](#).

A maneira mais simples de fazer isto manualmente é instalar os arquivos em um novo diretório, e trabalhar neles procurando por diferenças.

Faça uma Cópia de Segurança do seu `/etc`Embora, teoricamente, nada vá tocar em neste diretório automaticamente, é sempre melhor estar prevenido. Então copie seu diretório `/etc` para um lugar seguro. Algo do tipo:

```
# cp -Rp /etc /etc.old
```

A opção `-R` faz uma cópia recursiva, a opção `-p` preserva os tempos, propriedades dos arquivos e coisas do tipo.

Você precisa ter alguns diretórios temporários para instalar o novo `/etc` nestes diretórios. `/var/tmp/root` é uma boa escolha, e existe vários diretórios necessários abaixo dele.

```
# mkdir /var/tmp/root
# cd /usr/src/etc
# make DESTDIR=/var/tmp/root distrib-dirs distribution
```

Isto criará a estrutura de diretórios necessária e instalará os arquivos. Muitos subdiretórios que foram criados abaixo de `/var/tmp/root` estarão vazios e podem ser apagados. A maneira mais simples de se fazer isto é:

```
# cd /var/tmp/root
# find -d . -type d | xargs rmdir 2>/dev/null
```

Isto removerá os diretórios vazios. (O erro padrão é redirecionado para `/dev/null` para evitar que alertas a respeito de diretórios vazios sejam exibidos.)

`/var/tmp/root` agora contém todos os arquivos que devem ser colocados em locais apropriados abaixo de `.`. Você agora deve ir em cada um destes arquivos, determinando como eles diferem dos que estão no seu sistema.

Note que alguns dos arquivos que foram instalados em `/var/tmp/root` possuem um ```.` no início. Na hora da escrita os únicos arquivos como estes são arquivos de inicialização de linha de comando em `/var/tmp/root/` e `/var/tmp/root/root/`, embora possam existir outros (dependendo de quando você estiver lendo isto). Use o comando `ls -a` para pegá-los.

A maneira mais simples de se fazer isso é usar o [diff\(1\)](#) para comparar dois arquivos:

```
# diff /etc/shells /var/tmp/root/etc/shells
```

Isto exibirá as diferenças entre o seu arquivo `/etc/shells` e o novo `/var/tmp/root/etc/shells`. Assim você pode decidir por incluir as mudanças que você fez ou substituir seu arquivo antigo.

Nomeie o novo diretório Root (`/var/tmp/root`) com um Novo Selo de Hora, Assim Você Pode Facilmente Comparar as Diferenças Entre as Versões: Recompilar o sistema frequentemente significa ter que atualizar o `/etc` frequentemente também, que pode ser uma tarefa um pouco chata.

Você pode acelerar este processo mantendo um cópia do último conjunto de arquivos modificados que você fez a junção em `/etc`. O seguinte procedimento lhe dá uma idéia de como fazer isto.

1. Compile o sistema normalmente. Quando você quiser atualizar o `/etc` e outros diretórios, dê ao diretório de destino um nome baseado na data corrente. Se você estiver fazendo isto em 14 de fevereiro de 1998 você pode fazer o seguinte:
2. `# mkdir /var/tmp/root-19980214`
3. `# cd /usr/src/etc`
4. `# make DESTDIR=/var/tmp/root-19980214 \`

5. `distrib-dirs distribution`
6. Faça a junção destes diretórios como descrito abaixo.

Não remova o diretório `/var/tmp/root-19980214` quando você terminar.

7. Após ter pego e compilado a última versão dos fontes, siga o passo 1. Isto lhe dará um novo diretório, que poderá ser chamado `/var/tmp/root-19980221` (se você esperar uma semana para fazer a atualização).
8. Agora você pode ver as diferenças que foram feitas usando o [diff\(1\)](#) para criar um arquivo de diferenças recursivamente entre os dois diretórios:
9. `# cd /var/tmp`
10. `# diff -r root-19980214 root-19980221`

Tipicamente, será um conjunto de diferenças bem menor do que as diferenças entre `/var/tmp/root-19980221/etc` e `/etc`. Pelo fato das diferenças serem menores, é mais fácil migrar as mudanças no seu diretório `/etc`.

11. Agora você pode remover o mais antigo dos dois diretórios `/var/tmp/root-*`:
12. `# rm -rf /var/tmp/root-19980214`
13. Repita este processo sempre que precisar unir as mudanças para o `/etc`.

Você pode usar o [date\(1\)](#) para automatizar a geração de nomes de diretórios:

```
# mkdir /var/tmp/root-`date +%Y%m%d`
```

21.4.11. Atualize o `/dev`

Nota: Se você estiver executando o FreeBSD 5.0 ou superior você pode pular esta seção. Estas versões usam o [devfs\(5\)](#) para alocar nós de dispositivos transparentemente para o usuário.

Na maioria dos casos, a ferramenta [mergemaster\(8\)](#) saberá quando for necessário atualizar os nós de dispositivo, e se oferecerá para completar automaticamente a tarefa. Estas instruções lhe dirão como atualizar os nós de dispositivo manualmente.

Em nome da segurança, este é um processo passo a passo.

1. Copie o `/var/tmp/root/dev/MAKEDEV` para `/dev`:
2. `# cp /var/tmp/root/dev/MAKEDEV /dev`

Se você usou o [mergemaster\(8\)](#) para atualizar o `/etc`, então o seu *script* `MAKEDEV` já pode ter sido atualizado, embora não custe nada verificar (com [diff\(1\)](#)) e copiá-lo manualmente, se necessário.

3. Agora, tire uma fotografia do seu `/dev` atual. Esta foto precisa ter as permissões, donos, números de cada arquivo, mas não pode conter os selos de tempo. A maneira mais fácil de fazer isto é usar o [awk\(1\)](#) para retirar alguma informação:
4. `# cd /dev`
5. `# ls -l | awk '{print $1, $2, $3, $4, $5, $6, $NF}' > /var/tmp/dev.out`
6. Recompile todos os nós de dispositivo:
7. `# sh MAKEDEV all`

8. Tire outra foto do diretório, desta vez em `/var/tmp/dev2.out`. Agora olhe estes dois arquivos e procure por nós de dispositivos que você deixou de criar. Não deve ter nenhum, mas é melhor prevenir do que remediar.
9.

```
# diff /var/tmp/dev.out /var/tmp/dev2.out
```

Você está mais sujeito a ver discrepâncias de fatias de disco que envolvem comandos como:

```
# sh MAKEDEV sd0s1
```

para recriar as entradas de fatias de disco. As circunstâncias podem variar.

21.4.12. Atualize o `/stand`

Nota: Este fato é incluído apenas como complemento. Pode ser omitido seguramente. Se você está usando o FreeBSD 5.2 ou superior, o diretório `/rescue` é automaticamente atualizado para usuário com os binários atuais, compilados estaticamente durante o `make installworld`, tornando obsoleta a necessidade de atualizar o `/stand`.

Em nome da finalização, você pode querer atualizar os arquivos em `/stand`. Estes arquivos consistem em *links* para o binário `/stand/sysinstall`. Este binário deve ser ligado estaticamente, para que possa funcionar quando nenhum outro sistema de arquivos (e em particular o `/usr`) foi montado.

```
# cd /usr/src/release/sysinstall
# make all install
```

21.4.13. Reinicializando

Você está pronto. Após ter verificado que tudo parece estar em seu devido lugar você pode reinicializar o sistema. Um simples [shutdown\(8\)](#) fará isto:

```
# shutdown -r now
```

21.4.14. Finalizado

Agora você deve ter seu sistema FreeBSD atualizado. Parabéns.

Se as coisas deram um pouco errado, é fácil recompilar uma parte particular do sistema. Por exemplo, se você apagou acidentalmente o `/etc/magic` como parte da atualização ou junção do `/etc`, o comando [file\(1\)](#) vai parar de funcionar. Neste caso, o conserto seria executar:

```
# cd /usr/src/usr.bin/file
# make all install
```

21.4.15. Questões

21.4.15.1. [Eu preciso recompilar o sistema para toda mudança?](#)

21.4.15.2. [Minha compilação falhou com muitos `signal 11` \(ou outro número de sinal\). O que aconteceu?](#)

21.4.15.3. [Posso remover o /usr/obj quando eu terminar?](#)

21.4.15.4. [Compilações interrompidas podem ser continuadas?](#)

21.4.15.5. [Como eu posso acelerar a compilação do sistema?](#)

21.4.15.6. [O que eu faço se algo der errado?](#)

21.4.15.1. Eu preciso recompilar o sistema para toda mudança?

Não existe resposta fácil para esta, depende da natureza da mudança. Por exemplo, se você apenas executou o **CVSup**, e a saída foi a atualização dos seguintes arquivos:

```
src/games/cribbage/instr.c
src/games/sail/pl_main.c
src/release/sysinstall/config.c
src/release/sysinstall/media.c
src/share/mk/bsd.port.mk
```

provavelmente não merece recompilar todo o sistema. Você pode apenas ir aos subdiretórios apropriados e executar `make all install`. Mas se algo maior foi modificado, por exemplo `src/lib/stdlib`, então você deve recompilar todo o sistema ou pelo menos as partes que são estaticamente ligadas (como qualquer coisa que você pode ter adicionado que é ligado estaticamente).

No fim do dia, é por sua conta. Você pode estar feliz em recompilar o sistema toda noite, e deixar as mudanças acumularem por esta noite. Ou você pode querer recompilar apenas aquelas coisas que foram modificadas, e ter certeza de que você pode detectar todas as dependências.

E, é claro, isto tudo depende da frequência na qual você deseja atualizar, e qual versão você está, `FreeBSD-STABLE` ou `FreeBSD-CURRENT`.

21.4.15.2. Minha compilação falhou com muitos *signal 11* (ou outro número de sinal). O que aconteceu?

Isto normalmente é um indicativo de problemas de hardware. (Re)compilar o sistema é uma maneira efetiva de testar seu hardware, e frequentemente mostrará problemas de memória. Estes problemas normalmente se manifestarão, com o compilador morrendo misteriosamente na recepção de sinais estranhos.

Um indicador certo disto é que se você pode reinicializar a compilação e ela volta a falhar em um ponto diferente do processo.

Nesta instância pouco você pode fazer, exceto começar a trocar componentes da sua máquina para determinar qual está falhando.

21.4.15.3. Posso remover o `/usr/obj` quando eu terminar?

A resposta curta é sim.

`/usr/obj` contém todos os arquivos objeto que foram produzidos durante a fase de compilação. Normalmente, um dos primeiros passos do processo do ```make world``` é remover este diretório e começar do zero. Neste caso, manter o `/usr/obj` por perto depois de ter acabado faz algum sentido, e vai liberar um bom espaço em disco (atualmente em torno de 340 MB).

Entretanto, se você souber o que está fazendo, você pode fazer com que o ```make world``` pule este passo. Isto fará com que as compilações subsequentes sejam executadas mais rápido, uma vez que a maioria dos fontes não precisarão ser recompilados. O outro lado disso é que problemas subitos de dependência podem ocorrer, fazendo com que sua compilação falhe de maneira estranha. Isto frequentemente gera barulho nas listas de discussão do FreeBSD, quando uma pessoa reclama que sua compilação falhou, sem saber que a causa foi a tentativa de cortar caminho.

21.4.15.4. Compilações interrompidas podem ser continuadas?

Isto depende quão longe você chegou no processo até ter encontrado um problema.

Em geral (e isto não é uma regra rígida) o processo ```make world``` compila novas cópias de ferramentas essenciais (tais como [gcc\(1\)](#) e [make\(1\)](#)) e bibliotecas de sistema. Estas ferramentas e bibliotecas são então instaladas. Elas são usadas para compilarem a si próprias, e são instaladas novamente. Todo o sistema, (agora incluindo programas regulares de usuário, tais como [ls\(1\)](#) ou [grep\(1\)](#)) é então recompilado com os novos arquivos de sistema.

Se você está no último estágio, e você sabe disto (pois você olhou na saída do comando que você está guardando), então você pode (de forma consideravelmente segura) fazer:

```
... fix the problem ...
# cd /usr/src
# make -DNOCLEAN all
```

Isto não desfará o trabalho do ```make world``` anterior.

Se você ver a mensagem:

```
-----
Building everything..
-----
```

na saída do ```make world```, então provavelmente é seguro fazer isto.

Se você não ver esta mensagem, ou não tiver certeza, então é sempre melhor prevenir do que remediar, e reiniciar a compilação do zero.

21.4.15.5. Como eu posso acelerar a compilação do sistema?

- Execute em modo mono usuário.
- Coloque os diretórios `/usr/src` e `/usr/obj` em sistemas de arquivos em discos separados. Se possível, coloque estes discos em controladoras separadas.

- Melhor ainda, coloque estes sistemas de arquivo distribuídos em múltiplos discos usando o dispositivo [ccd\(4\)](#) (driver de disco concatenado).
- Desligue o *profiling* (set ```NOPROFILE=true``` no `/etc/make.conf`). É quase certo que você não vá precisar disto.
- Também no `/etc/make.conf`, configure `CFLAGS` para algo como `-O -pipe`. A otimização `-O2` é mais lenta, e a diferença de otimização entre a opção `-O` e `-O2` normalmente não é notada. A opção `-pipe` faz com que o compilador use *pipes* ao invés de usar arquivos temporários para comunicação, evitando acessos ao disco (por conta da memória).
- Passe a opção `-jn` para o [make\(1\)](#) para executar múltiplos processos em paralelo. Isto geralmente ajuda, não importando se sua máquina tem um ou vários processadores.
- O sistema de arquivos que hospedar o `/usr/src` pode ser montado (ou desmontado) com a opção *noatime*. Isto evita que o sistema de arquivos faça gravações durante o acesso à arquivo. Você provavelmente não precisa desta informação.
- ```
mount -u -o noatime /usr/src
```

**Atenção** Este exemplo assume que `/usr/src` está em seu próprio sistema de arquivos. Se não estiver (se for parte do `/usr`, por exemplo) então você não precisará usar este ponto de montagem para este sistema de arquivos.

- O sistema de arquivos onde `/usr/obj` estiver pode ser montado (ou remontado) com a opção *async*. Isto faz com que a escrita em disco aconteça assíncronamente. Em outras palavras, a escrita é completada imediatamente, e os dados são escritos no disco poucos segundos depois. Isto permite que as escritas sejam agupadas, e pode causar um grande ganho de performance.

**Atenção** Tenha em mente que esta opção faz com que seu sistema de arquivos fique mais frágil. Com esta opção aumentam as chances que, em uma falha de energia, o sistema de arquivos fique em um estado irrecoverável quando a máquina reinicializar.

Se `/usr/obj` é o único neste sistema de arquivos então isto não é problema. Se você tem outros dados de valor no mesmo sistema de arquivos, certifique-se que suas cópias de segurança estão recentes antes de habilitar esta opção.

```
mount -u -o async /usr/obj
```

**Atenção** Como acima, se `/usr/obj` não estiver em seu próprio sistema de arquivos, substitua pelo nome apropriado de ponto de montagem.

#### 21.4.15.6. O que eu faço se algo der errado?

Esteja absolutamente certo de que seu ambiente não experimentou anomalias nas compilações anteriores. Isto é o bastante.

```
chflags -R noschg /usr/obj/usr
rm -rf /usr/obj/usr
cd /usr/src
make cleandir
```

```
make cleandir
```

Sim, o comando `make cleandir` realmente deve ser executado duas vezes.

Então reinicie todo o processo, começando com o comando `make buildworld`.

Se você ainda tiver problemas, envie o erro e a saída do comando `uname -a` para [lista de discussão FreeBSD de perguntas genéricas](#). Esteja preparado para responder perguntas a respeito da sua configuração!

---

## 21.5. Preparando Múltiplas Máquinas

*Contribuição de Mike Meyer.*

Se você possui múltiplas máquinas que você quer manter a mesma árvore de fontes, então fazer com que todas façam download dos fontes e recompilem tudo parece ser uma perda de recursos: espaço em disco, banda de rede, e ciclos de CPU. E é, e uma idéia é possuir uma máquina para fazer a maior parte do trabalho, enquanto o resto das máquinas pegam este trabalho através de NFS. Esta seção expõe um método para fazer isto.

### 21.5.1. Preliminares

Primeiro, identifique um conjunto de máquinas que executarão o mesmo conjunto de binários, as quais chamaremos de *conjunto de compilação*. Cada máquina pode possuir um *kernel* customizado, mas elas estarão executando os mesmos binários de usuário. Deste conjunto, escolha uma máquina para ser a *máquina de compilação*. Será a máquina onde os binários de sistema e o *kernel* serão compilados. Idealmente, deve ser uma máquina rápida com folga para executar o comando `make world`. Você também desejará escolher uma *máquina de testes*, para testar as atualizações de programas antes de colocá-las em produção. Esta *deve* ser uma máquina que pode permanecer fora do ar por grandes períodos de tempo. Pode ser a máquina de compilação, mas não necessariamente.

Todas as máquinas neste conjunto de compilação precisam montar o `/usr/obj` e `/usr/src` da mesma máquina, e no mesmo ponto. Idealmente, estes são dois *drives* diferentes na máquina de compilação, mas eles podem ser montados via NFS nessa máquina também. Se você possui múltiplos conjuntos de compilação, o `/usr/src` deve estar na máquina de compilação, e montado através de NFS no resto.

Finalmente esteja certo que o arquivo `/etc/make.conf` em todas as máquinas do conjunto de compilação estejam de acordo com a máquina de compilação. Isto significa que a máquina de compilação deve ter todas as partes do sistema base que qualquer máquina do conjunto de instalação vá instalar. Além disso, cada máquina deve ter seu nome de *kernel* configurado através da variável `KERNCONF` em `/etc/make.conf` e a máquina de compilação deve listar todas em `KERNCONF`, listando seu próprio *kernel*

primeiro. A máquina de compilação deve ter os arquivos de configuração de *kernel* para cada máquina em `/usr/src/sys/arch/conf` se for compilar seus *kernels*.

### 21.5.2. O Sistema Base

Agora que tudo está feito, você está pronto para compilar tudo. Compile o *kernel* e o sistema como descrito em [Seção 21.4.6.2](#) na máquina de compilação, mas não instale nada. Depois que a compilação terminar, vá para a máquina de teste e instale o *kernel* que você acabou de compilar. Se esta máquina monta os diretórios `/usr/src` e `/usr/obj` via NFS, quando você reinicializar no modo mono usuário, você precisará habilitar a rede e montá-los. A maneira mais fácil de fazer isto é reinicializar no modo multi usuário e executar o comando `shutdown now` para ir para o modo mono usuário. Uma vez lá, você pode instalar o novo *kernel* e sistema e executar o `mergemaster` como você faz normalmente. Ao terminar, reinicialize e volte às operações normais de multi usuário desta máquina.

