Perguntas mais freqüentes sobre FreeBSD 2.X, 3.X e 4.X

Perguntas mais freqüentes sobre FreeBSD 2.X, 3.X e 4.X

Revisão: 43184
Copyright © 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002 Projeto de Documentação do FreeBSD

Resumo

Estas são as Perguntas Mais Freqüentes (FAQ) para as versões 2.X, 3.X e 4.X do FreeBSD. Deve-se assumir que todos os assuntos aqui tratados são relevantes para FreeBSD 2.0.5 ou posterior, a não ser que o contrário esteja explicitamente denotado. Todos os assuntos assinalados com <XXX> estão em processo de desenvolvimento. Se você estiver interessado em ajudar este projeto, envie e-mail para lista de discussão do projeto de documentação do FreeBSD. A versão mais atualizada deste documento está sempre disponível no servidor WWW do FreeBSD. Também pode ser obtida como um único grande arquivo HTML via HTTP; ou, como texto puro, ou nos formatos postscript, PDF, etc. no servidor FTP do FreeBSD. Você também pode querer realizar uma busca nas Perguntas Mais Freqüentes (FAQ).

Versão Traduzida para Português do Brasil

Esta é uma tradução não-oficial do Aviso Legal Padrão do Projeto de Documentação do FreeBSD para o Português do Brasil. Ela não foi publicada pelo Projeto de Documentação do FreeBSD, e legalmente não representa os termos de distribuição de documentação que utiliza o Aviso Legal Padrão do Projeto de Documentação do FreeBSD -- somente o texto original do Aviso Legal Padrão do Projeto de Documentação, em inglês, faz isso. Logo, a versão traduzida não deve ser utilizada como um aviso legal válido. Esperamos, contudo, que esta tradução ajude aos falantes de Português do Brasil a entender melhor o Aviso Legal Padrão do Projeto de Documentação do FreeBSD.

SEMPRE verifique a versão em Inglês mais recente do Aviso Legal Padrão do Projeto de Documentação do FreeBSD na versão em Inglês do Manual do FreeBSD.

Redistribuição e utilização do código fonte (SGML DocBook) ou formato "compilado" (SGML, HTML, PDF, PostScript, RTF e assim por diante) com ou sem modificação, são permitidas contanto que as seguintes condições sejam cumpridas:

1. As redistribuições do código fonte (SGML DocBook) devem reter o aviso de copyright acima, esta lista de condições e a seguinte nota de responsabilidade assim como as primeiras linhas deste arquivo não modificadas.

2. As redistribuições em forma compilada (transformada para outros DTDs, convertida para PDF, PostScript, RTF e outros formatos) devem reproduzir o aviso de copyright acima, esta lista de condições e a seguinte nota de responsabilidade na documentação e/ou outros materiais fornecidos com a distribuição.

Importante

ESTA DOCUMENTAÇÃO É FORNECIDA PELO PROJETO DE DOCUMENTAÇÃO DO FREEBSD "NO ESTADO" E QUAISQUER GARANTIAS EXPLÍCITAS OU IMPLÍCITAS, INCLUINDO, MAS NÃO LIMITADAS AS GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E A DE ADEQUAÇÃO PARA UMA FINALIDADE PARTICULAR SÃO DESMENTIDAS. EM NENHUM EVENTO O PROJETO DE DOCUMENTAÇÃO DO FREEBSD PODERÁ SER RESPONSABILIZADO POR QUAISQUER DANOS DIRETOS, INDIRETOS, INCIDENTAIS, ESPECIAIS, EXEMPLARES, OU CONSEQÜENTES (INCLUINDO, MAS NÃO LIMITADO A, OBTENÇÃO DE BENS OU SERVIÇOS SUBSTITUTOS; PERDA DE USO, DE DADOS, OU DE LUCROS; OU A INTERRUPÇÃO DE NEGÓCIOS) DE QUALQUER FORMA CAUSADO E EM QUALQUER TEORIA DE RESPONSABILIDADE, SE EM CONTRATO, EM RESPONSABILIDADE ESTRITA, OU PROCESSUAL (PASSÍVEL DE PROCESSO, INCLUINDO NEGLIGÊNCIA OU NÃO) LEVANTADA DE QUALQUER FORMA PELO USO DESTA DOCUMENTAÇÃO, MESMO QUE AVISADO DA POSSIBILIDADE DE TAIS DANOS.

English Version

This is an unofficial translation of the Standard FreeBSD Documentation Project Legal Notice into Brazilian Portuguese. It was not published by The FreeBSD Documentation Project, and does not legally state the distribution terms for documentation that uses the Standard FreeBSD Documentation Project Legal Notice -- only the original English text of the Standard FreeBSD Documentation Project Legal Notice does that. Therefore, the translated version should not be used as a valid legal notice. However, we hope that this translation will help Brazilian Portuguese speakers understand the Standard FreeBSD Documentation Project Legal Notice better.

Make sure to ALWAYS check the latest English version of the Standard FreeBSD Documentation Project Legal Notice in the English documentation version of the FreeBSD Handbook.

Redistribution and use in source (SGML DocBook) and "compiled" forms (SGML, HTML, PDF, PostScript, RTF and so forth) with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code (SGML DocBook) must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.

2. Redistributions in compiled form (transformed to other DTDs, converted to PDF, PostScript, RTF and other formats) must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Importante

THIS DOCUMENTATION IS PROVIDED BY THE FREEBSD DOCUMENTATION PROJECT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FREEBSD DOCUMENTATION PROJECT BE LIABLE FOR ANY DIRECT,

INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Índice

1. Introdução 1
2. Documentação e Suporte 9
3. Instalação 15
4. Compatibilidade de Hardware 29
5. Resolução de Problemas 43
6. Aplicações Comerciais 61
7. Aplicações de Usuário 67
8. Configuraçãao do *Kernel* 73
9. Discos, Sistemas de Arquivos e Carregadores de Inicialização (*Boot Loaders*) 77
10. Administração do Sistema 91
11. O sistema X, sistema de interface gráfica e os Consoles Virtuais 107
12. Redes 123
13. Segurança 131
14. PPP 133
15. Comunicações Seriais 149
16. Perguntas Variadas 157
17. As gracinhas do FreeBSD 167
18. Tópicos Avançados 173
19. Reconhecimentos 183

Lista de Tabelas

3.1. Tamanho máximo dos arquivos. 27
12.1. Interfaces de rede baseadas no chipset DEC PCI 126

Lista de Exemplos

11.1. Seção “Pointer” no XF86Config para o mouse com bolinha de scroll, da série 3.3.x do XFree86, usando a tradução se;rie 3.3.x do XFree86, usando a tradução por meio do moused 109
11.2. Seção “InputDevice” do XF86Config para usar a tradução do X Server na série 4.X do XFree86. 109
11.3. Exemplo de “.emacs” para usar paginamento em mouse com bolinha de scroll. 109
11.4. Seção “Pointer” do `XF86Config` com um mouse com scroll na série 3.3.x do XFree86. 110
11.5. Seção “InputDevice” do `XF86Config` com um mouse com scroll na série 4.x do XFree86. 110
11.6. Arquivo “.emacs” para usar paginamento em mouse com bolinha de scroll. 111
11.7. Configuração do Emacs para Imwheel 111
11.8. Configuração do Xemacs para Imwheel 112

Capítulo 1. Introdução

Bem vindo às Perguntas Mais Freqüentes (FAQ) para as versões 2.X à 4.X do FreeBSD!

Como é de costume em Perguntas Mais Freqüentes (FAQ) da Usenet, este documento pretende cobrir as perguntas mais freqüentes relacionadas ao sistema operacional FreeBSD (e claro, respondê-las todas!). Embora originalmente tais documentos tivessem apenas a finalidade de reduzir a utilização da largura de banda da rede ao evitar que o mesmo tipo de pergunta antiga fosse sempre repetida, FAQs tornaram-se reconhecidamente uma fonte valiosa de informações.