Após estar certo de que tudo na máquina de teste está funcionando apropriadamente, use o mesmo procedimento para instalar em cada uma das máquinas do conjunto de compilação.

### 21.5.3. Ports

A mesma idéia pode ser usada na árvore de aplicações portadas. O primeiro passo crítico é montar o diretório `/usr/ports` a partir da mesma máquina para todas as do conjunto de compilação. Você pode então configurar o arquivo `/etc/make.conf` apropriadamente para compartilhar os *distfiles*. Você deve configurar a variável `DISTDIR` para um diretório comum compartilhado com permissão de escrita para qualquer usuário `root` esteja conectado aos diretórios montados via NFS. Cada máquina deve configurar a variável `WRKDIRPREFIX` para um diretório local de compilação. Finalmente, se você for compilar e distribuir pacotes, você deve configurar a variável `PACKAGES` para um diretório similar ao da variável `DISTDIR`.

## Capítulo 22. Compatibilidade Binária Linux

### Índice

- 22.1. [Sinopse](#)
- 22.2. [Instalação](#)
- 22.3. [Instalando o Mathematica®](#)
- 22.4. [Instalando o Maple™](#)
- 22.5. [Instalando o MATLAB®](#)
- 22.6. [Instalando Oracle®](#)
- 22.7. [Instalando o SAP® R/3®](#)
- 22.8. [Tópicos Anvaçados](#)

*Reestruturado e partes atualizadas por Jim Mock. Originalmente contribuído por Brian N. Handy e Rich Murphey.*



## 22.1. Sinopse

FreeBSD oferece compatibilidade binária com vários outros sistemas operacionais UNIX® incluindo Linux. Neste momento você deve estar perguntando a si mesmo, por que o FreeBSD necessita ser capaz de executar binários Linux? A resposta para esta questão é bem simples. Muitas empresas e desenvolvedores desenvolvem apenas para Linux, uma vez que ele é a novidade "mais quente" no mundo da computação. Isto nos deixa, usuários do FreeBSD, incomodando as mesmas empresas e desenvolvedores para disponibilizar versões nativas de suas aplicações para FreeBSD. O problema é, a maioria destas empresas não se dá conta de que muitas pessoas usariam seus produtos se existisse uma versão para FreeBSD também, e muitas continuam desenvolvendo apenas para Linux. Então o que o usuário do FreeBSD faz? Aí é que a Compatibilidade binária Linux do FreeBSD entra em cena.

Resumidamente, a compatibilidade permite aos usuários do FreeBSD executar por volta de 90% de todas as aplicações Linux sem modificação. Isto inclui aplicações tais como **StarOffice™**, a versão Linux do **Netscape®**, **Adobe® Acrobat®**, **RealPlayer® 5 e 7**, **VMware™**, **Oracle®**, **WordPerfect®**, **Doom**, **Quake**, e mais. É também sabido que em algumas situações, binários do Linux têm melhor performance no FreeBSD do que no próprio Linux.

Existem, entretanto, algumas características de sistema operacional específicas do Linux que não são suportadas no FreeBSD. Binários Linux não funcionarão no FreeBSD se eles usarem excessivamente o sistema de arquivos `/proc` (que é diferente do sistema de arquivo `/proc` do FreeBSD), ou chamadas i386™ específicas, tais como a habilitação do modo virtual 8086.

Após ler este capítulo, você saberá:

- Como habilitar a compatibilidade binária Linux em seu sistema.
- Como adicionar bibliotecas compartilhadas Linux adicionais.
- Como instalar aplicações Linux em seu sistema FreeBSD.
- Detalhes de implementação da compatibilidade Linux no FreeBSD.

Antes de ler este capítulo, você deve:

- Saber como instalar programas adicionais de terceiros ([Capítulo 4](#)).

## 22.2. Instalação

A compatibilidade Linux não é ligada por padrão. A maneira mais fácil de habilitar esta funcionalidade é carregando o objeto KLD `linux` ("Kernel Loadable object"). Você pode carregar este módulo simplesmente digitando `linux` na linha de comando.

Se você quer que a compatibilidade Linux esteja sempre habilitada, então você deve adicionar a seguinte linha no `/etc/rc.conf`:

```
linux_enable="YES"
```

O comando [kldstat\(8\)](#) pode ser usado para verificar se o KLD está carregado:

```
% kldstat
Id Refs Address Size Name
 1 2 0xc0100000 16bdb8 kernel
 7 1 0xc24db000 d000 linux.ko
```

Se, por alguma razão, você não quer ou não pode carregar o KLD, então você pode ligar estaticamente a compatibilidade binária Linux no kernel adicionando *options LINUX* ao seu arquivo de configuração. Depois instale seu novo *kernel* como como descrito em [Capítulo 9](#).

### 22.2.1. Instalando Bibliotecas Linux de Tempo de Execução

Isto pode ser feito de duas maneiras, usando o port [linux\\_base](#), ou instalando as bibliotecas [manualmente](#).

#### 22.2.1.1. Instalando o port linux\_base

Este é de longe o método mais fácil para instalar as bibliotecas de tempo de execução. É como instalar qualquer outra aplicação da árvore de aplicações [ports collection](#). Simplesmente faça o seguinte:

```
cd /usr/ports/emulators/linux_base
make install distclean
```

Agora você deve ter a compatibilidade binária Linux funcionando. Alguns programas podem reclamar a respeito de pequenas diferenças de versão das bibliotecas de sistema. Em geral, entretanto, isto não parece ser problema.

**Nota:** Podem existir múltiplas versões do port [emulators/linux\\_base](#) disponíveis, correspondendo a diferentes versões de várias distribuições Linux. Você deve instalar a versão que mais se aproxima dos requisitos das aplicações Linux que você deseja instalar.

#### 22.2.1.2. Instalando Manualmente as Bibliotecas

Se você não possui a coleção ``ports'' instalada, você pode instalar as mesmas manualmente. Você precisará das bibliotecas compartilhadas Linux das quais o programa depende e um ligador em tempo de execução. Além disso, você precisa criar um diretório ``shadow root'', `/compat/linux`, para as bibliotecas Linux no seu sistema FreeBSD. Qualquer biblioteca compartilhada aberta por aplicações Linux sob o FreeBSD buscarão este diretório primeiro. Sendo assim, se um programa Linux carrega, por exemplo, o arquivo `/lib/libc.so`, o FreeBSD primeiro vai tentar abrir `/compat/linux/lib/libc.so`, e se não existir, então tentará `/lib/libc.so`. Bibliotecas compartilhadas devem ser instaladas na árvore `/compat/linux/lib` ao invés dos caminhos que a biblioteca Linux `ld.so` reporta.

Geralmente, você precisará procurar pelas bibliotecas compartilhadas das quais as bibliotecas Linux dependem apenas nas primeiras vezes que você instalar um programa

Linux no seu sistema FreeBSD. Depois de um tempo, você possuirá um conjunto suficiente de bibliotecas Linux compartilhadas no seu sistema que serão capazes de executar binários Linux recentemente importados sem qualquer trabalho extra.

### 22.2.1.3. Como Instalar Bibliotecas Compartilhadas Adicionais

E se você instalar o `linux_base` e sua aplicação ainda reclamar que faltam bibliotecas compartilhadas? Como você sabe quais bibliotecas compartilhadas os binários Linux precisam, e onde obtê-las? Basicamente, existem duas possibilidades (ao seguir estas instruções você precisará estar como `root` no seu sistema FreeBSD).

Se você tem acesso a um sistema Linux, veja quais bibliotecas compartilhadas a aplicação necessita, e copie-as para o seu sistema FreeBSD. Veja o seguinte exemplo:

Vamos assumir que você usou FTP para obter o binário Linux do **Doom**, e colocou em um sistema Linux ao qual você tem acesso. Você então pode verificar quais bibliotecas compartilhadas ele precisa executando `ldd linuxdoom`, como:

```
% ldd linuxdoom
libXt.so.3 (DLL Jump 3.1) => /usr/X11/lib/libXt.so.3.1.0
libX11.so.3 (DLL Jump 3.1) => /usr/X11/lib/libX11.so.3.1.0
libc.so.4 (DLL Jump 4.5p126) => /lib/libc.so.4.6.29
```

Você precisará pegar todos os arquivos na última coluna e colocá-los em `/compat/linux`, com os nomes da primeira coluna como ligações simbólicas apontando para eles. Isto significa que eventualmente tem estes arquivos em seu sistema FreeBSD:

```
/compat/linux/usr/X11/lib/libXt.so.3.1.0
/compat/linux/usr/X11/lib/libXt.so.3 -> libXt.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3 -> libX11.so.3.1.0
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

**Nota:** Note que se você já possui uma biblioteca Linux compartilhada em que as versões principais batem com a da primeira coluna da saída do comando `ldd`, você não precisará copiar o arquivo com o nome da última coluna para o seu sistema, a versão que você tem deve funcionar. Entretanto, é aconselhável copiar a biblioteca compartilhada se a versão for mais nova. Você pode remover a antiga assim que você fizer o link simbólico apontando para a mais nova. Então, se você possuir estas bibliotecas em seu sistema:

```
/compat/linux/lib/libc.so.4.6.27
/compat/linux/lib/libc.so.4 -> libc.so.4.6.27
```

e você tem um novo binário que requer a última versão, de acordo com a saída do `ldd`:

```
libc.so.4 (DLL Jump 4.5p126) -> libc.so.4.6.29
```

Se for apenas uma ou duas versões atrasadas no dígito final então não se preocupe em copiar o `/lib/libc.so.4.6.29` também, pois o programa deve funcionar bem com a

versão ligeiramente mais antiga. Entretanto, se você quiser, você pode decidir substituir o arquivo `libc.so` e isto fará com que você fique com:

```
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

**Nota:** O mecanismo de ligação simbólica *somente* é necessário para binários Linux. O ligador em tempo de execução do FreeBSD se encarrega de procurar pelos números de versão mais recentes e você não precisa se preocupar.

### 22.2.2. Instalando Binários Linux ELF

Binários ELF algumas vezes requerem um passo extra de "marcação". Se você tentar executar um binário ELF não marcado, você obterá uma mensagem de erro como a seguinte:

```
% ./my-linux-elf-binary
ELF binary type not known
Abort
```

Para ajudar o *kernel* do FreeBSD a distinguir entre um binário ELF do FreeBSD e um binário Linux, use o utilitário [brandelf\(1\)](#).

```
% brandelf -t Linux my-linux-elf-binary
```

O "GNU toolchain" agora coloca a informação de marcação apropriada automaticamente nos binários ELF, de maneira que este passo deve se tornar cada vez mais desnecessário no futuro.

### 22.2.3. Configurando o Resolvedor de nomes de Host

Se o DNS não funciona ou se você recebe esta mensagem:

```
resolv+: "bind" is an invalid keyword resolv+:
"hosts" is an invalid keyword
```

Você precisará configurar o arquivo `/compat/linux/etc/host.conf` com o conteúdo:

```
order hosts, bind
multi on
```

O "order" aqui especifica que a procura é feita primeiro no arquivo `/etc/hosts` e depois no DNS. Quando o arquivo `/compat/linux/etc/host.conf` não está instalado, as aplicações Linux encontram o `/etc/host.conf` do FreeBSD e reclamam da sintaxe incompatível. Você deve remover o *bind* se você não configurou um servidor de nomes usando o arquivo `/etc/resolv.conf`.

## 22.3. Instalando o Mathematica®

*Atualizado para Mathematica 4.X por Murray Stokely. Unido ao trabalho de Bojan Bistrovic.*

Este documento descreve o processo de instalação da versão Linux do **Mathematica® 4.X** em um sistema FreeBSD.

A versão Linux do **Mathematica** roda perfeitamente sob o FreeBSD, entretanto os binários distribuídos pela Wolfram precisam ser marcados para que o FreeBSD saiba que deve usar o Linux ABI para executá-los.

A versão Linux do **Mathematica** ou **Mathematica para Estudantes** pode ser solicitada diretamente em <http://www.wolfram.com/>.

### 22.3.1. Marcando os Binários Linux

Os binários Linux estão localizados no diretório `unix` do CDROM do **Mathematica** distribuído pela Wolfram. Você precisa copiar esta árvore de diretório para o seu disco local para que você possa marcar os binários Linux com [brandelf\(1\)](#) antes de executar o instalador:

```
mount /cdrom
cp -rp /cdrom/Unix/ /localdir/
brandelf -t Linux
/localdir/Files/SystemFiles/Kernel/Binaries/Linux/*
brandelf -t Linux
/localdir/Files/SystemFiles/FrontEnd/Binaries/Linux/*
brandelf -t Linux
/localdir/Files/SystemFiles/Installation/Binaries/Linux/*
brandelf -t Linux
/localdir/Files/SystemFiles/Graphics/Binaries/Linux/*
brandelf -t Linux
/localdir/Files/SystemFiles/Converters/Binaries/Linux/*
brandelf -t Linux
/localdir/Files/SystemFiles/LicenseManager/Binaries/Linux/mathlm
cd /localdir/Installers/Linux/
./MathInstaller
```

Alternativamente, você pode simplesmente configurar a marca ELF Linux padrão para todos os binários não marcados com o comando:

```
sysctl kern.fallback_elf_brand=3
```

Isto fará o FreeBSD assumir que os binários ELF não marcados usam o Linux ABI e assim você deve ser capaz de executar o instalador direto do CDROM.

### 22.3.2. Obtendo sua Senha do Mathematica

Antes que você possa executar o **Mathematica** você precisará obter uma senha da Wolfram, que corresponde a sua ``ID de máquina".

Uma vez instaladas as bibliotecas Linux de compatibilidade em tempo de execução e desempacotado o **Mathematica** você pode obter a "ID de máquina" executando o programa `mathinfo` no diretório de instalação. Esta ID de máquina é baseada somente no endereço MAC da sua primeira placa de Ethernet.

```
cd /localdir/Files/SystemFiles/Installation/Binaries/Linux
mathinfo
disco.example.com 7115-70839-20412
```

Ao obter o registro com a Wolfram, seja por correio eletrônico, telefone ou fax, você fornecerá a eles a "ID da máquina" e eles responderão com uma senha que consiste em um grupo de números. Você pode então entrar com esta informação quando tentar executar o **Mathematica** pela primeira vez, exatamente da mesma maneira que você faria para qualquer outra plataforma do **Mathematica**.

### 22.3.3. Executando o Frontend do Mathematica através da Rede

**Mathematica** usa algumas fontes especiais para mostrar caracteres não presentes em qualquer conjunto padrão de fontes (integrais, somas, letras Gregas, etc.). O protocolo X requer que estas fontes sejam instaladas *localmente*. Isto significa que você precisa copiar estas fontes do CDROM ou de uma máquina com o **Mathematica** instalado para o seu sistema local. Estas fontes normalmente são armazenadas em `/cdrom/Unix/Files/SystemFiles/Fonts` no CDROM ou em `/usr/local/mathematica/SystemFiles/Fonts` em seu disco rígido. As fontes atuais estão em subdiretórios em `Type1` e `X`. Existem várias maneiras de utilizá-las, como descrito abaixo.

A primeira maneira é copiá-las em um dos diretórios de fontes existente em `/usr/X11R6/lib/X11/fonts`. Para isto você precisará editar o arquivo `fonts.dir`, adicionando os nomes das fontes a ele, e modificando o número de fontes na primeira linha. Alternativamente, você também deve ser capaz de executar o [mkfontdir\(1\)](#) no diretório ao qual você as copiou.

A segunda maneira de fazer isto é copiar os diretórios para `/usr/X11R6/lib/X11/fonts`:

```
cd /usr/X11R6/lib/X11/fonts
mkdir X
mkdir MathType1
cd /cdrom/Unix/Files/SystemFiles/Fonts
cp X/* /usr/X11R6/lib/X11/fonts/X
cp Type1/* /usr/X11R6/lib/X11/fonts/MathType1
cd /usr/X11R6/lib/X11/fonts/X
mkfontdir
cd ../MathType1
mkfontdir
```

Agora adicione os novos diretórios de fontes ao seu caminho de fontes:

```
xset fp+ /usr/X11R6/lib/X11/fonts/X
xset fp+ /usr/X11R6/lib/X11/fonts/MathType1
xset fp rehash
```

Se você está usando o servidor **XFree86™**, você pode ter estas fontes carregadas automaticamente adicionando-as ao seu arquivo `XF86Config`.

Se você já *não* possui um diretório chamado `/usr/X11R6/lib/X11/fonts/Type1`, você pode mudar o nome do diretório `MathType1` do exemplo acima para `Type1`.

---

## 22.4. Instalando o Maple™

*Contribuição de Aaron Kaplan. Agradimentos à Robert Getschmann.*

**Maple™** é um programa comercial similar ao **Mathematica®**. Você deve comprar este programa em <http://www.maplesoft.com/> e fazer o registro para obter um arquivo de licença. Para instalar este software no FreeBSD, por favor siga estes passos simples.

1. Execute o *shell script* `INSTALL` da distribuição do produto. Escolha a opção ```RedHat"` quando o programa de instalação perguntar. Um diretório típico de instalação pode ser `/usr/local/maple`.
2. Se você não solicitou, peça uma licença para o **Maple** na Maple Waterloo Software (<http://register.maplesoft.com/>) e copie-a para o arquivo `/usr/local/maple/license/license.dat`.
3. Instale o gerenciador de licenças **FLEXlm** executando o *script shell* de instalação `license manager by running the INSTALL_LIC` que vem junto com o **Maple**. Especifique o nome do servidor primário na sua máquina para o servidor de licença.
4. Corrija o arquivo `/usr/local/maple/bin/maple.system.type` com o seguinte:
5. `----- snip -----`
6. `*** maple.system.type.orig Sun Jul 8 16:35:33 2001`
7. `--- maple.system.type Sun Jul 8 16:35:51 2001`
8. `*****`
9. `*** 72,77 ****`
10. `--- 72,78 ----`
11. `# the IBM RS/6000 AIX case`
12. `MAPLE_BIN="bin.IBM_RISC_UNIX"`
13. `;;`
14. `+ "FreeBSD"|\`
15. `"Linux")`
16. `# the Linux/x86 case`
17. `# We have two Linux implementations, one for Red Hat`
18. `and`
18. `----- snip end of patch -----`

Note que depois do `"FreeBSD"|\` não deve existir espaço em branco

Esta correção instrui o **Maple** a reconhecer o ```FreeBSD"` como um tipo de sistema Linux. O *script shell* `bin/maple` chama o *script shell* `bin/maple.system.type` que por sua vez executa `uname -a` para descobrir o nome do sistema operacional. Dependendo do nome do sistema operacional ele saberá que binários utilizar.

## 19. Inicialize o Servidor de Licenças.

O seguinte *script*, instalado como `/usr/local/etc/rc.d/lmgrd.sh` é uma maneira conveniente de inicializar o `lmgrd`:

```
----- snip -----

#!/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin
PATH=${PATH}:/usr/local/maple/bin:/usr/local/maple/FLEXlm/UNIX/LINUX
export PATH

LICENSE_FILE=/usr/local/maple/license/license.dat
LOG=/var/log/lmgrd.log

case "$1" in
start)
 lmgrd -c ${LICENSE_FILE} 2>> ${LOG} 1>&2
 echo -n " lmgrd"
 ;;
stop)
 lmgrd -c ${LICENSE_FILE} -x lmdown 2>> ${LOG} 1>&2
 ;;
*)
 echo "Usage: `basename $0` {start|stop}" 1>&2
 exit 64
 ;;
esac

exit 0
----- snip -----
```

## 20. Inicialize o Maple:

21. % `cd /usr/local/maple/bin`
22. % `./xmaple`

Ele deve estar executando. Escreva para a Maplesoft para fazê-los saber que você gostaria de uma versão nativa para FreeBSD!

### 22.4.1. Problemas Comuns

- O gerenciador de licenças **FLEXlm** pode ser uma ferramenta difícil de se trabalhar. Informações adicionais sobre o assunto pode ser encontradas em <http://www.globetrotter.com/>.
- `lmgrd` é conhecido por ser problemático com o arquivo de licença e com despejo de memória. Um arquivo de licença correto se parece com:
  - # =====
  - # License File for UNIX Installations ("Pointer File")
  - # =====
  - SERVER chillig ANY
  - #USE\_SERVER
  - VENDOR maplelmg
  - 
  - FEATURE Maple maplelmg 2000.0831 permanent 1 XXXXXXXXXXXXX \



- PLATFORMS=i86\_r ISSUER="Waterloo Maple Inc." \
- ISSUED=11-may-2000 NOTICE=" Technische Universitat  
Wien" \
- SN=XXXXXXXXXX

**Nota:** Aqui letra ``X" aparece no lugar do número de série e da chave. E chillig é o nome do servidor.

Editar o arquivo de licença funciona se você não mexer na linha ``FEATURE" (que é protegida pela chave de licença).

---

## 22.5. Instalando o MATLAB®

*Contribuição de Dan Pelleg.*

Este documento descreve o processo de instalação da versão Linux do **MATLAB® versão 6.5** em um sistema FreeBSD. Funciona razoavelmente bem, com a exceção do **Java Virtual Machine™** (veja [Seção 22.5.6](#)).

A versão Linux do **MATLAB** pode ser solicitada diretamente do *The MathWorks* em <http://www.mathworks.com>. Certifique-se que você obteve o arquivo de licença ou as instruções de como criá-lo. Enquanto você estiver por lá, faça com que eles saibam que você gostaria de ter uma versão do programa deles nativa para FreeBSD.

### 22.5.1. Instalando o MATLAB

Para instalar o **MATLAB**, faça o seguinte:

1. Insira o CD de instalação de monte-o. Torne-se `root`, como recomendado pelo *script* de instalação. Para inicializar a instalação digite:
2. `# /compat/linux/bin/sh /cdrom/install`

**Dica:** O instalador é gráfico. Se você obtiver erros do tipo não posso abrir um *display*, digite `setenv HOME ~USER`, onde *USER* é o usuário que executou o [su\(1\)](#).

3. Quando o **MATLAB** perguntar pelo diretório `root`, digite:  
`/compat/linux/usr/local/matlab.`

**Dica:** Para facilitar a digitação durante o resto do processo de instalação, digite isto na sua linha de comando: `set  
MATLAB=/compat/linux/usr/local/matlab`

4. Edite o arquivo de licença de acordo as instruções obtidas no arquivo de licença do **MATLAB**.

**Dica:** Você pode preparar este arquivo antecipadamente usando seu editor favorito, e copiando-o para `$MATLAB/license.dat` antes que o instalador solicite que você o edite.

5. Complete o processo de instalação.

Neste ponto a instalação do seu **MATLAB** está completa. Os seguintes passos aplicam uma ``cola" para conectá-lo ao seu sistema FreeBSD.

### 22.5.2. Inicialização do Gerenciador de Licenças

1. Crie ligações simbólicas para os *scripts* do gerenciador de licenças:
2. `# ln -s $MATLAB/etc/lmboot /usr/local/etc/lmboot_TMW`
3. `# ln -s $MATLAB/etc/lmdown /usr/local/etc/lmdown_TMW`
4. Crie um arquivo de inicialização em `/usr/local/etc/rc.d/flexlm.sh`. O exemplo acima é uma versão modificada do `$MATLAB/etc/rc.lm.glnx86`. As mudanças são em localização de arquivos, e inicialização do gerenciador de licenças sob emulação Linux.
5. `#!/bin/sh`
6. `case "$1" in`
7. `start)`
8. `if [ -f /usr/local/etc/lmboot_TMW ]; then`
9. `/compat/linux/bin/sh /usr/local/etc/lmboot_TMW -u`  
`username && echo 'MATLAB_lmgrd'`
10. `fi`
11. `;;`
12. `stop)`
13. `if [ -f /usr/local/etc/lmdown_TMW ]; then`
14. `/compat/linux/bin/sh /usr/local/etc/lmdown_TMW >`  
`/dev/null 2>&1`
15. `fi`
16. `;;`
17. `*)`
18. `echo "Usage: $0 {start|stop}"`
19. `exit 1`
20. `;;`
21. `esac`
- 22.
23. `exit 0`

**Importante:** O arquivo deve ser feito executável:

```
chmod +x /usr/local/etc/rc.d/flexlm.sh
```

Você também deve substituir *username* acima por um nome válido em seu sistema (e não `root`).

24. Inicialize o gerenciador de licença com o comando:
25. `# /usr/local/etc/rc.d/flexlm.sh start`

### 22.5.3. Ligando o Ambiente de Tempo de Execução Java™

Modifique a ligação do Ambiente de Tempo de Execução **Java™** (JRE) para um funcionando no FreeBSD:

```
cd $MATLAB/sys/java/jre/glnx86/
unlink jre; ln -s ./jre1.1.8 ./jre
```

### 22.5.4. Crie um *Script* de Inicialização MATLAB

1. Coloque o seguinte *script* de inicialização em `/usr/local/bin/matlab`:
2. `#!/bin/sh`
3. `/compat/linux/bin/sh /compat/linux/usr/local/matlab/bin/matlab "$@"`
4. Então digite o comando `chmod +x /usr/local/bin/matlab`.

**Dica:** Dependendo da sua versão do [emulators/linux\\_base](#), você pode obter erros ao executar este *script*. Para evitar isso, edite o arquivo

`/compat/linux/usr/local/matlab/bin/matlab`, e modifique a linha que diz:

```
if [`expr "$lscmd" : '.*->.*'` -ne 0]; then
```

(in version 13.0.1 it is on line 410) to this line:

```
if test -L $newbase; then
```

### 22.5.5. Criando um *Script* de Finalização do MATLAB

O seguinte procedimento é necessário para resolver o problema do MATLAB não sair corretamente.

1. Crie o arquivo `$MATLAB/toolbox/local/finish.m`, e insira a única linha:
2. `! $MATLAB/bin/finish.sh`

**Nota:** O `$MATLAB` é literal.

**Dica:** No mesmo diretório, você encontrará os arquivos `finishsav.m` e `finishdlg.m`, que permitirão a você salvar seu espaço de trabalho antes de sair. Se você usar qualquer um deles, insira a linha acima imediatamente depois do comando `save`.

3. Crie o arquivo `$MATLAB/bin/finish.sh`, que possuirá o seguinte conteúdo:
4. `#!/usr/compat/linux/bin/sh`
5. `(sleep 5; killall -1 matlab_helper) &`
6. `exit 0`
7. Faça com que o arquivo seja executável:
8. `# chmod +x $MATLAB/bin/finish.sh`

### 22.5.6. Usando o MATLAB

Neste ponto você está pronto para digitar `matlab` e começar a usá-lo.

## 22.6. Instalando Oracle®

Contribuição de Marcel Moolenaar.

### 22.6.1. Prefacio

Este documento descreve o processo de instalação do **Oracle® 8.0.5** e **Oracle 8.0.5.1 Enterprise Edition** para Linux em uma máquina FreeBSD.

### 22.6.2. Instalando o Ambiente Linux

Esteja certo de ter instalado o [emulators/linux\\_base](#) e [devel/linux\\_devtools](#) a partir da coleção do ports. Se você passar por dificuldades com estes ports, você pode usar os pacotes ou versões antigas disponíveis na coleção dos ports.

Se você quer executar o "intelligent agent", você também precisará instalar o pacote Tcl do RedHat: `tc1-8.0.3-20.i386.rpm`. O comando geral para instalar pacotes com o port oficial do **RPM** ([archivers/rpm](#)) é:

```
rpm -i --ignoreos --root /compat/linux --dbpath /var/lib/rpm pacote
```

Instalação do *pacote* não deve gerar erros.

### 22.6.3. Criando o Ambiente Oracle

Antes que você possa instalar o **Oracle**, você precisa configurar um ambiente adequado. Este documento apenas descreve o que fazer *especificamente* procedimentos para executar o **Oracle** para Linux no FreeBSD, e não o que está descrito no guia de instalação do **Oracle**.

#### 22.6.3.1. Ajustes do Kernel

Como descrito no guia de instalação do **Oracle**, você precisa configurar o tamanho máximo da memória compartilhada. Não use o *SHMMAX* no FreeBSD. *SHMMAX* é meramente calculado através do *SHMMAXPGS* e *PGSIZE*. Sendo assim, defina o *SHMMAXPGS*. Todas as outras opções podem ser usadas como descritas no guia. Por exemplo:

```
options SHMMAXPGS=10000
options SHMMNI=100
options SHMSEG=10
options SEMMNS=200
options SEMMNI=70
options SEMMSL=61
```

Configure estas opções para atender suas necessidades no uso do **Oracle**.

Também tenha certeza de ter as seguintes opções em seu arquivo de configuração de *kernel*:

```
options SYSVSHM #SysV shared memory
options SYSVSEM #SysV semaphores
options SYSVMSG #SysV interprocess communication
```

### 22.6.3.2. Conta de Usuário Oracle

Crie uma conta `oracle` assim como você criaria qualquer outra conta. A conta `oracle` é especial apenas para possuir um *shell* Linux. Adicione `/compat/linux/bin/bash` ao `/etc/shells` e configure o *shell* da conta `oracle` para `/compat/linux/bin/bash`.

### 22.6.3.3. Ambiente

Além das variáveis usuais do **Oracle** como `ORACLE_HOME` e `ORACLE_SID` você deve configurar as seguintes variáveis de ambiente:

| Variável                     | Valor                                                                                                                                                                 |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>LD_LIBRARY_PATH</code> | <code>\$ORACLE_HOME/lib</code>                                                                                                                                        |
| <code>CLASSPATH</code>       | <code>\$ORACLE_HOME/jdbc/lib/classes111.zip</code>                                                                                                                    |
| <code>PATH</code>            | <code>/compat/linux/bin /compat/linux/sbin /compat/linux/usr/bin<br/>/compat/linux/usr/sbin /bin /sbin /usr/bin /usr/sbin /usr/local/bin<br/>\$ORACLE_HOME/bin</code> |

É aconselhável configurar todas as variáveis de ambiente no arquivo `.profile`. Um exemplo completo seria:

```
ORACLE_BASE=/oracle; export ORACLE_BASE
ORACLE_HOME=/oracle; export ORACLE_HOME
LD_LIBRARY_PATH=$ORACLE_HOME/lib
export LD_LIBRARY_PATH
ORACLE_SID=ORCL; export ORACLE_SID
ORACLE_TERM=386x; export ORACLE_TERM
CLASSPATH=$ORACLE_HOME/jdbc/lib/classes111.zip
export CLASSPATH
PATH=/compat/linux/bin:/compat/linux/sbin:/compat/linux/usr/bin
PATH=$PATH:/compat/linux/usr/sbin:/bin:/sbin:/usr/bin:/usr/sbin
PATH=$PATH:/usr/local/bin:$ORACLE_HOME/bin
export PATH
```

### 22.6.4. Instalando o Oracle

Devido a uma pequena inconsistência no emulador Linux, você precisa criar um diretório com o nome `.oracle` em `/var/tmp` antes de inicializar o instalador. Faça com que ele tenha permissão de escrita para todos ou faça com que seu proprietário seja o usuário `oracle`. Você deve ser capaz de instalar o **Oracle** sem problemas. Se você tiver problemas, verifique a sua distribuição **Oracle** e/ou configuração primeiro! Após ter instalado o **Oracle**, aplique as correções descritas nas próximas duas subseções.

Um problema frequente é que o adaptador do protocolo TCP não é instalado corretamente. Como consequência, você não pode iniciar qualquer aplicação de escuta TCP. As seguintes ações ajudam a resolver este problema:

```
cd $ORACLE_HOME/network/lib
make -f ins_network.mk ntcontab.o
cd $ORACLE_HOME/lib
ar r libnetwork.a ntcontab.o
cd $ORACLE_HOME/network/lib
make -f ins_network.mk install
```

Não esqueça de executar o `root.sh` novamente!

### 22.6.4.1. Corrigindo o `root.sh`

Ao instalar o **Oracle**, algumas ações, precisam ser efetuadas como `root`, são gravadas em um *shell script* `root.sh`. Este *script* é escrito no diretório `orainst`. Aplique a seguinte correção em `root.sh`, para que ele use o local correto do `chown` ou alternativamente execute o *script* em um *shell* nativo Linux.

```
*** orainst/root.sh.orig Tue Oct 6 21:57:33 1998
--- orainst/root.sh Mon Dec 28 15:58:53 1998

*** 31,37 ****
This is the default value for CHOWN
It will redefined later in this script for those ports
which have it conditionally defined in ss_install.h
! CHOWN=/bin/chown
#
Define variables to be used in this script
--- 31,37 ----
This is the default value for CHOWN
It will redefined later in this script for those ports
which have it conditionally defined in ss_install.h
! CHOWN=/usr/sbin/chown
#
Define variables to be used in this script
```

Quando você não instala o **Oracle** a partir do CD, você pode corrigir o fonte do `root.sh`. Seu nome é `rthd.sh` e está localizado no diretório `orainst` na árvore dos fontes.

### 22.6.4.2. Corrigindo o `genclntsh`

O *script* `genclntsh` é usado para criar uma única biblioteca cliente compartilhada. Ela é usada quando os *demos* são compilados. Aplique a seguinte correção para comentar a definição do `PATH`:

```
*** bin/genclntsh.orig Wed Sep 30 07:37:19 1998
--- bin/genclntsh Tue Dec 22 15:36:49 1998

*** 32,38 ****
#
Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
```

```

! PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
each product MUST provide a $PRODUCT/admin/shrept.lst
--- 32,38 ----
#
Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! #PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
each product MUST provide a $PRODUCT/admin/shrept.lst

```

## 22.6.5. Executando o Oracle

Ao seguir estas instruções você deve ser capaz de executar o **Oracle** como se estivesse no Linux.

# 22.7. Instalando o SAP® R/3®

*Contribuição de Holger Kipp. Versão Original convertida para SGML por Valentino Vaschetto.*

Instalações de Sistemas **SAP®** usando FreeBSD não serão suportadas pela equipe de suporte da SAP, eles apenas oferecem suporte para plataformas certificadas.

## 22.7.1. Prefácio

Este documento descreve uma maneira possível de instalar um Sistema **SAP R/3®** com **Banco de Dados Oracle®** para Linux em uma máquina FreeBSD, incluindo a instalação do FreeBSD e do **Oracle**. Duas configurações diferentes serão descritas:

- **SAP R/3 4.6B (IDES)** com **Oracle 8.0.5** no FreeBSD 4.3-STABLE
- **SAP R/3 4.6C** com **Oracle 8.1.7** no FreeBSD 4.5-STABLE

Ainda que este documento tente descrever todos os passos importantes em grande detalhe, o mesmo não serve como substituto para os guias de instalação do **Oracle** e do **SAP R/3**.

Por favor veja a documentação que acompanha a edição Linux do **SAP R/3** para o **SAP** e questões específicas do **Oracle**, assim como recursos da **Oracle** e **SAP OSS**.

## 22.7.2. Software

Os seguintes CD-ROMs foram usados nas instalações do **SAP**:

### 22.7.2.1. SAP R/3 4.6B, Oracle® 8.0.5

| Nome | Número | Descrição |
|------|--------|-----------|
|      |        |           |

| Nome    | Número   | Descrição                                              |
|---------|----------|--------------------------------------------------------|
| KERNEL  | 51009113 | SAP Kernel Oracle / Installation / AIX, Linux, Solaris |
| RDBMS   | 51007558 | Oracle / RDBMS 8.0.5.X / Linux                         |
| EXPORT1 | 51010208 | IDES / DB-Export / Disco 1 de 6                        |
| EXPORT2 | 51010209 | IDES / DB-Export / Disco 2 de 6                        |
| EXPORT3 | 51010210 | IDES / DB-Export / Disco 3 de 6                        |
| EXPORT4 | 51010211 | IDES / DB-Export / Disco 4 de 6                        |
| EXPORT5 | 51010212 | IDES / DB-Export / Disco 5 de 6                        |
| EXPORT6 | 51010213 | IDES / DB-Export / Disco 6 de 6                        |

Adicionalmente, usamos o CD do **Oracle 8 Server** (versão de pré-produção para Linux, *kernel 2.0.33*) que não é tão necessário, e o FreeBSD 4.3-STABLE (que estava há poucos dias na RELEASE 4.3).