Inúmeros esforços foram feitos para tornar este FAQ o mais informativo possível; se você tiver alguma sugestão de como esse documento pode ser melhorado, sinta-se a vontade para enviar suas sugestões por e-mail para o Responsável pelo FAQ <faq@FreeBSD.org >.

P: O que é FreeBSD?

R: Em síntese, FreeBSD é um sistema operacional *UN*X-like* para plataformas i386 e Alpha/AXP, baseado no "4.4BSD-Lite" da Universidade da Califórnia em Berkeley, com alguns aprimoramentos adotados do "4.4BSD-Lite2". O FreeBSD também é baseado, indiretamente, na conversão de William Jolitz conhecida como "386BSD" para a plataforma i386 do "Net/2" da Universidade da Califórnia, em Berkeley; apesar que pouquíssimo código originado do 386BSD ainda exista no FreeBSD. Uma descrição mais abrangente do que é FreeBSD e como o sistema funciona, pode ser encontrada na página principal do FreeBSD.

O FreeBSD é amplamente utilizado por empresas, Provedores de Serviço Internet, pesquisadores, profissionais de informática, estudantes e usuários domésticos no mundo todo, para trabalho, educação e recreação. Alguns destes exemplos podem ser observados na Galeria FreeBSD,.

Para informações mais detalhadas sobre o FreeBSD, por favor, leia o Manual do FreeBSD.

P: Qual o objetivo do Projeto FreeBSD?

R: O objetivo do Projeto FreeBSD é oferecer *software* que possa ser utilizado para qualquer finalidade e sem obrigações anexadas à esse código. Muitos de nós investimos significantemente no código (e no projeto como um todo), e com certeza não nos importaríamos em receber algum tipo de compensação financeira neste momento ou qualquer outro no futuro, mas ninguém no projeto insistirá nisso. Acreditamos que a nossa primeira e mais importante missão é oferecer código para toda e qualquer pessoa, que possa ser utilizado para qualquer propósito, de forma que esse código ofereça o maior número possível de benefícios e formas de uso. Nós acreditamos que este é um dos objetivos fundamentais do *Software* Livre, e é um dos quais nós apoiamos com entusiasmo.

O código fonte em nossa árvore que é distribuído sob a Licença Pública Geral GNU (GPL) ou sob a Licença Pública Geral de Bibliotecas GNU (LGPL) inclue, pode-se dizer, algumas obrigações anexadas a ele; contudo tais restrições visam garantir o acesso livre a esse código, e não o contrário. Devido à complexidades adicionais que envolvem a utilização comercial de software licenciado sob GPL, nós procuramos substituir tais softwares sob a mais relaxada licença de direito autoral FreeBSD sempre que possível;.

P: A licença FreeBSD tem alguma restrição?

R: Sim. Entrentanto, essas restrições não definem regras a respeito de como o código deve ser utilizado, mas de como você deve tratar o Projeto FreeBSD ao utilizar código distribuído pelo mesmo. Se você tem sérias dúvidas sobre o licençiamento, sinta-se a vontade para ler a licença. Para os meramente curiosos, a licença pode ser resumida em:

- Não alegue que o código foi escrito por você.
- Não nos processe se o código falhar.

P: O FreeBSD pode substituir meu sistema operacional atual?

R: Para maioria das pessoas, sim. Mas essa não é uma pergunta tão simples assim.

A maioria das pessoas, na verdade, não utiliza um sistema operacional. As aplicações utilizadas pelos usuários é que realmente usam o sistema operacional. O FreeBSD é projetado de forma a oferecer um ambiente robusto e completo para as aplicações. Suporta uma enorme variedade de navegadores internet, de suítes de escritório, clientes de e-mail, programas de manipulação gráfica, ambientes de programação, servidores e serviços de rede, e praticamente tudo mais que você pode desejar. A maioria destas aplicações podem ainda ser gerenciadas através da Coleção de `Ports`.