### 22.7.2.2. SAP R/3 4.6C SR2, Oracle 8.1.7

| Nome    | Número   | Descrição                                                |
|---------|----------|----------------------------------------------------------|
| KERNEL  | 51014004 | SAP Kernel Oracle / SAP Kernel Version 4.6D / DEC, Linux |
| RDBMS   | 51012930 | Oracle 8.1.7/ RDBMS / Linux                              |
| EXPORT1 | 51013953 | Release 4.6C SR2 / Export / Disco 1 de 4                 |
| EXPORT1 | 51013953 | Release 4.6C SR2 / Export / Disco 2 de 4                 |
| EXPORT1 | 51013953 | Release 4.6C SR2 / Export / Disco 3 de 4                 |
| EXPORT1 | 51013953 | Release 4.6C SR2 / Export / Disco 4 de 4                 |
| LANG1   | 51013954 | Release 4.6C SR2 / Language / DE, EN, FR / Disc 1 of 3   |

Dependendo das línguas que você deseja instalar, CDs adicionais de línguas podem ser necessários. Aqui estamos usando apenas DE e EN, sendo assim, o primeiro CD de língua é necessário. Como uma pequena nota, os números de todos os quatro CDs EXPORT são idênticos. Todos os três CDs de língua também possuem o mesmo número (que são diferentes da versão de numeração do 4.6B IDES). No momento em



que redigimos este texto esta instalação está sendo executada em um FreeBSD 4.5-STABLE (20.03.2002).

### 22.7.3. Notas do SAP

As seguintes notas devem ser lidas antes da instalação do **SAP R/3**, pois provaram ser úteis:

#### 22.7.3.1. SAP R/3 4.6B, Oracle 8.0.5

| Número  | Título                                                |
|---------|-------------------------------------------------------|
| 0171356 | SAP Software on Linux: Essential Comments             |
| 0201147 | INST: 4.6C R/3 Inst. on UNIX - Oracle                 |
| 0373203 | Update / Migration Oracle 8.0.5 --> 8.0.6/8.1.6 LINUX |
| 0072984 | Release of Digital UNIX 4.0B for Oracle               |
| 0130581 | R3SETUP step DIPGNTAB terminates                      |
| 0144978 | Your system has not been installed correctly          |
| 0162266 | Questions and tips for R3SETUP on Windows NT / W2K    |

#### 22.7.3.2. SAP R/3 4.6C, Oracle 8.1.7

| Número  | Título                                        |
|---------|-----------------------------------------------|
| 0015023 | Initializing table TCPDB (RSXP0004) (EBCDIC)  |
| 0045619 | R/3 with several languages or typefaces       |
| 0171356 | SAP Software on Linux: Essential Comments     |
| 0195603 | RedHat 6.1 Enterprise version: Known problems |
| 0212876 | The new archiving tool SAPCAR                 |
| 0300900 | Linux: Released DELL Hardware                 |
| 0377187 | RedHat 6.2: important remarks                 |

| Número  | Título                                         |
|---------|------------------------------------------------|
| 0387074 | INST: R/3 4.6C SR2 Installation on UNIX        |
| 0387077 | INST: R/3 4.6C SR2 Inst. on UNIX - Oracle      |
| 0387078 | SAP Software on UNIX: OS Dependencies 4.6C SR2 |

#### 22.7.4. Hardware Necessário

O seguinte equipamento é suficiente para a instalação de um Sistema **SAP R/3**. Para produção, um tamanho maior para o processo de extração é necessário:

| Componente      | 4.6B                    | 4.6C                   |
|-----------------|-------------------------|------------------------|
| Processador     | 2 x 800MHz Pentium® III | 2 x 800MHz Pentium III |
| Memoria         | 1GB ECC                 | 2GB ECC                |
| Espaço em Disco | 50-60GB (IDES)          | 50-60GB (IDES)         |

Para uso em produção, Processadores Xeon™ com *cache*, alta velocidade de acesso à disco (SCSI, controladora de hardware RAID), USV e memória RAM ECC é recomendado. As grandes quantidades de disco são para o Sistem IDEs pré-configurado, que cria 27 GB de arquivos de banco e dados durante a instalação. Este espaço também é inicial para sistemas de produção e aplicações de dados.

##### 22.7.4.1. SAP R/3 4.6B, Oracle 8.0.5

O seguinte hardwar foi usado: uma placa bi-processada com dois processadores Pentium III 800 MHz, adaptadora SCSI Adaptec® 29160 Ultra160 (para acesso à uma unidade DLT 40/80 GB e um CDROM), controladora Mylex® AcceleRAID™ (2 canais, firmware 6.00-1-00 com 32 MB de RAM). Para a controladora Mylex nós conectamos dois discos de 17 GB (espelhados) e quatro discos de 36 GB (RAID nível 5).

##### 22.7.4.2. SAP R/3 4.6C, Oracle 8.1.7

Para esta instalação um Dell™ PowerEdge™ 2500 foi usado: placa bi-processada com dois processadores Pentium III de 1000 MHz (256 kB de *Cache*), 2 GB de memória PC133 ECC SDRAM, Controladora PERC/3 DC PCI RAID com 128 MB, e um drive EIDE DVD-ROM. Na a controladora RAID estão ligados dois discos de 18 GB (espelhados) e quatro discos de 36 GB (RAID nível 5).

## 22.7.5. Instalação do FreeBSD

Primeiro você deve instalar o FreeBSD. Existem várias maneiras de se fazer isto (o FreeBSD 4.3 foi instalado via FTP, o FreeBSD 4.5 diretamente do CD), para mais informações leia a [Seção 2.13](#).

### 22.7.5.1. Disposição do Disco

Para manter a simplicidade, a mesma disposição de disco para a instalação do **SAP R/3 46B** e do **SAP R/3 46C SR2** foi usada. Apenas os nomes dos dispositivos foi modificada, pois as instalações foram feitas em equipamentos distintos (`/dev/da` e `/dev/amr` respectivamente, então usando o AMI MegaRAID®, uma enxergará `/dev/amr0s1a` ao invés de `/dev/da0s1a`):

| Sistema de arquivos      | Tamanho (blocos de 1k) | Tamanho (GB) | Montado em                         |
|--------------------------|------------------------|--------------|------------------------------------|
| <code>/dev/da0s1a</code> | 1.016.303              | 1            | <code>/</code>                     |
| <code>/dev/da0s1b</code> |                        | 6            | <code>swap</code>                  |
| <code>/dev/da0s1e</code> | 2.032.623              | 2            | <code>/var</code>                  |
| <code>/dev/da0s1f</code> | 8.205.339              | 8            | <code>/usr</code>                  |
| <code>/dev/da1s1e</code> | 45.734.361             | 45           | <code>/compat/linux/oracle</code>  |
| <code>/dev/da1s1f</code> | 2.032.623              | 2            | <code>/compat/linux/sapmnt</code>  |
| <code>/dev/da1s1g</code> | 2.032.623              | 2            | <code>/compat/linux/usr/sap</code> |

Configure e inicialize dois drives lógicos antecipadamente com o software da Mylex ou PERC/3 RAID. O software pode ser inicializado durante a fase de inicialização da BIOS.

Por favor note que as disposições de disco diferem um pouco das recomendações da SAP, pois a sugestão é que se monte os subdiretórios do **Oracle** (e alguns outros) separadamente -- nós apenas dedicamos criá-los em subdiretórios reais por simplicidade.

### 22.7.5.2. `make world` e um Novo Kernel

Baixe da Internet os últimos fontes do `-STABLE`. Recompile seu sistema e seu *kernel* após editar seu arquivo de configuração de *kernel*. Aqui você também deve incluir seus [parâmetros de kernel](#) que são necessários tanto para o **SAP R/3** como para o **Oracle**.

## 22.7.6. Instalando o Ambiente Linux

### 22.7.6.1. Instalando o Sistema Base Linux

Primeiro o [linux\\_base](#) precisa ser instalado (como root):

```
cd /usr/ports/emulators/linux_base
make install distclean
```

### 22.7.6.2. Instalando o Ambiente de Desenvolvimento Linux

O ambiente de desenvolvimento Linux é necessário, se você deseja instalar o **Oracle** no FreeBSD, de acordo com a [Seção 22.6](#):

```
cd /usr/ports/devel/linux_devtools
make install distclean
```

O ambiente de desenvolvimento Linux foi apenas instalado para o **SAP R/3 46B IDES**. Não é necessário, se o **Oracle DB** não é religado no sistema FreeBSD. Este é o caso se você estiver usando uma instalação do **Oracle** vinda de um sistema Linux.

### 22.7.6.3. Instalando os RPMs Necessários

Para inicializar o programa `R3SETUP`, o suporte a PAM é necessário. Durante a primeira instalação do **SAP** no FreeBSD 4.3-STABLE nós tentamos instalar o PAM com todos os pacotes requeridos e finalmente forçamos a instalação do pacote PAM, que funcionou. Para o **SAP R/3 4.6C SR2** nós forçamos diretamente a instalação do RPM do PAM, que também funcionou, causando a impressão de que os pacotes dependentes não são necessários:

```
rpm -i --ignoreos --nodeps --root /compat/linux --dbpath
/var/lib/rpm \
pam-0.68-7.i386.rpm
```

Para o **Oracle 8.0.5** executar o *intelligent agent*, nós também tivemos que instalar o pacote Tcl RedHat `tc1-8.0.5-30.i386.rpm` (caso contrário o religamento durante a instalação do **Oracle** não funcionaria). Existem outras questões a respeito da religação do **Oracle**, mas isso é uma questão do **Oracle** para Linux, e não específica do FreeBSD.

### 22.7.6.4. Algumas Dicas Adicionais

Também pode ser uma boa idéia adicionar o *linprocfs* ao `/etc/fstab`, para mais informações veja a página de manual [linprocfs\(5\)](#). Outro parâmetro a ser configurado é o `kern.fallback_elf_brand=3` que é feito no arquivo `/etc/sysctl.conf`.

## 22.7.7. Criando o Ambiente SAP R/3

### 22.7.7.1. Criando os Sistemas de Arquivo e Pontos de Montagem Necessários

Para uma instalação simples, é suficiente criar os seguintes sistemas de arquivo:

| Ponto de Montagem | tamanho em GB |
|-------------------|---------------|
|-------------------|---------------|

| Ponto de Montagem     | tamanho em GB |
|-----------------------|---------------|
| /compat/linux/oracle  | 45 GB         |
| /compat/linux/sapmnt  | 2 GB          |
| /compat/linux/usr/sap | 2 GB          |

Também é necessário criar algumas ligações. Caso contrário o Instalador **SAP** reclamará ao verificá-las:

```
ln -s /compat/linux/oracle /oracle
ln -s /compat/linux/sapmnt /sapmnt
ln -s /compat/linux/usr/sap /usr/sap
```

Possíveis erros durante a instalação (aqui com o Sistema *PRD* e o **SAP R/3 4.6C SR2**):

```
INFO 2002-03-19 16:45:36 R3LINKS_IND_IND SyLinkCreate:200
Checking existence of symbolic link /usr/sap/PRD/SYS/exe/dbg to
/sapmnt/PRD/exe. Creating if it does not exist...
```

```
WARNING 2002-03-19 16:45:36 R3LINKS_IND_IND SyLinkCreate:400
Link /usr/sap/PRD/SYS/exe/dbg exists but it points to file
/compat/linux/sapmnt/PRD/exe instead of /sapmnt/PRD/exe. The
program cannot go on as long as this link exists at this
location. Move the link to another location.
```

```
ERROR 2002-03-19 16:45:36 R3LINKS_IND_IND Ins_SetupLinks:0
can not setup link '/usr/sap/PRD/SYS/exe/dbg' with content
'/sapmnt/PRD/exe'
```

### 22.7.7.2. Criando Usuários e Diretórios

**SAP R/3** precisa de dois usuários e três grupos. O nome do usuário depende do ID de sistema (SID) do **SAP**, que consistem em três letras. Algumas destas SIDs são reservadas pelo **SAP** (por exemplo *SAP* e *NIX*. Para uma lista completa por favor veja a documentação do **SAP**). Para a instalação do IDES nós usamos *IDS*, para a instalação do 4.6C SR2 *PRD*, pois é intenção fazer com que o sistema seja de produção. Por esta razão criamos os seguintes grupos (IDs de grupo podem ser diferentes, estes são apenas os valores que usamos na nossa instalação):

| ID do grupo | nome do grupo | descrição               |
|-------------|---------------|-------------------------|
| 100         | dba           | Data Base Administrator |
| 101         | sapsys        | SAP System              |
| 102         | oper          | Data Base Operator      |

Para uma instalação padrão do **Oracle** apenas o grupo `dba` é usado. Assim como o grupo `oper` um também usa o grupo `dba` (veja a documentação do **Oracle** e do **SAP** para mais informações).

Também precisamos dos seguintes usuários:

| ID do usuário | nome do usuário | nome genérico | grupo  | grupos adicionais | descrição            |
|---------------|-----------------|---------------|--------|-------------------|----------------------|
| 1000          | idsadm/prdadm   | <i>sidadm</i> | sapsys | oper              | SAP Administrator    |
| 1002          | oraids/oraprd   | <i>orasid</i> | dba    | oper              | Oracle Administrator |

Adicionar os usuários com [adduser\(8\)](#) requer as seguintes entradas (por favor note o *shell* e o diretório *home*) para o ``SAP Administrator``:

```
Name: sidadm
Password: *****
Fullname: SAP Administrator SID
Uid: 1000
Gid: 101 (sapsys)
Class:
Groups: sapsys dba
HOME: /home/sidadm
Shell: bash (/compat/linux/bin/bash)
```

e para o ``Oracle Administrator``:

```
Name: orasid
Password: *****
Fullname: Oracle Administrator SID
Uid: 1002
Gid: 100 (dba)
Class:
Groups: dba
HOME: /oracle/sid
Shell: bash (/compat/linux/bin/bash)
```

Isto também deve incluir o grupo `oper` caso você esteja usando ambos os grupos `dba` e `oper`.

### 22.7.7.3. Criando Diretórios

Estes diretórios normalmente são criados como sistemas de arquivos distintos. Isto depende completamente das suas necessidades. Nós escolhemos criá-los simplesmente como diretórios, pois, de qualquer forma, estão todos localizados no mesmo RAID 5:

Primeiro vamos configurar os proprietários e os direitos de alguns diretórios (como usuário `root`):

```
chmod 775 /oracle
chmod 777 /sapmnt
chown root:dba /oracle
chown sidadm:sapsys /compat/linux/usr/sap
chmod 775 /compat/linux/usr/sap
```

Segundo, vamos criar diretórios como usuário *orasid*. Estes serão subdiretórios de */oracle/SID*:

```
su - orasid
cd /oracle/SID
mkdir mirrlogA mirrlogB origlogA origlogB
mkdir sapdata1 sapdata2 sapdata3 sapdata4 sapdata5 sapdata6
mkdir saparch sapreorg
exit
```

Para o **Oracle 8.1.7** alguns diretórios adicionais são necessários:

```
su - orasid
cd /oracle
mkdir 805_32
mkdir client stage
mkdir client/80x_32
mkdir stage/817_32
cd /oracle/SID
mkdir 817_32
```

**Nota:** O diretório *client/80x\_32* é usado exatamente com este nome. Não substitua o *x* com um número ou qualquer outra coisa.

No terceiro passo criaremos diretórios como o usuário *sidadm*:

```
su - sidadm
cd /usr/sap
mkdir SID
mkdir trans
exit
```

#### 22.7.7.4. Entradas em */etc/services*

**SAP R/3** requer algumas entradas no */etc/services*, que não serão configuradas corretamente na instalação do FreeBSD. Por favor adicione as seguintes entradas (você precisa pelo menos das entradas correspondentes ao número da instancia -- neste caso, *00*. Inserir todas as entradas de *00* à *99* para *dp*, *gw*, *sp* e *ms* não causará problemas). Se você for utilizar o **SAProuter** ou precise acessar o **SAP OSS**, você também precisa do *99*, uma vez que a porta 3299 é usada para o processo do **SAProuter** no sistema de destino:

```
sapdp00 3200/tcp # SAP Dispatcher. 3200 + Instance-Number
sapgw00 3300/tcp # SAP Gateway. 3300 + Instance-Number
sapsp00 3400/tcp # 3400 + Instance-Number
sapms00 3500/tcp # 3500 + Instance-Number
sapmsSID 3600/tcp # SAP Message Server. 3600 + Instance-Number
sapgw00s 4800/tcp # SAP Secure Gateway 4800 + Instance-Number
```

### 22.7.7.5. Localização Necessária

SAP requer pelo menos dois locais que não são parte da instalação padrão do RedHat. SAP oferece os RPMs através de seu servidor de FTP (que somente é acessível se você for um cliente com acesso OSS). Veja a nota 0171356 para a lista de RPMs que você precisa.

Também é possível criar as ligações apropriadas (por exemplo de *de\_DE* e *en\_US*), mas nós não recomendamos isto para um ambiente de produção (até agora funcionou com o sistema IDES sem nenhum problema). As seguintes localizações são necessárias:

```
de_DE.ISO-8859-1
en_US.ISO-8859-1
```

Crie as ligações como segue:

```
cd /compat/linux/usr/share/locale
ln -s de_DE de_DE.ISO-8859-1
ln -s en_US en_US.ISO-8859-1
```

Se elas não estiverem presentes, alguns problemas ocorrerão durante a instalação. Se forem ignorados (configurando-se o *STATUS* dos passos problemáticos para *OK* no arquivo *CENTRDB.R3S*), será impossível efetuar *logon* no sistema **SAP** sem um esforço adicional.

### 22.7.7.6. Ajustes de Kernel

Sistemas **SAP R/3** precisam de muitos recursos. Sendo assim, adicionamos os seguintes parâmetros no arquivos de configuração do *kernel*

```
Set these for memory pigs (SAP and Oracle):
options MAXDSIZ="(1024*1024*1024)"
options DFLDSIZ="(1024*1024*1024)"
System V options needed.
options SYSVSHM #SYSV-style shared memory
options SHMMAXPGS=262144 #max amount of shared mem. pages
#options SHMMAXPGS=393216 #use this for the 46C inst.parameters
options SHMMNI=256 #max number of shared memory ident if.
options SHMSEG=100 #max shared mem.segs per process
options SYSVMSG #SYSV-style message queues
options MSGSEG=32767 #max num. of mes.segments in system
options MSGSSZ=32 #size of msg-seg. MUST be power of 2
options MSGMNB=65535 #max char. per message queue
options MSGTQL=2046 #max amount of msgs in system
options SYSVSEM #SYSV-style semaphores
options SEMMNU=256 #number of semaphore UNDO structures
options SEMMNS=1024 #number of semaphores in system
options SEMMNI=520 #number of semaphore identifiers
options SEMUME=100 #number of UNDO keys
```

Os valores mínimos são especificados na documentação que vem com o SAP. Como não existe descrição para Linux, veja a seção HP-UX (32-bit) para mais informações. Como a instalação do sistema 4.6C SR2 tem mais memória, os segmentos



compartilhados pode ser maiores tanto para o **SAP** quanto para o **Oracle**, por esta razão escolha um número maior de páginas de memória compartilhada.

**Nota:** Com a instalação padrão do FreeBSD 4.5 em um i386™, deixe *MAXDSIZ* e *DFLDSIZ* em 1 GB no máximo. Caso contrário, erros estranhos como ``ORA-27102: out of memory" e ``Linux Error: 12: Cannot allocate memory" podem acontecer.

## 22.7.8. Instalando o SAP R/3

### 22.7.8.1. Preparando os CDROMs do SAP

Existem muitos CDROMs a serem montados e desmontados durante a instalação. Assumindo que você possui drives de CDROM suficientes, você pode montar todos. Nós decidimos copiar o conteúdo dos CDROMs para os diretórios correspondentes:

```
/oracle/SID/sapreorg/cd-name
```

onde *cd-name* foi um dos *KERNEL*, *RDBMS*, *EXPORT1*, *EXPORT2*, *EXPORT3*, *EXPORT4*, *EXPORT5* and *EXPORT6* para a instalação do 4.6B/IDES, e *KERNEL*, *RDBMS*, *DISK1*, *DISK2*, *DISK3*, *DISK4* e *LANG* para a instalação do 4.6C SR2. Todos os nomes de arquivos nos CDs montados devem estar em letras maiúsculas, caso contrário use a opção *-g* para montar. Então use os seguintes comandos:

```
mount_cd9660 -g /dev/cd0a /mnt
cp -R /mnt/* /oracle/SID/sapreorg/cd-name
umount /mnt
```

### 22.7.8.2. Executando o Script de Instalação

Primeiro você deve preparar um diretório *install*:

```
cd /oracle/SID/sapreorg
mkdir install
cd install
```

Então quando o *script* for inicializado, o mesmo copiará todos os arquivos relevantes para o diretório *install*:

```
/oracle/SID/sapreorg/KERNEL/UNIX/INSTTOOL.SH
```

A instalação **IDES (4.6B)** vem com um sistema SAP R/3 de demonstração completamente personalizado, então existem seis ao invés de três CDs *EXPORT*. Neste ponto o modelo *CENTRDB.R3S* é para instalação de de uma instância central padrão (**R/3** e base de dados), não a instância central **IDES**, sendo necessário copiar o *CENTRDB.R3S* correspondente do diretório *EXPORT1*, caso contrário o *R3SETUP* perguntará apenas pelos três CDs *EXPORT*.

A nova versão do **SAP 4.6C SR2** vem com quatro CDs *EXPORT*. O arquivo de parâmetro que controla os passos de instalação é o *CENTRAL.R3S*. Ao contrário das versões anteriores não existem modelos de instalação separados para uma instância

central com ou sem base de dados. O **SAP** está usando um modelo separado para a instalação de base de dados. Para reiniciar a instalação posteriormente basta reiniciar com o arquivo original.

Durante e depois da instalação, o **SAP** requer o `hostname` para retornar apenas o nome do computador, não o nome completo de domínio. Então configure o nome do computador de acordo, ou configure um apelido com `alias hostname='hostname -s'` para `orasid` e `sidadm` (e para `root` pelo menos durante os passos da instalação feitos como `root`). Também é possível ajustar os arquivos `.profile` e `.login` existentes de ambos os usuários durante a instalação do **SAP**.

### 22.7.8.3. Inicialize o R3SETUP 4.6B

Certifique-se de que `LD_LIBRARY_PATH` está configurada corretamente:

```
export
LD_LIBRARY_PATH=/oracle/IDS/lib:/sapmnt/IDS/exe:/oracle/805_32/lib
```

Inicie o `R3SETUP` como `root` a partir do diretório de instalação:

```
cd /oracle/IDS/sapreorg/install
./R3SETUP -f CENTRDB.R3S
```

O *script* faz algumas perguntas (o padrão está entre colchetes, seguido da entrada real):

| Questão                                                                                               | Padrão                | Entrada  |
|-------------------------------------------------------------------------------------------------------|-----------------------|----------|
| Enter SAP System ID                                                                                   | [C11]                 | IDSEnter |
| Enter SAP Instance Number                                                                             | [00]                  | Enter    |
| Enter SAPMOUNT Directory                                                                              | [/sapmnt]             | Enter    |
| Enter name of SAP central host                                                                        | [troubadix.domain.de] | Enter    |
| Enter name of SAP db host                                                                             | [troubadix]           | Enter    |
| Select character set                                                                                  | [1] (WE8DEC)          | Enter    |
| Enter Oracle server version<br>(1) Oracle 8.0.5, (2) Oracle 8.0.6, (3) Oracle 8.1.5, (4) Oracle 8.1.6 |                       | 1Enter   |
| Extract Oracle Client archive                                                                         | [1] (Yes, extract)    | Enter    |

| Questão                          | Padrão                         | Entrada                         |
|----------------------------------|--------------------------------|---------------------------------|
| Enter path to KERNEL CD          | [/sapcd]                       | /oracle/IDS/sapreorg/KERNEL     |
| Enter path to RDBMS CD           | [/sapcd]                       | /oracle/IDS/sapreorg/RDBMS      |
| Enter path to EXPORT1 CD         | [/sapcd]                       | /oracle/IDS/sapreorg/EXPORT1    |
| Directory to copy EXPORT1 CD     | [/oracle/IDS/sapreorg/CD4_DIR] | <b>Enter</b>                    |
| Enter path to EXPORT2 CD         | [/sapcd]                       | /oracle/IDS/sapreorg/EXPORT2    |
| Directory to copy EXPORT2 CD     | [/oracle/IDS/sapreorg/CD5_DIR] | <b>Enter</b>                    |
| Enter path to EXPORT3 CD         | [/sapcd]                       | /oracle/IDS/sapreorg/EXPORT3    |
| Directory to copy EXPORT3 CD     | [/oracle/IDS/sapreorg/CD6_DIR] | <b>Enter</b>                    |
| Enter path to EXPORT4 CD         | [/sapcd]                       | /oracle/IDS/sapreorg/EXPORT4    |
| Directory to copy EXPORT4 CD     | [/oracle/IDS/sapreorg/CD7_DIR] | <b>Enter</b>                    |
| Enter path to EXPORT5 CD         | [/sapcd]                       | /oracle/IDS/sapreorg/EXPORT5    |
| Directory to copy EXPORT5 CD     | [/oracle/IDS/sapreorg/CD8_DIR] | <b>Enter</b>                    |
| Enter path to EXPORT6 CD         | [/sapcd]                       | /oracle/IDS/sapreorg/EXPORT6    |
| Directory to copy EXPORT6 CD     | [/oracle/IDS/sapreorg/CD9_DIR] | <b>Enter</b>                    |
| Enter amount of RAM for SAP + DB |                                | 850 <b>Enter</b> (in Megabytes) |
| Service Entry Message Server     | [3600]                         | <b>Enter</b>                    |
| Enter Group-ID of sapsys         | [101]                          | <b>Enter</b>                    |
| Enter Group-ID of oper           | [102]                          | <b>Enter</b>                    |

| Questão                         | Padrão | Entrada |
|---------------------------------|--------|---------|
| Enter Group-ID of dba           | [100]  | Enter   |
| Enter User-ID of <i>sidadm</i>  | [1000] | Enter   |
| Enter User-ID of <i>orasisd</i> | [1002] | Enter   |
| Number of parallel procs        | [2]    | Enter   |

Se você não copiou os CDs para locais diferentes, então o instalador do **SAP** não pode encontrar o CD (identificado pelo arquivo `LABEL.ASC` no CD) e pode então solicitar que você insira e monte o CD e confirme ou que entre um caminho de montagem.

`CENTRDB.R3S` pode não estar livre de erros. Em nosso caso, solicitei o CD `EXPORT4` novamente, mas indicou a chave correta (`6_LOCATION`, então `7_LOCATION`, etc.), assim um pode continuar quando entramos com os valores corretos.

Fora alguns problemas mencionados abaixo, tudo deve correr bem até o ponto onde a base de dados Oracle precisa ser instalada.

#### 22.7.8.4. Inicialize o `R3SETUP 4.6C SR2`

Certifique-se de que `LD_LIBRARY_PATH` está configurada corretamente. Este é um valor diferente da da instalação 4.6B com o **Oracle 8.0.5**:

```
export LD_LIBRARY_PATH=/sapmnt/PRD/exe:/oracle/PRD/817_32/lib
```

Inicie o `R3SETUP` como `root` partir do diretório de instalação:

```
cd /oracle/PRD/sapreorg/install
./R3SETUP -f CENTRAL.R3S
```

O *script* faz algumas questões (o padrão está entre colchetes, seguido da entrada real):

| Questão                        | Padrão     | Entrada  |
|--------------------------------|------------|----------|
| Enter SAP System ID            | [C11]      | PRDEnter |
| Enter SAP Instance Number      | [00]       | Enter    |
| Enter SAPMOUNT Directory       | [/sapmnt]  | Enter    |
| Enter name of SAP central host | [majestix] | Enter    |
| Enter Database System ID       | [PRD]      | PRDEnter |

| Questão                                      | Padrão             | Entrada                     |
|----------------------------------------------|--------------------|-----------------------------|
| Enter name of SAP db host                    | [majestix]         | Enter                       |
| Select character set                         | [1] (WE8DEC)       | Enter                       |
| Enter Oracle server version (2) Oracle 8.1.7 |                    | 2Enter                      |
| Extract Oracle Client archive                | [1] (Yes, extract) | Enter                       |
| Enter path to KERNEL CD                      | [/sapcd]           | /oracle/PRD/sapreorg/KERNEL |
| Enter amount of RAM for SAP + DB             | 2044               | 1800Enter (in Megabytes)    |
| Service Entry Message Server                 | [3600]             | Enter                       |
| Enter Group-ID of sapsys                     | [100]              | Enter                       |
| Enter Group-ID of oper                       | [101]              | Enter                       |
| Enter Group-ID of dba                        | [102]              | Enter                       |
| Enter User-ID of oraprd                      | [1002]             | Enter                       |
| Enter User-ID of prdadm                      | [1000]             | Enter                       |
| LDAP support                                 |                    | 3Enter (no support)         |
| Installation step completed                  | [1] (continue)     | Enter                       |
| Choose installation service                  | [1] (DB inst,file) | Enter                       |

Até agora, a criação de usuários dá um erro durante a instalação em OSUSERDBSID\_IND\_ORA (para criar o usuário *orasid*) e OSUSERSIDADM\_IND\_ORA (para criar o usuário *sidadm*).

Fora alguns problemas mencionados abaixo, tudo deve correr bem até o ponto onde a base de dados Oracle precisa ser instalada.

### 22.7.9. Instalando o Oracle 8.0.5

Por favor veja as notas correspondentes ao SAP e ao Oracle nos arquivos *Readmea* respeito do Linux e do banco de dados **Oracle** para possíveis problemas. A maioria dos problemas provém de bibliotecas incompatíveis.

Para mais informações sobre a instalação do **Oracle**, veja o capítulo [Instalando o Oracle](#).

### 22.7.9.1. Instalando o Oracle 8.0.5 com o `orainst`

Se o **Oracle 8.0.5** for usado, algumas bibliotecas adicionais serão necessárias para uma religação bem sucedida, pois o **Oracle 8.0.5** foi ligado com uma *glibc* antiga (RedHat 6.0), mas o RedHat 6.1 já utiliza uma nova *glibc*. Então você tem que instalar os seguintes pacotes adicionais para ter certeza que a ligação vai funcionar:

```
compat-libs-5.2-2.i386.rpm
```

```
compat-glibc-5.2-2.0.7.2.i386.rpm
```

```
compat-egcs-5.2-1.0.3a.1.i386.rpm
```

```
compat-egcs-c++-5.2-1.0.3a.1.i386.rpm
```

```
compat-binutils-5.2-2.9.1.0.23.1.i386.rpm
```

Veja as notas correspondentes ao SAP e ao Oracle nos arquivos `Readme` para mais informações. Se isto não é uma opção (no momento da instalação não tivemos tempo de verificar isso), um pode usar os binários originais, ou usar os binários religados originalmente no sistema RedHat.

Para compilar o *intelligent agent*, o pacote `Tcl` do RedHat deve estar instalado. Se você não puder pegar o `tcl-8.0.3-20.i386.rpm`, um mais novo como o `tcl-8.0.5-30.i386.rpm` para RedHat 6.1 deve ser suficiente.

Fora a religação, a instalação é direta:

```
su - oraids
export TERM=xterm
export ORACLE_TERM=xterm
export ORACLE_HOME=/oracle/IDS
cd /ORACLE_HOME/orainst_sap
./orainst
```

Confirme todas as telas com **Enter** até que o programa seja instalado, exceto a que tem a opção para deselegionar o *Oracle On-Line Text Viewer*, pois não está disponível para Linux, então o **Oracle** religa com o `i386-glibc20-linux-gcc` ao invés do `gcc` disponível, `egcs` ou `i386-redhat-linux-gcc`.

Devido a restrições de tempo decidimos usar binários de uma versão do **Oracle 8.0.5 PreProduction**, depois da primeira tentativa frustrada de tentar fazer funcionar a versão do RDBMS do CD, procurar e acessar os RPMs corretos naquela ocasião foi um pesadelo.

### 22.7.9.2. Instalando o Oracle 8.0.5 Pre-production Release para Linux (Kernel 2.0.33)

Esta instalação é bem fácil. Monte o CD, inicie o instalador. Então ele perguntará pelo local do diretório *home* do Oracle, e copiará todos os binários para lá. Nós não apagamos os resquícios das nossas tentativas anteriores de instalar o RDBMS.

Depois disso, o Banco de Dados **Oracle** pode ser instalado sem problemas.

### 22.7.10. Instalando o Tarball Linux do Oracle 8.1.7

Pegue o *tarball* `oracle81732.tgz` que você produziu no diretório de instalação do sistema Linux e descompacte-o em `/oracle/SID/817_32/`.

### 22.7.11. Continue com a Instalação do SAP R/3

Primeiro verifique as configurações de ambiente dos usuários `idsamd` (*sidadm*) e `oraids` (*orasid*). Ambos devem possuir os arquivos `.profile`, `.login` e `.cshrc` que estarão usando o `hostname`. No caso em que o nome da máquina seja o nome completo qualificado do domínio, você deve mudar `hostname` para `hostname -s` nos três arquivos.

#### 22.7.11.1. Carga da Base de Dados

Mais tarde, o `R3SETUP` pode ser reinicializado ou continuar com sua execução (dependendo da escolha de sair ou não). O `R3SETUP` então cria os *tablespaces* e carrega os dados (para o 46B IDES, do EXPORT1 ao EXPORT6, para o 46C do DISK1 ao DISK4) com `R3load` no banco de dados.

Quando a carga da base de dados terminar (pode levar algumas horas), algumas senhas serão requisitadas. Para instalações de teste, pode-se usar senhas padrão conhecidas (use senhas diferentes se a segurança for uma preocupação para você!):

| Questão                     | Entrada                |
|-----------------------------|------------------------|
| Enter Password for sapr3    | sapEnter               |
| Confirum Password for sapr3 | sapEnter               |
| Enter Password for sys      | change_on_installEnter |
| Confirm Password for sys    | change_on_installEnter |
| Enter Password for system   | managerEnter           |
| Confirm Password for system | managerEnter           |

Neste ponto tivemos alguns problemas com o `digpntab` durante a instalação do 4.6B.

#### 22.7.11.2. Listener

Inicie o *Listener* do **Oracle** como usuário *oracid* como segue:

```
% umask 0; lsnrctl start
```

Caso contrário você pode ter o erro ORA-12546, pois os *sockets* não possuirão as permissões corretas. Veja a nota SAP 072984.

### 22.7.11.3. Atualizando as Tabelas MNLS

Se você planeja importar linguagens que não sejam Latin-1 para o sistema **SAP**, você tem que atualizar as tabelas de Suporte a Línguas Multi Nacionais. Isto está descrito nas notas OSS 15023 e 45619 do SAP, caso contrário, você pode pular esta questão durante a instalação do **SAP**.

**Nota:** Se você não precisa do MNLS, ainda é necessário verificar a tabela TCPDB e iniciá-la se isso não foi feito. Veja as notas 0015023 e 0045619 do SAP para mais informações.

## 22.7.12. Passos Pós-Instalação

### 22.7.12.1. Solicite a Chave de Licença do SAP R/3

Você precisa solicitar sua Chave de Licença do **SAP R/3**. Isto é necessário, pois a licença temporária que foi usada durante a instalação é válida apenas por quatro semanas. Primeiro obtenha a licença de hardware. Entre como o usuário *idsadm* e execute o *saplicense*:

```
/sapmnt/IDS/exe/saplicense -get
```

Executando o *saplicense* sem parâmetros obtemos uma lista de opções. Recebendo a chave de licença, pode-se instalá-la usando:

```
/sapmnt/IDS/exe/saplicense -install
```

Então você é solicitado a entrar com os seguintes valores:

```
SAP SYSTEM ID = SID, 3 chars
CUSTOMER KEY = hardware key, 11 chars
INSTALLATION NO = installation, 10 digits
EXPIRATION DATE = yyyyymmdd, usually "99991231"
LICENSE KEY = license key, 24 chars
```

### 22.7.12.2. Criando Usuários

Crie um usuário dentro do cliente 000 (algumas tarefas são feitas com o cliente 000, mas com um usuário diferente do *sap\** e *ddic*). Como um nome de usuário, normalmente escolhemos *wartung* (ou *service* em Inglês). Os perfis requeridos são *sap\_new* e *sap\_all*. Para segurança adicione uma senha dos usuários padrão dentro de todos os devem ser modificadas (isto inclui os usuários *sap\** e *ddic*).



### 22.7.12.3. Configure o Sistema de Transporte, Perfil, Modos de Operação, Etc.

Dentro do cliente 000, com usuário diferente de `ddic` e `sap*`, faça pelo menos o seguinte:

| Tarefa                                                                         | Transação |
|--------------------------------------------------------------------------------|-----------|
| Configure Transport System, e.g. as <i>Stand-Alone Transport Domain Entity</i> | STMS      |
| Create / Edit Profile for System                                               | RZ10      |
| Maintain Operation Modes and Instances                                         | RZ04      |

Este e todos os passos pós-instalação são amplamente descritos nos guias de instalação do **SAP**.