Em circunstâncias nas quais precise usar uma aplicação disponível apenas para um determinado sistema operacional, não é possível substituir aquele sistema operacional. Entretanto, há uma boa chance que alguma aplicação similar à que você precisa, exista para FreeBSD. Se você quer ter, desde um sólido conjunto de aplicações para escritório, até um robusto e altamente escalável servidor Internet, ou simplesmente uma estação de trabalho confiável, onde você possa realizar seu trabalho sem interrupções, FreeBSD provavelmente vai suprir todas as suas necessidades. Inúmeras pessoas pelo mundo todo, desde usuários novatos à administradores de sistemas UNIX experientes usam FreeBSD como seu único sistema operacional para *desktop*.

Se você está migrando para FreeBSD a partir de algum outro ambiente UNIX, provavelmente já sabe quase tudo o que precisa pra começar a se envolver com o sistema. Entretanto, se o seu histórico em computação envolveu somente sistemas operacionais baseados em ambientes gráficos como Windows e antigos Mac OS, será necessário investir algum tempo a mais aprendendo a maneira UNIX de fazer

as coisas. Este FAQ e o Manual do FreeBSD são excelentes formas de começar sua jornada.

P: Por que o sistema se chama FreeBSD?

R:
- Pode ser utilizado sem nenhum encargo monetário, inclusive para uso comercial.
- O código fonte completo do sistema operacional é livremente distribuído, e pode ser adquirido gratuitamente. O menor número possível de restrições foram colocadas sobre o uso do sistema, sua distribuição e sua incorporação à outro projeto (comercial ou não).
- Qualquer pessoa que tiver feito alguma correção ou aprimoramento do código do sistema pode livremente enviar suas alterações e ter seu código adicionado à árvore de código fonte do sistema (obviamente sujeito a prévias análises).

É importante ressaltar que a palavra de origem inglesa “free” em português pode ser traduzida como “livre” e “gratuito”. Além disso, a palavra “free” está sendo usada aqui com dois significados: “sem custo” e “você pode fazer o que quiser”. “Free” no nome do sistema operacional remete aos dois significados da palavra. O sistema pode ser utilizado “sem nenhum custo”, e pode ser utilizado “da forma que você quiser”. Exceto por algumas poucas coisinhas que você não pode fazer com o FreeBSD (por exemplo, fingir que foi você quem o escreveu), você pode realmente fazer o que bem entender com o sistema.

P: Qual a versão mais recente do FreeBSD?

R: A versão 9.2 é a versão *RELEASE* mais recente; lançada em September 2013. Esta também é a versão *STABLE* mais recente.

Resumidamente, *-STABLE* é a série voltada para Provedores de Serviço de Internet, usuários corporativos, ou qualquer usuário que deseje estabilidade e um número mínimo de alterações e novas características adotadas do *snapshot -CURRENT*. Lançamentos podem vir de qualquer um dos ramos de desenvolvimento; a série *-CURRENT*, todavia, deveria ser utilizada apenas por usuários preparados para um ambiente em constante modificação, instável em muitas de suas características e extremamente sem garantias (ao menos, quando comparado ao *-STABLE*).

Lançamentos são realizados de alguns em alguns meses. Muitos usuários mantém o código fonte de seus sistemas em mais sincronia com a árvore de desenvolvimento do FreeBSD (veja as perguntas sobre FreeBSD-CURRENT e FreeBSD-STABLE) que isto, fazer isto é uma demonstração de interesse e compromisso visto que o código fonte sofre constantes modificações.

P: O que é FreeBSD-CURRENT?

R: FreeBSD-CURRENT é a versão de desenvolvimento do sistema operacional, que brevemente se tornará a série 5.0-RELEASE. Exatamente por ser uma série de desenvolvimento, e portanto sem garantias de estabilidade, o uso desse sistema

operacional é de interesse exclusivo de desenvolvedores que trabalham no sistema, usuários extremamente experientes que acompanham e analisam (testam) o novo sistema ou daqueles que o fazem por *hobby*. Veja a seção relevante no Manual do FreeBSD.