### 22.7.12.4. Edite o `initsid.sap` (`initIDS.sap`)

O arquivo `/oracle/IDS/dbs/initIDS.sap` contém a cópia de segurança do perfil do **SAP**. Aqui o tamanho da fita a ser usada, tipo de compressão e o que mais precisa ser definido. Para ter isto rodando com o `sapdba / brbackup`, nós mudamos os seguintes valores:

```
compress = hardware
archive_function = copy_delete_save
cpio_flags = "-ov --format=newc --block-size=128 --quiet"
cpio_in_flags = "-iuv --block-size=128 --quiet"
tape_size = 38000M
tape_address = /dev/nsa0
tape_address_rew = /dev/sa0
```

Explicações:

*compress*: A unidade de fita que usamos foi uma HP DLT1, que faz compressão por hardware.

*archive\_function*: Isto define o comportamento padrão do Oracle para a gravação de logs: novos arquivos de log são gravados na fita, arquivos de log anteriormente gravados são novamente gravados e depois apagados. Isto previne muitos problemas se você precisa recuperar a base de dados e uma das fitas deu problema.

*cpio\_flags*: O padrão é usar a opção `-B` que configura o tamanho dos blocos para 5120 Bytes. Para fitas DLT, a HP recomenda pelo menos 32 K de tamanho de blocos, então usamos a opção `--block-size=128` para 64 K. `--format=newc` é necessário pois temos números de inodo maiores que 65535. A última opção `--quiet` é necessário, caso contrário o `brbackup` reclamará assim que o `cpio` retorne o número de blocos gravados.

*cpio\_in\_flags*: Parâmetros necessários para carregar os dados de volta para a fita. O formato é reconhecido automaticamente.

*tape\_size*: Isto normalmente oferece a capacidade nativa de armazenamento da fita. Por razões de segurança (nós usamos compressão por hardware), o valor é discretamente menor do que o valor atual.

*tape\_address*: O dispositivo não- rebobinável a ser usado com o *cpio*.

*tape\_address\_rew*: O dispositivo rebobinável a ser usado com o *cpio*.

### 22.7.12.5. Questões de Configuração após a Instalação

Os seguintes parâmetros do **SAP** devem ser ligados depois da instalação (exemplos para o IDES 46B, 1 GB memory):

| Nome                  | Valor     |
|-----------------------|-----------|
| ztta/roll_extension   | 250000000 |
| abap/heap_area_dia    | 300000000 |
| abap/heap_area_nondia | 400000000 |
| em/initial_size_MB    | 256       |
| em/blocksize_kB       | 1024      |
| ipc/shm_psize_40      | 70000000  |

Nota SAP 0013026:

| Nome             | Valor   |
|------------------|---------|
| ztta/dynpro_area | 2500000 |

Nota SAP 0157246:

| Nome             | Valor |
|------------------|-------|
| rdisp/ROLL_MAXFS | 16000 |
| rdisp/PG_MAXFS   | 30000 |

**Nota:** Com os parâmetros acima, em um sistema com 1 gigabyte de memória, pode-se encontrar um consumo de memória similar a:

Mem: 547M Active, 305M Inact, 109M Wired, 40M Cache, 112M Buf, 3492K Free

## 22.7.13. Problemas durante a Instalação

### 22.7.13.1. Reinicialize o R3SETUP após Resolver um Problema

R3SETUP para se encontrar um erro. Se você olhou os arquivos de log correspondentes e consertou o erro, você tem que inicializar o R3SETUP novamente, normalmente selecionando REPEAT como uma opção para o último passo que o R3SETUP reclamou.

Para reinicializar o R3SETUP, apenas inicie com o arquivo R3S correspondente:

```
./R3SETUP -f CENTRDB.R3S
```

para o 4.6B, ou com

```
./R3SETUP -f CENTRAL.R3S
```

para o 4.6C, não importando o erro que tenha ocorrido com o CENTRAL.R3S ou DATABASE.R3S.

**Nota:** Em alguns estágios, o R3SETUP assume que ambos os processos do banco de dados e do **SAP** estão no ar e sendo executados (uma vez que aqueles passos foram completados). Erros podem ocorrer e, por exemplo, o banco de dados não pode ser inicializado, você tem que inicializar ambos na mão após ter consertado os erros e antes de iniciar o R3SETUP novamente.

Não esqueça também de iniciar o *listener* do **Oracle** novamente (como *orasid* com `umask 0; lsnrctl start`) se ele também foi parado (por exemplo, devido à uma reinicialização necessária do sistema).

### 22.7.13.2. OSUSERSIDADM\_IND\_ORA durante o R3SETUP

Se o R3SETUP reclamar neste estágio, edite o arquivo modelo R3SETUP usado em (CENTRDB.R3S (4.6B) ou CENTRAL.R3S ou DATABASE.R3S (4.6C)). Localize `[OSUSERSIDADM_IND_ORA]` ou procure apenas a entrada `STATUS=ERROR` e edite os seguintes valores:

```
HOME=/home/sidadm (was empty)
STATUS=OK (had status ERROR)
```

Então você pode reiniciar o R3SETUP novamente.

### 22.7.13.3. OSUSERDBSID\_IND\_ORA durante o R3SETUP

Possivelmente o R3SETUP também reclame neste estágio. O erro aqui é similar ao OSUSERSIDADM\_IND\_ORA. Apenas edite o arquivo modelo R3SETUP usado em (CENTRDB.R3S (4.6B) ou CENTRAL.R3S ou DATABASE.R3S (4.6C)). Localize

[*OSUSERDBSID\_IND\_ORA*] ou procure apenas a entrada *STATUS=ERROR* e edite os seguintes valores naquela seção:

```
STATUS=OK
```

Então reinicie o *R3SETUP*.

#### 22.7.13.4. ``oraview.vrf FILE NOT FOUND'' durante a Instalação do Oracle

Você não removeu a seleção *Oracle On-Line Text Viewer* antes de iniciar a instalação. Isto é marcado pela instalação mesmo que não esteja disponível para Linux. Desmarque esta opção dentro do menu de instalação do **Oracle** e reinicie a instalação.

#### 22.7.13.5. ``TEXTENV\_INVALID'' durante o *R3SETUP*, RFC ou SAPgui Start

Se este erro for encontrado, o local correto está faltando. A nota 0171356 do SAP lista os RPMs que precisam ser instalados (e.g. *saplocales-1.0-3*, *saposcheck-1.0-1* para o RedHat 6.1). Caso você tenha ignorado todos os erros relacionados e configurou o *STATUS* correspondente de *ERROR* para *OK* (em *CENTRDB.R3S*) e todas as vezes o *R3SETUP* reclamou e simplesmente reiniciou o *R3SETUP*, o sistema **SAP** não será configurado corretamente e incapaz de conectar-se a um sistema com **SAPgui**, mesmo que o sistema possa ser inicializado. Tentar conectar com o antigo **SAPgui** para Linux retornou as seguintes mensagens:

```
Sat May 5 14:23:14 2001
*** ERROR => no valid userarea given [trgmsg0. 0401]
Sat May 5 14:23:22 2001
*** ERROR => ERROR NR 24 occurred [trgmsgi. 0410]
*** ERROR => Error when generating text environment. [trgmsgi. 0435]
*** ERROR => function failed [trgmsgi. 0447]
*** ERROR => no socket operation allowed [trxio.c 3363]
Speicherzugriffsfehler
```

Este comportamento é devido ao **SAP R/3** ser incapaz de determinar corretamente o local e também de não estar corretamente configurado (entradas faltando em algumas tabelas do banco de dados). Para ser capaz de conectar ao **SAP**, adicione as seguintes entradas ao arquivo *DEFAULT.PFL* (veja Nota 0043288):

```
abap/set_etct_env_at_new_mode = 0
install/collate/active = 0
rscp/TCP0B = TCP0B
```

Reinicie o Sistema **SAP**. Agora você pode conectar-se ao sistema, embora a língua específica do país não funcionar como esperado. Após corrigir as configurações de país (e oferecer os locais corretos), estas entradas podem ser removidas de *DEFAULT.PFL* e o sistema **SAP** pode ser reinicializado.

#### 22.7.13.6. ORA-00001

Este erro ocorreu apenas com o **Oracle 8.1.7** no FreeBSD 4.5. A razão foi que a base de dados **Oracle** não inicializou corretamente e falhou, deixando semáforos e memória

compartilhada no sistema. A próxima tentativa de inicializar o banco de dados retornou o erro ORA-00001.

Encontre-os com `ipcs -a` e remova-os com `ipcrm`.

### 22.7.13.7. ORA-00445 (Background Process PMON Did Not Start)

Este erro aconteceu com o **Oracle 8.1.7**. Este erro é reportado se o banco de dados é inicializado com o *script* `startsap` (por exemplo `startsap_majestix_00`) como o usuário `prdadm`.

Uma solução temporária é inicializar o banco de dados com o usuário `oraprd` ao invés de `svrmgrl`:

```
% svrmgrl
SVRMGR> connect internal;
SVRMGR> startup;
SVRMGR> exit
```

### 22.7.13.8. ORA-12546 (Start Listener with Correct Permissions)

Inicie o **Oracle listener** como o usuário `oraids` com os seguintes comandos:

```
umask 0; lsnrctl start
```

Caso contrário você pode ter o erro ORA-12546, pois os *sockets* não possuirão as permissões corretas. Veja a Nota SAP 0072984.

### 22.7.13.9. ORA-27102 (Out of Memory)

Este erro aconteceu enquanto se tentava usar valores maiores que 1 GB (1024x1024x1024) para *MAXDSIZ* e *DFLDSIZ*. Adicionalmente, obtivemos o erro `Linux Error 12: Cannot allocate memory`.

### 22.7.13.10. [DIPGNTAB\_IND\_IND] durante o R3SETUP

Em geral, veja Nota SAP 0130581 (o passo *DIPGNTAB* do *R3SETUP* é terminado). Durante instalação específica do IDES, por algumas razões o processo de instalação não estava usando o nome apropriado do sistema `IDS` do **SAP**, mas estava usando uma *string* `""`. Isto leva a alguns pequenos problemas, pois os caminhos são gerados dinamicamente usando *SID* (IDES neste caso). Então ao invés de acessar:

```
/usr/sap/IDS/SYS/...
/usr/sap/IDS/DVMGS00
```

os seguintes caminhos foram usados:

```
/usr/sap//SYS/...
/usr/sap/D00
```

Para continuar com a instalação, criamos uma ligação e um diretório adicional:

```

pwd
/compat/linux/usr/sap
ls -l
total 4
drwxr-xr-x 3 idsadm sapsys 512 May 5 11:20 D00
drwxr-x--x 5 idsadm sapsys 512 May 5 11:35 IDS
lrwxr-xr-x 1 root sapsys 7 May 5 11:35 SYS -> IDS/SYS
drwxrwxr-x 2 idsadm sapsys 512 May 5 13:00 tmp
drwxrwxr-x 11 idsadm sapsys 512 May 4 14:20 trans

```

Nós também encontramos as notas SAP (0029227 e 0008401) descrevendo este comportamento. Não encontramos este tipo de problema na instalação do **SAP 4.6C**.

### 22.7.13.11. [RFCRSWBOINI\_IND\_IND] durante o R3SETUP

Durante a instalação do **SAP 4.6C**, este erro foi apenas o resultado de outro erro acontecido antes, durante a instalação. Neste caso, você precisa olhar os arquivos de log correspondentes e corrigir o problema real.

Se depois de olhar os arquivos de log, este erro for o correto (verifique as Notas SAP), você pode configurar o *STATUS* do passo problemático de *ERROR* para *OK* (arquivo `CENTRDB.R3S`) e reiniciar o `R3SETUP`. Após a instalação, você tem que executar o relatório *RSWBOINS* da transação `SE38`. Veja a Nota SAP 0162266 para informações adicionais a respeito da fase *RFCRSWBOINI* e *RFCRADDBDIF*.

### 22.7.13.12. [RFCRADDBDIF\_IND\_IND] durante o R3SETUP

Aqui as mesmas restrições se aplicam: verifique os arquivos de log, este erro não é causado por problemas anteriores.

Se você puder confirmar que a Nota SAP 0162266 se aplica, configure o *STATUS* do passo problemático de *ERROR* para *OK* (arquivo `CENTRDB.R3S`) e reinicie o `R3SETUP`. Após a instalação, você tem que executar o relatório *RADDBDIF* da transação `SE38`.

### 22.7.13.13. sigaction sig31: File size limit exceeded

Este erro ocorreu durante a inicialização dos processos *disp+work* do **SAP**. Se inicializarmos o **SAP** com o *script startsap*, então os subprocessos são inicializados sem acoplamento e fazem o trabalho sujo de inicializar todos os outros processos do **SAP**. Como resultado, o *script* por si só não reportará nada se algo der errado.

Para verificar se os processos do **SAP** inicializaram corretamente, você tem que olhar a situação dos processos com `ps ax | grep SID`, que lhe dará uma lista de todos os processos **Oracle** e **SAP**. Se parecer que alguns processos estão faltando ou se você não conseguir conectar ao sistema **SAP**, verifique os arquivos de log correspondentes, que podem ser encontrados em `/usr/sap/SID/DVEBMGSnr/work/`. Os arquivos a serem verificados são `dev_ms` e `dev_disp`.

``Signal 31" acontece aqui se a quantidade de memória compartilhada usada pelo **Oracle** e pelo **SAP** excederem o que foi definido no arquivo de configuração de *kernel*, podendo ser resolvido usando-se um valor maior:

```
larger value for 46C production systems:
options SHMMAXPGS=393216
smaller value sufficient for 46B:
#options SHMMAXPGS=262144
```

### 22.7.13.14. A inicialização do `saposcol` Falhou

Existem alguns problemas com o programa `saposcol` (versão 4.6D). O sistema **SAP** está usando o `saposcol` para colher dados sobre a performance do sistema. Este programa não é necessário no uso do sistema **SAP**, assim este problema pode ser considerado pequeno. As versões antigas (4.6B) funcionam, mas não colhem todos os dados (algumas chamadas retornarão apenas 0, por exemplo o uso da CPU).

---

## 22.8. Tópicos Avançados

Se você está curioso para saber como a compatibilidade binária Linux funciona, esta é a seção que você quer ler. Muito do que se segue é pesadamente baseado em um mensagem de correio escrita para [lista de discussão FreeBSD bate papo \(chat\)](#) por Terry Lambert <[tlambert@primenet.com](mailto:tlambert@primenet.com)> (Message ID: <199906020108.SAA07001@usr09.primenet.com>).

### 22.8.1. Como Funciona?

O FreeBSD possui uma abstração chamada "execution class loader". Isto é uma espécie de cunha na chamada de sistema [execve\(2\)](#).

Acontece que o FreeBSD possui uma lista de carregadores, ao invés de apenas um carregador que volta para o carregador `#!` para executar qualquer interpretador de linha de comando ou *shell scripts*.

Historicamente, o único carregador na plataforma UNIX® que examinou o número mágico (geralmente os primeiros 4 ou 8 bytes do arquivo) para ver se era um binário conhecido do sistema, e se fosse, invocaria o carregador do binário.

Se não fosse o tipo de binário para o sistema, o [execve\(2\)](#) retornaria uma falha, e o *shell* tentaria executá-lo como um comando.

Assumiasse que o padrão era "qualquer que fosse o shell".

Mais tarde, um ajuste foi feito no [sh\(1\)](#) para examinar os dois primeiros caracteres, e se eles fossem `:n`, então o *shell* [csh\(1\)](#) seria invocado (acreditamos a SCO foi a primeira a fazer este ajuste).

O que o FreeBSD faz agora é ir à uma lista de carregadores, com um carregador `#!` genérico que sabe a respeito dos interpretadores como os caracteres que seguem o próximo espaço em branco próximo ao último caractere, seguidos de um retorno para `/bin/sh`.

Para o suporte ABI pra Linux, o FreeBSD vê o número mágico como um binário ELF (não faz distinção entre FreeBSD, Solaris™, Linux ou qualquer outro SO que possui um tipo de imagem ELF, neste momento).

O carregador ELF procura por uma *marca* especializada, que é uma seção de comentário na imagem ELF, que não está presente nos binários ELF do SVR4/Solaris.

Para que os binários Linux funcionem, eles devem estar *marcados* como tipo *Linux* de [brandelf\(1\)](#):

```
brandelf -t Linux file
```

Quando isto estiver feito, o carregador ELF verá a marca *Linux* no arquivo.

Quando o carregador ELF vê uma marca *Linux*, ele substitui um ponteiro na estrutura do *proc*. Todas as chamadas de sistema indexadas através deste ponteiro (em um sistema UNIX tradicional, que pode ser a estrutura de *array sysent[]*, contendo as chamadas de sistema). Adicionalmente, o processo é marcado para uma manipulação especial do vetor para um código de sinal trampolim, e uma série de outros (pequenos) ajustes que são manipulados pelo módulo *kernel* do Linux.

O vetor de chamada Linux contém, entre outras coisas, uma lista de entradas *sysent[]* as quais seus endereços residem no módulo de *kernel*.

Quando uma chamada de sistema é feita por um binário Linux, o código de armadilha diferencia o ponteiro da função de chamada de sistema como estando fora da estrutura *proc*, e pega os pontos de entrada de chamada de sistema Linux, e não FreeBSD.

Além disso, o modo Linux dinamicamente *redireciona* as procuras; que é, na verdade, o que a opção *union* faz para as montagens de sistema de arquivo (*não* o tipo de sistema de arquivo *unionfs!*). Primeiro, uma tentativa é feita no diretório

`/compat/linux/original-path`, *então* se falhar, a procura é feita no diretório `/original-path`. Isto certifica que os binários que requerem outros binários possam ser executados (e.g., o *toolchain* Linux pode rodar sob o suporte ABI Linux). Isto também significa que os binários Linux pode carregar e executar binários FreeBSD, se não binários Linux correspondentes presentes, e que e que você possa colocar um comando [uname\(1\)](#) na árvore do diretório `/compat/linux` para ter certeza de que os binários Linux não possam dizer que eles não estão rodando no Linux.