Se você não tem familiaridade com o sistema operacional, não é um usuário experiente ou não consegue distinguir a diferença entre um problema de verdade e um problema temporário, então é desaconselhável que você use o FreeBSD-CURRENT. Essa série, as vezes, evolui de forma extremamente rápida, e pode se tornar extremamente instável e subutilizável por vários dias seguidos. Usuários do FreeBSD-CURRENT devem ser capazes de analisar qualquer problema no sistema, e apenas relatar a falha se o problema tratar-se de um erro ou um engano no desenvolvimento do mesmo ao invés de "pequenos problemas temporários de instabilidade (*glitches*)". Perguntas sobre o porquê de "make world produzir erros a respeito de grupos" são devidamente ignoradas ou escrachadas na lista de discussão da série -CURRENT.

Diariamente, *snapshots* são lançados baseados no estado atual de desenvolvimento dos ramos -CURRENT e -STABLE. Atualmente, distribuições ocasionais de *snapshots* estão sendo disponibilizadas. Os objetivos por trás do lançamento de cada *snapshot* são:

- Testar a versão mais recente do programa de instalação.
- Dar a oportunidade para aqueles que querem usar o -CURRENT ou o -STABLE - mas não tem tempo ou não tem uma conexão Internet rápida o suficiente para estarem diariamente sincronizados com a versão mais atualizada do código no projeto.
- Manter um ponto de referência fixo, em relação ao código em desenvolvimento e o código disponível até então, para o caso de nós seriamente "quebrarmos" alguma coisa. (Embora CVS normalmente previna que desastres horríveis como este aconteçam :)
- Garantir que todas as novas características e funções do sistema que precisem ser testadas, tenham o maior número possível de pessoas potencialmente testando-as.

Sob nenhuma circunstância, nenhum *snapshot* -CURRENT pode ser considerado software de "qualidade de produção" para qualquer que seja o propósito, e por mais maduro que o código -CURRENT atual possa parecer. Se a intenção é usar um sistema estável e completamente testado, você deverá usar apenas lançamentos, ou *snapshots* do ramo -STABLE.

Os *snapshots* lançados podem ser diretamente acessados em ftp://current.FreeBSD.org/pub/FreeBSD/ para a série 5.0-CURRENT e em releng4.FreeBSD.org para *snapshots* da série 4-STABLE. *Snapshots* para a série 3-

STABLE não estão sendo produzidos na data em que este documento foi escrito (Maio de 2000).

Normalmente, os snapshots são gerados uma vez ao dia, para todas as séries em desenvolvimento ativo.

P: Qual o conceito por trás do FreeBSD-STABLE?

R: Nos primórdios do projeto quando o FreeBSD 2.0.5 foi lançado, a árvore de desenvolvimento do sistema foi dividida em dois ramos. Um ramo foi chamado -STABLE e o outro -CURRENT. O FreeBSD-STABLE é direcionado para Provedores de Serviços de Internet e para outros empreendimentos comerciais que não pretendem conviver com mudanças bruscas ou testar novas características experimentais do sistema. Ele recebe apenas código que tenha sido totalmente testado, correções de problemas e outras pequenas inovações incrementais. O FreeBSD-CURRENT, por outro lado, tem sido uma linha sem interrupções visando ao 5.0-RELEASE (e além) desde o lançamento 2.0. Se uma pequena ilustração em arte ASCII ajudasse, isto seria o que pareceria:

```
                2.0
                 -|
                 -|
                 -|  [2.1-STABLE]
 *BRANCH*       2.0.5 --> 2.1 --> 2.1.5 --> 2.1.6 --> 2.1.7.1  ↺
[2.1-STABLE termina]
                 -|                              (Mar 1997)
                 -|
                 -|
                 -|  [2.2-STABLE]
 *BRANCH*       2.2.1 --> 2.2.2-RELEASE --> 2.2.5 --> 2.2.6 --> ↺
2.2.7 --> 2.2.8 [fim]
                 -|       (Mar 1997)    (Out 97) (Abr 98) (Jul ↺
98) (Dez 98)
                 -|
                 -|
              3.0-SNAPs  (inicio Q1 1997)
                 -|
                 -|
              3.0-RELEASE (Out 1998)
                 -|
                 -|  [3.0-STABLE]
 *BRANCH*      3.1-RELEASE  (Fev 1999) --> 3.2 --> 3.3 --> 3.4 --
> 3.5 --> 3.5.1
                 -|                      (Mai 1999) (Set 1999) ↺
(Dez 1999) (Jun 2000) (Jul 2000)
                 -|
                 -|  [4.0-STABLE]
 *BRANCH*        4.0  (Mar 2000) --> 4.1 --> 4.1.1 --> 4.2 --> ↺
4.3 --> 4.4 --> -... Lançamentos 4.x futuro -...
                 -|
                 -|              (Jul 2000)   (Set 2000)   ↺
(Nov 2000)
```

```
                    \|/
                     +
            [5.0-CURRENT continua]
```