Efetivamente, existe um *kernel* Linux dentro no *kernel* do FreeBSD; as mais variadas funções internas que implementam todos os serviços oferecidos pelo *kernel* são idênticos de ambas as entradas das tabelas de chamadas de sistema do FreeBSD, e as entradas das tabelas de chamada do Linux: operações de sistema de arquivo, operações de memória virtual, entrega de sinais, IPC do SystemV, etc... A única diferença é que os binários do FreeBSD obtém funções ```cola''` FreeBSD e binários Linux obtém funções ```cola''` Linux (SO's muito mais antigos possuem suas próprias funções ```cola''`: endereçamento de funções em uma estrutura *array sysent[]* estática global, ao invés de endereçamento de funções não referenciadas de um ponteiro dinamicamente inicializado na estrutura *proc* da realização da chamada do processo).



Qual é a ABI nativa do FreeBSD? Não importa. Basicamente a única diferença é que (atualmente; isso pode ser facilmente modificado em uma versão futura, e provavelmente será depois disso) as funções ``cola" FreeBSD são estaticamente ligadas no *kernel*, e as funções ``cola" do Linux podem ser estaticamente ligadas, ou podem ser acessadas através de um módulo de *kernel*.

Sim, mas isso realmente é emulação? Não. É uma implementação ABI. Não existe emulador envolvido (ou simulador, para cortar a próxima questão).

Então por que é chamado de ``emulação Linux"? Para fazer com que seja difícil vender o FreeBSD! Verdade, é porque a implementação histórica foi feita numa época quando não havia outra palavra além desta para descrever o que estava acontecendo; dizer que o FreeBSD executava binários Linux não era verdade, se você não compilou o código ou carregou o módulo, então existia uma expressão para descrever o que estava sendo executado-- ``o emulador Linux".

---

## Apêndice B. Bibliografia

### Índice

- B.1. [Livros & Revistas Específicos sobre FreeBSD](#)
- B.2. [Guias para Usuários](#)
- B.3. [Guias para Administradores](#)
- B.4. [Guias para Programadores](#)
- B.5. [Detalhes Internos do Sistema Operacional](#)
- B.6. [Referente a Segurança](#)
- B.7. [Referente a Hardware](#)
- B.8. [História do UNIX®](#)
- B.9. [Revistas e Periódicos](#)

Enquanto as páginas de manual fornecem a referência definitiva para partes individuais do sistema operacional FreeBSD, elas são notórias por não mostrar como juntar as peças para fazer todo o sistema operacional executar bem. Para isto, não existe substituto para um bom livro sobre administração de UNIX® e um bom manual de usuário.

## B.1. Livros & Revistas Específicos sobre FreeBSD

*Livros & Revistas internacionais :*

- [Using FreeBSD](#) (em Chinês).
- FreeBSD for PC 98'ers (em Japonês), publicado por SHUWA System Co, LTD. ISBN 4-87966-468-5 C3055 P2900E.

- FreeBSD (em Japonês), publicado por CUTT. ISBN 4-906391-22-2 C3055 P2400E.
- [Complete Introduction to FreeBSD](#) (em Japonês), publicado por [Shoeisha Co., Ltd.](#) ISBN 4-88135-473-6 P3600E.
- [Personal UNIX Starter Kit FreeBSD](#) (em Japonês), publicado por [ASCII](#). ISBN 4-7561-1733-3 P3000E.
- FreeBSD Handbook (tradução Japonesa), publicado por [ASCII](#). ISBN 4-7561-1580-2 P3800E.
- FreeBSD mit Methode (em Alemão), publicado por [Computer und Literatur Verlag/Vertrieb Hanser](#), 1998. ISBN 3-932311-31-0.
- [FreeBSD 4 - Installieren, Konfigurieren, Administrieren](#) (em Alemão), publicado por [Computer und Literatur Verlag](#), 2001. ISBN 3-932311-88-4.
- [FreeBSD 5 - Installieren, Konfigurieren, Administrieren](#) (em Alemão), publicado por [Computer und Literatur Verlag](#), 2003. ISBN 3-936546-06-1.
- [FreeBSD de Luxe](#) (em Alemão), publicado por [Verlag Modere Industrie](#), 2003. ISBN 3-8266-1343-0.
- [FreeBSD Install and Utilization Manual](#) (em Japonês), publicado por [Mainichi Communications Inc.](#).
- Onno W Purbo, Dodi Maryanto, Syahrial Hubbany, Widjil Widodo [Building Internet Server with FreeBSD](#) (in Indonesia Language), publicado por [Elex Media Komputindo](#).

*Livros & Revistas em inglês:*

- [Absolute BSD: The Ultimate Guide to FreeBSD](#), publicado por [No Starch Press](#), 2002. ISBN: 1886411743
- [The Complete FreeBSD](#), publicado por [O'Reilly](#), 2003. ISBN: 0596005164
- [The FreeBSD Corporate Networker's Guide](#), publicado por [Addison-Wesley](#), 2000. ISBN: 0201704811
- [FreeBSD: An Open-Source Operating System for Your Personal Computer](#), publicado por The Bit Tree Press, 2001. ISBN: 0971204500
- Teach Yourself FreeBSD in 24 Hours, publicado por [Sams](#), 2002. ISBN: 0672324245
- FreeBSD unleashed, publicado por [Sams](#), 2002. ISBN: 0672324563
- FreeBSD: The Complete Reference, publicado por [McGrawHill](#), 2003. ISBN: 0072224096

## Apêndice C. Recursos na Internet

### Índice

- C.1. [Listas de Discussão](#)
- C.2. [Usenet Newsgroups](#)
- C.3. [World Wide Web Servers](#)
- C.4. [Email Addresses](#)
- C.5. [Shell Accounts](#)

O ritmo acelerado de progresso do FreeBSD torna a mídia impressa pouco prática para acompanhar os últimos desenvolvimentos. Os recursos eletrônicos são o melhor, e freqüentemente os únicos, meios de se manter informado sobre os últimos avanços. Sendo o FreeBSD um esforço voluntário, a própria comunidade de usuários geralmente atua como "departamento de suporte técnico" of sorts, com o correio eletrônico e anúncios da USENET sendo os mais eficientes meios de alcançar esta comunidade.

Os mais importantes pontos de contato com a comunidade de usuários do FreeBSD são mostrados abaixo. Se você estiver ciente de novos recursos que não são mencionados aqui, por favor envie-os para [lista de discussão do projeto de documentação do FreeBSD](#) de modo que possam também se incluídos.

## C.1. Listas de Discussão

Embora muitos desenvolvedores do FreeBSD leiam a USENET, nós não podemos garantir que teremos acesso às suas perguntas no momento oportuno (ou em qualquer um) se você postá-la somente em um dos grupos *comp.unix.bsd.freebsd.\**. Endereçando suas questões para a lista de discussão apropriada você alcançará a nós e a uma audiência concentrada em FreeBSD, assegurando, invariavelmente, uma melhor resposta (ou ao menos mais rápida).

As normas para as várias listas são dadas mais a frente neste documento. *Por favor, leia-as antes de inscrever-se ou enviar um e-mail para qualquer lista.* A maioria dos inscritos em nossas listas recebem centenas de mensagens relacionadas a FreeBSD todos os dias, e estabelecendo normas e regras para uso apropriado nós estamos nos esforçando em manter a relação sinal/ruído destas listas alta. Negligenciar estes pontos pode, em última análise, levar a que as listas falhem como um meio efetivo de comunicação para o projeto.

Arquivos são mantidos para todas as listas e podem ser pesquisados utilizando o [servidor FreeBSD WWW](#). O arquivo, pesquisável por palavras chaves, oferece uma maneira excelente de encontrar perguntas feitas com freqüência e deve ser consultado antes de se postar uma questão.

### C.1.1. Sumário das Listas

*Listas gerais:* As seguintes listas são gerais. Qualquer um é livre (e encorajado) a participar:

| Lista                            | Propósito                                             |
|----------------------------------|-------------------------------------------------------|
| <a href="#">cvs-all</a>          | Modificações efetuadas na árvore de fontes do FreeBSD |
| <a href="#">freebsd-advocacy</a> | Evangelização do FreeBSD                              |
| <a href="#">freebsd-announce</a> | Eventos importantes e marcos do projeto               |

| <b>Lista</b>                                   | <b>Propósito</b>                                                                                                       |
|------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| <a href="#">freebsd-arch</a>                   | Discussões sobre arquitetura e design                                                                                  |
| <a href="#">freebsd-bugbusters</a>             | Discussões pertinentes a manutenção do banco de dados de relatórios de problemas do FreeBSD e ferramentas relacionadas |
| <a href="#">freebsd-bugs</a>                   | Relatórios de erros                                                                                                    |
| <a href="#">freebsd-chat</a>                   | Itens não técnicos relacionados a comunidade FreeBSD                                                                   |
| <a href="#">freebsd-config</a>                 | Desenvolvimento de ferramentas de instalação e configuração do FreeBSD                                                 |
| <a href="#">freebsd-current</a>                | Discussões concernentes ao uso do FreeBSD-CURRENT                                                                      |
| <a href="#">freebsd-isp</a>                    | Assuntos para provedores de serviço de internet utilizando FreeBSD                                                     |
| <a href="#">freebsd-jobs</a>                   | Oportunidades de empregos e consultorias em FreeBSD                                                                    |
| <a href="#">freebsd-newbies</a>                | Atividades e discussões para novos usuários de FreeBSD                                                                 |
| <a href="#">freebsd-policy</a>                 | Decisões políticas do Core team do FreeBSD. Baixo volume de mensagens e somente leitura                                |
| <a href="#">freebsd-questions</a>              | Questões de usuários e suporte técnico                                                                                 |
| <a href="#">freebsd-security-notifications</a> | Notificações de segurança                                                                                              |
| <a href="#">freebsd-stable</a>                 | Discussões concernentes ao uso do FreeBSD-STABLE                                                                       |
| <a href="#">freebsd-test</a>                   | Para onde enviar suas mensagens de teste ao invés das listas atuais                                                    |

*Listas técnicas:* As seguintes listas são para discussões técnicas. Você deve ler as normas para cada lista cuidadosamente antes de participar ou enviar e-mail uma vez que há regras rígidas para seu uso e conteúdo.

| <b>Lista</b>                    | <b>Propósito</b>                                    |
|---------------------------------|-----------------------------------------------------|
| <a href="#">freebsd-afs</a>     | Convertendo AFS para o FreeBSD                      |
| <a href="#">freebsd-aic7xxx</a> | Desenvolvimento de drivers para a Adaptec® AIC 7xxx |
| <a href="#">freebsd-alpha</a>   | Convertendo FreeBSD para a Alpha                    |

| <b>Lista</b>                      | <b>Propósito</b>                                                       |
|-----------------------------------|------------------------------------------------------------------------|
| <a href="#">freebsd-amd64</a>     | Convertendo FreeBSD para sistemas AMD64                                |
| <a href="#">freebsd-arm</a>       | Convertendo FreeBSD para processadores ARM®                            |
| <a href="#">freebsd-atm</a>       | Usando redes ATM com FreeBSD                                           |
| <a href="#">freebsd-audit</a>     | Projeto de auditoria de código fonte                                   |
| <a href="#">freebsd-binup</a>     | Desenho e desenvolvimento do sistema de atualizações de binários       |
| <a href="#">freebsd-cluster</a>   | Usando FreeBSD em ambiente de cluster                                  |
| <a href="#">freebsd-cvsweb</a>    | Manutenção de CVSweb                                                   |
| <a href="#">freebsd-database</a>  | Discutindo utilização e desenvolvimento de bancos de dados sob FreeBSD |
| <a href="#">freebsd-doc</a>       | Criando documentos relacionados a FreeBSD                              |
| <a href="#">freebsd-emulation</a> | Emulando outros sistemas como Linux/DOS/Windows®                       |
| <a href="#">freebsd-firewire</a>  | Discussão técnica sobre FreeBSD FireWire® (iLink, IEEE 1394)           |
| <a href="#">freebsd-fs</a>        | Sistemas de arquivos                                                   |
| <a href="#">freebsd-gnome</a>     | Convertendo <b>GNOME</b> e aplicações do <b>GNOME</b>                  |
| <a href="#">freebsd-hackers</a>   | Discussões técnicas gerais                                             |
| <a href="#">freebsd-hardware</a>  | Discussões gerais de hardware para executar o FreeBSD                  |
| <a href="#">freebsd-i18n</a>      | Internacionalização do FreeBSD                                         |
| <a href="#">freebsd-ia32</a>      | FreeBSD na plataforma IA-32 (Intel® x86)                               |
| <a href="#">freebsd-ia64</a>      | Convertendo o FreeBSD para os futuros sistemas Intel IA64              |
| <a href="#">freebsd-ipfw</a>      | Discussões técnicas concernentes ao redesenho do código do IP firewall |
| <a href="#">freebsd-isdn</a>      | Desenvolvedores de ISDN                                                |
| <a href="#">freebsd-java</a>      | Java™ desenvolvedores e pessoas portando JDK™s para FreeBSD            |
| <a href="#">freebsd-kde</a>       | Convertendo <b>KDE</b> e aplicações para <b>KDE</b>                    |
| <a href="#">freebsd-lfs</a>       | Convertendo o LFS para o FreeBSD                                       |

| <b>Lista</b>                        | <b>Propósito</b>                                                                     |
|-------------------------------------|--------------------------------------------------------------------------------------|
| <a href="#">freebsd-libh</a>        | A segunda geração de sistemas de pacotes e instalação                                |
| <a href="#">freebsd-mips</a>        | Convertendo FreeBSD para o MIPS®                                                     |
| <a href="#">freebsd-mobile</a>      | Discussões sobre computação móvel                                                    |
| <a href="#">freebsd-mozilla</a>     | Convertendo <b>Mozilla</b> para FreeBSD                                              |
| <a href="#">freebsd-multimedia</a>  | Aplicações multemídia                                                                |
| <a href="#">freebsd-new-bus</a>     | Discussões técnicas sobre a arquitetura bus                                          |
| <a href="#">freebsd-net</a>         | Discussões sobre redes e código TCP/IP                                               |
| <a href="#">freebsd-openoffice</a>  | Convertendo <b>OpenOffice.org</b> e <b>StarOffice™</b> para FreeBSD                  |
| <a href="#">freebsd-performance</a> | Questões de performance para instalações de alta performance/carga                   |
| <a href="#">freebsd-platforms</a>   | <i>Ports</i> relativos a plataformas de arquitetura não Intel                        |
| <a href="#">freebsd-ports</a>       | Discussões sobre a coleção de <i>ports</i>                                           |
| <a href="#">freebsd-ports-bugs</a>  | Discussões sobre <i>bugs</i> /PRs nos <i>ports</i>                                   |
| <a href="#">freebsd-ppc</a>         | Convertendo o FreeBSD para o PowerPC®                                                |
| <a href="#">freebsd-qa</a>          | Discussões sobre Certificação de Qualidade, geralmente associada a um <i>release</i> |
| <a href="#">freebsd-realtime</a>    | Desenvolvimento de extensões <i>realtime</i> para FreeBSD                            |
| <a href="#">freebsd-scsi</a>        | O subsistema SCSI                                                                    |
| <a href="#">freebsd-security</a>    | Problemas de segurança afetando FreeBSD                                              |
| <a href="#">freebsd-small</a>       | Utilizando FreeBSD em aplicações embarcadas                                          |
| <a href="#">freebsd-smp</a>         | Discussões de desenho para multiprocessamento [a]ssimétrico                          |
| <a href="#">freebsd-sparc64</a>     | Convertendo FreeBSD para sistemas baseados em Sparc®                                 |
| <a href="#">freebsd-standards</a>   | Conformidade com o C99 e os padrões POSIX®                                           |
| <a href="#">freebsd-threads</a>     | <i>Threading</i> no FreeBSD                                                          |

| Lista                             | Propósito                                       |
|-----------------------------------|-------------------------------------------------|
| <a href="#">freebsd-testing</a>   | Testes de performance e estabilidade no FreeBSD |
| <a href="#">freebsd-tokenring</a> | Suporte a <i>Token Ring</i> no FreeBSD          |
| <a href="#">freebsd-x11</a>       | Manutenção e suporte ao X11 no FreeBSD          |

*Listas limitadas:* As seguintes listas são para audiências mais especializadas (e exigentes) e provavelmente não são de interesse para o público geral. É também uma boa idéia estabelecer presença em listas técnicas antes de juntar-se a uma destas lista limitadas de modo que você possa compreender a etiqueta de comunicações envolvida.

| Lista                               | Propósito                                                            |
|-------------------------------------|----------------------------------------------------------------------|
| <a href="#">freebsd-hubs</a>        | Pessoas executando sites espelhos (suporte infraestrutural)          |
| <a href="#">freebsd-user-groups</a> | Coordenação de grupos de usuários                                    |
| <a href="#">freebsd-vendors</a>     | Coordenação de vendedores de pre-lançamento                          |
| <a href="#">freebsd-www</a>         | Mantenedores do <a href="http://www.FreeBSD.org">www.FreeBSD.org</a> |

*Listas sumário:* Todas as listas acima estão disponíveis em formato *digest*. Uma vez inscrito em uma lista, você pode modificar suas opções *digest* na seção de opções de sua conta.

*Listas CVS:* As seguintes listas são para pessoas interessadas em observar as mensagens de log para as modificações nas várias áreas da árvore do código fonte. Elas são listas *somente para leitura* não existe correio enviado a elas.

| Lista                     | Área fonte                                         | Área de descrição (fonte para)                                                                        |
|---------------------------|----------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| <a href="#">cvs-all</a>   | <code>/usr/(CVSROOT doc ports projects src)</code> | Todas as modificações em qualquer lugar da árvore (supergrupo de outras listas de <i>cvs commit</i> ) |
| <a href="#">cvs-doc</a>   | <code>/usr/doc</code>                              | Todas as mudanças na árvore de documentação                                                           |
| <a href="#">cvs-ports</a> | <code>/usr/ports</code>                            | Todas as mudanças na árvore do sistema de <i>ports</i>                                                |
| <a href="#">cvs-</a>      | <code>/usr/projects</code>                         | Todas as mudanças na árvore de                                                                        |

| Lista                    | Área fonte | Área de descrição (fonte para)        |
|--------------------------|------------|---------------------------------------|
| <a href="#">projects</a> |            | projetos                              |
| <a href="#">cvs-src</a>  | /usr/src   | Todas as mudanças na árvore de fontes |

### C.1.2. Como se inscrever

Para se inscrever em uma lista, clique no nome da lista acima ou vá para <http://lists.FreeBSD.org/mailman/listinfo> e clique na lista a qual está interessado. A página de listas contém todas as instruções necessárias para a inscrição.

Atualmente, para postar em uma dada lista você simplesmente envia um e-mail para <nome da lista@FreeBSD.org>. E ele então será redistribuído para os membros da lista ao redor do mundo.

Para se desinscrever de uma lista, clique na URL encontrada na base de cada e-mail recebido da lista. Também é possível enviar um e-mail para `freebsd-[nome da lista]-unsubscribe@FreeBSD.org` para se desinscrever.

Novamente, nós gostaríamos de pedir que você mantivesse a discussão em uma lista técnica, dentro de aspectos técnicos. Se você está interessado somente em anúncios importantes, então é sugerido que você se junte a [lista de discussão dos anúncios FreeBSD](#), a qual é voltada somente para baixo tráfego.

### C.1.3. Características das listas

Todas as lista de correspondência do FreeBSD possuem certas regras básicas que devem ser aderidas por qualquer um que as utilize. O não cumprimento destas regras irá resultar em duas (2) mensagens de aviso do Postmaster do FreeBSD <[postmaster@FreeBSD.org](mailto:postmaster@FreeBSD.org)>, após o que, em uma terceira ofensa, aquele que postou será removido de todas as listas de e-mail do FreeBSD e filtrado em futuras postagens. Lastimamos que regras e medidas como estas sejam necessárias, mas a Internet atual é um ambiente um tanto ruidoso, isto pode ser visto, e muitos falham em apreciar o quão frágeis são alguns mecanismos.

Regras da Estrada:

- O tópico de qualquer postagem deve aderir a característica base da lista em que é postado, i.e. se a lista versa sobre assuntos técnicos, então sua postagem deve conter uma discussão técnica. Levar conversas irrelevantes ou fazer *flame* apenas subtrai valor da lista de discussão para qualquer um e não será telerado. Para discussões livres em nenhum tópico em particular, a [lista de discussão FreeBSD bate papo \(chat\)](#) é disponível livremente e deve ser usada em alternativa.
- Nenhuma postagem deve ser feita para mais de duas listas, e somente para duas quando existir uma necessidade clara e óbvia de postar para ambas. Para a



maioria das listas, existe grande redundância de assinantes e exceto para as misturas mais exóticas (algo como ``-stable & -scsi"), realmente não existe razão para postar em mais de uma lista ao mesmo tempo. Caso um mensagem seja enviada para você de maneira que múltiplas listas apareçam na linha de *Cc* então a linha *Cc* deve ser removida antes de remetê-la novamente. *Você continua responsável pelas suas postagens cruzadas, não interessando quem tenha começado.*

- Ataques pessoais e profanação (no contexto de um argumento) não são permitidos, e isso inclui usuários e desenvolvedores igualmente. Burlas excessivas de ``netiqueta", como fazer excertos ou reenviar mensagens privadas quando não há ou não aparenta haver permissão para isso, são desagradáveis, mas não especificamente coagidas. *Contudo*, existem muito poucos casos onde este conteúdo pode adequar-se às regras da lista e provavelmente poderá gerar um alerta (ou banimento) por si mesmo.
- Propaganda de produtos e serviços não relacionados a FreeBSD é estritamente proibida e irá resultar em banimento imediato caso fique claro que o ofensor está fazendo propaganda através de *spam*.

*Regras específicas a listas:*

#### [freebsd-afs](#)

*Sistemas de Arquivos Andrew*

Esta lista é para a discussão em converter e utilizar o AFS de CMU/Transarc

#### [freebsd-announce](#)

*Eventos e marcos importantes*

Esta é a lista para as pessoas interessadas somente em anúncios ocasionais de eventos significantes no FreeBSD. Esta inclui anúncios sobre *snapshots* e outros lançamentos. Ela contém anúncios de novas capacidades do FreeBSD. Ela pode apresentar chamadas para voluntários etc. Esta é uma lista de baixo volume, estritamente moderada.

#### [freebsd-arch](#)

*Discussões sobre arquitetura e design*

Esta lista é para discussões sobre a arquitetura do FreeBSD. A maioria das mensagens devem manter sua natureza estritamente técnica. Exemplos de tópicos possíveis são:

- Como revampirizar a construção do sistema para ter muitas construções customizadas rodando ao mesmo tempo.
- O que precisa ser concertado no VSF para fazer as camadas Heidemann funcionarem.
- Como mudar a interface de *drivers* de dispositivo para poder usar os mesmos *drivers* em diversos *buses* e arquiteturas.

- Como escrever um *driver* de rede.

#### [freebsd-audit](#)

##### *Projeto de auditoria de código fonte*

Esta é a lista para o projeto de auditoria de código fonte do FreeBSD. Embora esta seja originalmente voltada para modificações relacionadas a segurança, seu caráter foi expandido para revisar qualquer modificação do código.

Esta lista é carregada de correções, e provavelmente não interessa ao usuário médio do FreeBSD. Discussões de segurança não relacionadas a uma mudança em um código particular são retidas para a *freebsd-security*. Inversamente, todos os desenvolvedores são encorajados a enviarem seus *patches* aqui para revisão, especialmente se eles tocam em uma parte do sistema onde um defeito pode afetar adversamente a integridade do sistema.

#### [freebsd-binup](#)

##### *Projeto de atualização binária do FreeBSD*

Esta lista existe para prover discussão para o sistema de atualização binária, ou **binup**. Questões de desenho, detalhes de implementação, *patches*, relatórios de erros, relatórios de situação, requisição de características, registros de *commit*, e todas as demais coisas relacionadas a **binup** são bem aceitas.

#### [freebsd-bugbusters](#)

##### *Coordenação de esforços para o tratamento de problemas reportados*

O propósito desta lista é servir como coordenação e fórum de discussões para o *Bugmeister*, seus *Bugbusters*, e outras partes que tem o genuíno interesse na base de PR. Esta lista não é para discussões sobre problemas específicos, *patches* ou PRs.

#### [freebsd-bugs](#)

##### *Registro de problemas*

Esta é a lista para reportar defeitos no FreeBSD. Sempre que possível, problemas devem ser submetidos usando o comando [send-pr\(1\)](#) ou a [interface WEB](#) para isso.

#### [freebsd-chat](#)

##### *Itens não técnicos relacionados a comunidade FreeBSD*

Esta lista contém a sobrecarga das outras listas sobre informação social e não técnica. Inclui discussões sobre se Jordan parece ou não um furão dos desenhos

animados, quando digitar em maiúsculas ou não, quem está exagerando no café, onde a melhor cerveja é fermentada, quem está fazendo cerveja no porão, e por aí vai. Anúncios casuais de eventos importantes (como festas, casamentos, nascimentos, novos empregos, etc) podem ser feitos nas listas técnicas, mas suas considerações subseqüentes devem ser direcionadas para esta lista de -papo.

[freebsd-core](#)

#### *FreeBSD core team*

Esta é uma lista interna para uso de membros do *core*. Mensagens podem ser enviadas para esta lista quando um assunto sério relacionado a FreeBSD requer o arbítrio a partir de uma inspeção de alto nível.

[freebsd-current](#)

#### *Discussões sobre o uso do FreeBSD-CURRENT*

Esta é a lista para usuários do FreeBSD-CURRENT. Ela inclui alertas sobre novas características surgindo no -CURRENT que irão afetar usuários, e instruções sobre os passos que precisam ser tomados para continuar -CURRENT. Qualquer um rodando ``CURRENT" deve assinar a esta lista. É uma lista técnica para a qual conteúdo estritamente técnico é esperado.

[freebsd-cvsweb](#)

#### *Projeto FreeBSD CVSWeb*

Discussões técnicas sobre o uso, desenvolvimento e manutenção do FreeBSD-CVSweb.

[freebsd-doc](#)

#### *Projeto de documentação*

Esta é a lista para discussão de questões e projetos relacionados a criação de documentação para FreeBSD. Os membros desta lista são coletivamente associados ao ``Projeto de Documentação FreeBSD". É uma lista aberta; sintase livre para entrar e contribuir!

[freebsd-firewire](#)

#### *FireWire (iLink, IEEE 1394)*

Esta é a lista para a discussão do desenho e implementação do subsistema FireWire (conhecido também como IEEE 1394 ou iLink) para FreeBSD. Tópicos relevantes incluem especificamente os padrões, dispositivos de barramento e seus protocolos, adaptadores placas/cartões/*chips sets*, e a arquitetura e implementação do código para seu suporte adequado.

### [freebsd-fs](#)

#### *Sistemas de arquivos*

Discussões concernentes aos sistemas de arquivos do FreeBSD. É uma lista técnica para a qual conteúdo extritamente técnico é esperado.

### [freebsd-gnome](#)

#### *GNOME*

Discussions concerning The **GNOME** Desktop Environment for FreeBSD systems. This is a technical mailing list for which strictly technical content is expected.

### [freebsd-ipfw](#)

#### *IP Firewall*

This is the forum for technical discussions concerning the redesign of the IP firewall code in FreeBSD. This is a technical mailing list for which strictly technical content is expected.

### [freebsd-ia64](#)

#### *Porting FreeBSD to IA64*

This is a technical mailing list for individuals actively working on porting FreeBSD to the IA-64 platform from Intel, to bring up problems or discuss alternative solutions. Individuals interested in following the technical discussion are also welcome.

### [freebsd-isdn](#)

#### *ISDN Communications*

This is the mailing list for people discussing the development of ISDN support for FreeBSD.

### [freebsd-java](#)

#### *Java Development*

This is the mailing list for people discussing the development of significant Java applications for FreeBSD and the porting and maintenance of JDKs.

### [freebsd-kde](#)

#### *KDE*

Discussions concerning **KDE** on FreeBSD systems. This is a technical mailing list for which strictly technical content is expected.

#### [freebsd-hackers](#)

##### *Technical discussions*

This is a forum for technical discussions related to FreeBSD. This is the primary technical mailing list. It is for individuals actively working on FreeBSD, to bring up problems or discuss alternative solutions. Individuals interested in following the technical discussion are also welcome. This is a technical mailing list for which strictly technical content is expected.

#### [freebsd-hardware](#)

##### *General discussion of FreeBSD hardware*

General discussion about the types of hardware that FreeBSD runs on, various problems and suggestions concerning what to buy or avoid.

#### [freebsd-hubs](#)

##### *Mirror sites*

Announcements and discussion for people who run FreeBSD mirror sites.

#### [freebsd-isp](#)

##### *Issues for Internet Service Providers*

This mailing list is for discussing topics relevant to Internet Service Providers (ISPs) using FreeBSD. This is a technical mailing list for which strictly technical content is expected.

#### [freebsd-newbies](#)

##### *Newbies activities discussion*

We cover any of the activities of newbies that are not already dealt with elsewhere, including: independent learning and problem solving techniques, finding and using resources and asking for help elsewhere, how to use mailing lists and which lists to use, general chat, making mistakes, boasting, sharing ideas, stories, moral (but not technical) support, and taking an active part in the FreeBSD community. We take our problems and support questions to freebsd-questions, and use freebsd-newbies to meet others who are doing the same things that we do as newbies.

#### [freebsd-openoffice](#)

*OpenOffice.org*

Discussions concerning the porting and maintenance of **OpenOffice.org** and **StarOffice**.

[freebsd-performance](#)

*Discussions about tuning or speeding up FreeBSD*

This mailing list exists to provide a place for hackers, administrators, and/or concerned parties to discuss performance related topics pertaining to FreeBSD. Acceptable topics includes talking about FreeBSD installations that are either under high load, are experiencing performance problems, or are pushing the limits of FreeBSD. Concerned parties that are willing to work toward improving the performance of FreeBSD are highly encouraged to subscribe to this list. This is a highly technical list ideally suited for experienced FreeBSD users, hackers, or administrators interested in keeping FreeBSD fast, robust, and scalable. This list is not a question-and-answer list that replaces reading through documentation, but it is a place to make contributions or inquire about unanswered performance related topics.

[freebsd-platforms](#)

*Porting to Non Intel platforms*

Cross-platform FreeBSD issues, general discussion and proposals for non Intel FreeBSD ports. This is a technical mailing list for which strictly technical content is expected.

[freebsd-policy](#)

*Core team policy decisions*

This is a low volume, read-only mailing list for FreeBSD Core Team Policy decisions.

[freebsd-ports](#)

*Discussion of ``ports''*

Discussions concerning FreeBSD's ``ports collection" (`/usr/ports`), ports infrastructure, and general ports coordination efforts. This is a technical mailing list for which strictly technical content is expected.

[freebsd-ports-bugs](#)

*Discussion of ``ports'' bugs*

Discussions concerning problem reports for FreeBSD's ``ports collection" (/usr/ports), proposed ports, or modifications to ports. This is a technical mailing list for which strictly technical content is expected.

### [freebsd-questions](#)

#### *User questions*

This is the mailing list for questions about FreeBSD. You should not send ``how to" questions to the technical lists unless you consider the question to be pretty technical.

### [freebsd-scsi](#)

#### *SCSI subsystem*

This is the mailing list for people working on the SCSI subsystem for FreeBSD. This is a technical mailing list for which strictly technical content is expected.

### [freebsd-security](#)

#### *Security issues*

FreeBSD computer security issues (DES, Kerberos, known security holes and fixes, etc). This is a technical mailing list for which strictly technical discussion is expected. Note that this is not a question-and-answer list, but that contributions (BOTH question AND answer) to the FAQ are welcome.

### [freebsd-security-notifications](#)

#### *Security Notifications*

Notifications of FreeBSD security problems and fixes. This is not a discussion list. The discussion list is FreeBSD-security.

### [freebsd-small](#)

#### *Using FreeBSD in embedded applications*

This list discusses topics related to unusually small and embedded FreeBSD installations. This is a technical mailing list for which strictly technical content is expected.

### [freebsd-stable](#)

#### *Discussions about the use of FreeBSD-STABLE*

This is the mailing list for users of FreeBSD-STABLE. It includes warnings about new features coming out in -STABLE that will affect the users, and

instructions on steps that must be taken to remain -STABLE. Anyone running ``STABLE" should subscribe to this list. This is a technical mailing list for which strictly technical content is expected.

#### [freebsd-standards](#)

##### *C99 & POSIX Conformance*

This is a forum for technical discussions related to FreeBSD Conformance to the C99 and the POSIX standards.

#### [freebsd-user-groups](#)

##### *User Group Coordination List*

This is the mailing list for the coordinators from each of the local area Users Groups to discuss matters with each other and a designated individual from the Core Team. This mail list should be limited to meeting synopsis and coordination of projects that span User Groups.

#### [freebsd-vendors](#)

##### *Vendors*

Coordination discussions between The FreeBSD Project and Vendors of software and hardware for FreeBSD.

### **C.1.4. Filtering on the Mailing Lists**

The FreeBSD mailing lists are filtered in multiple ways to avoid the distribution of spam, viruses, and other unwanted emails. The filtering actions described in this section do not include all those used to protect the mailing lists.

Only certain types of attachments are allowed on the mailing lists. All attachments with a MIME content type not found in the list below will be stripped before an email is distributed on the mailing lists.

- application/octet-stream
- application/pdf
- application/pgp-signature
- application/x-pkcs7-signature
- message/rfc822
- multipart/alternative
- multipart/related
- multipart/signed
- text/html
- text/plain
- text/x-diff
- text/x-patch



**Nota:** Some of the mailing lists might allow attachments of other MIME content types, but the above list should be applicable for most of the mailing lists.

If an email contains both an HTML and a plain text version, the HTML version will be removed. If an email contains only an HTML version, it will be converted to plain text.

---

## C.2. Usenet Newsgroups

In addition to two FreeBSD specific newsgroups, there are many others in which FreeBSD is discussed or are otherwise relevant to FreeBSD users. [Keyword searchable archives](#) are available for some of these newsgroups from courtesy of Warren Toomey <[wkt@cs.adfa.edu.au](mailto:wkt@cs.adfa.edu.au)>.

### C.2.1. BSD Specific Newsgroups

- [comp.unix.bsd.freebsd.announce](#)
- [comp.unix.bsd.freebsd.misc](#)
- [de.comp.os.unix.bsd](#) (German)
- [fr.comp.os.bsd](#) (French)

### C.2.2. Other UNIX® Newsgroups of Interest

- [comp.unix](#)
- [comp.unix.questions](#)
- [comp.unix.admin](#)
- [comp.unix.programmer](#)
- [comp.unix.shell](#)
- [comp.unix.user-friendly](#)
- [comp.security.unix](#)
- [comp.sources.unix](#)
- [comp.unix.advocacy](#)
- [comp.unix.misc](#)
- [comp.bugs.4bsd](#)
- [comp.bugs.4bsd.ucb-fixes](#)
- [comp.unix.bsd](#)

### C.2.3. X Window System

- [comp.windows.x.i386unix](#)
- [comp.windows.x](#)
- [comp.windows.x.apps](#)
- [comp.windows.x.announce](#)
- [comp.windows.x.intrinsics](#)
- [comp.windows.x.motif](#)
- [comp.windows.x.pex](#)
- [comp.emulators.ms-windows.wine](#)

## C.4. Email Addresses

The following user groups provide FreeBSD related email addresses for their members. The listed administrator reserves the right to revoke the address if it is abused in any way.

| Domain              | Facilities      | User Group                                                                         | Administrator                                                                  |
|---------------------|-----------------|------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| ukug.uk.FreeBSD.org | Forwarding only | < <a href="mailto:freebsd-users@uk.FreeBSD.org">freebsd-users@uk.FreeBSD.org</a> > | Lee Johnston<br>< <a href="mailto:lee@uk.FreeBSD.org">lee@uk.FreeBSD.org</a> > |

## Considerações finais

Este livro é a combinação do trabalho de centenas de colaboradores do "Projeto de Documentação do FreeBSD". O texto é elaborado em SGML de acordo com os DTD do DocBook e é formatado a partir do SGML para muitos formatos diferentes de apresentação utilizando o **Jade**, um mecanismo DSSSL de código aberto. As folhas de estilo do Norm Walsh são utilizadas com uma camada adicional de customização para fornecer as instruções de apresentação para o **Jade**. A versão impressa deste documento não seria possível sem a linguagem de composição **TeX** de autoria do Donald Knuth, sem o **LaTeX** do Leslie Lamport ou sem o pacote de macros **JadeTeX** do Sebastian Rahtz.

*Compilado em PDF por Xiiico.*  
[franciscoemars@gmail.com](mailto:franciscoemars@gmail.com)

**Compilação não oficial.**

**08/08/2009**