O ramo 2.2-STABLE saiu de produção com o lançamento 2.2.8. O ramo 3-STABLE saiu de produção com o lançamento 3.5.1, que foi também o último -RELEASE 3.X. As únicas modificações ainda realizadas em quaisquer destes ramos são praticamente relacionados apenas à correções de segurança.

O 4-STABLE é o ramo -STABLE em desenvolvimento ativo. A versão mais recente da série 4-STABLE é 9.2-RELEASE, lançada em September 2013.

O ramo 5-CURRENT está lentamente progredindo para o que se tornará o FreeBSD 5.0-RELEASE e além. Veja O que é FreeBSD-CURRENT? para obter mais informações sobre este ramo.

P: Quando são realizados lançamentos FreeBSD?

R: O Time de engenharia de Lançamento <re@FreeBSD.org > lança uma nova versão do FreeBSD, em média, a cada 4 meses. As datas de lançamento são anunciadas com uma certa antecedência, de forma que os desenvolvedores trabalhando no sistema saibam quando seus projetos precisam estar terminados e testados. Um período de testes antecede cada novo lançamento, de forma a garantir que a adição de novas características não comprometa a estabilidade do lançamento. Muitos usuários consideram tais precauções uma das principais vantagens do projeto FreeBSD, mesmo admitindo que, as vezes, esperar que as novidades sejam adotadas pelo ramo -STABLE possa ser um pouco frustante.

Mais informações sobre o processo de engenharia de lançamento (incluindo a programação de novos lançamentos) podem ser obtidas nas páginas de engenharia de lançamento no sítio WWW do FreeBSD.

Para as pessoas que precisam, ou desejam um pouco mais de emoção, *snapshots* binários são feitos diariamente como discutido acima.

P: Quem é responsável pelo FreeBSD?

R: As principais decisões relacionadas ao Projeto FreeBSD, como os objetivos e direção geral do projeto, e quem tem permissão para adicionar código à árvore de código, são tomadas por um grupo central (`core team`) composto de 9 pessoas. Existe um grupo muito maior, composto de mais de 200 desenvolvedores, denominados *committers*, que tem autorização para fazer alterações diretamente na árvore de código do FreeBSD.

Entretanto, a maioria das alterações não triviais são previamente discutidas nas listas de discussões, e não existe restrição quanto a quem pode participar das discussões.

P: Onde posso obter o FreeBSD?

R: Todo lançamento significativo do FreeBSD está disponível via FTP anônimo no sítio FTP do Projeto FreeBSD:

- Para obter o lançamento 3.X-STABLE corrente, 3.5.1-RELEASE, veja diretório 3.5.1-RELEASE.
- O lançamento 4-STABLE corrente, 9.2-RELEASE pode ser encontrado no diretório 9.2-RELEASE.
- *Snapshots* 4.X são normalmente criados uma vez ao dia.
- Lançamentos *Snapshot* 5.0 são feitos uma vez ao dia no ramo -CURRENT, útil apenas tanto para aqueles que gostam de viver no limite quanto para aqueles que precisam usar a versão mais recente possível com todas as últimas características; sejam pessoas conduzindo testes ou desenvolvedores.

Informação sobre como obter o FreeBSD em CD, DVD, e outras mídias, pode ser encontrada no Manual do FreeBSD.

P: Como eu acesso o banco de dados de Relatórios de Problemas?

R: A base de dados de Relatórios de Problemas é um banco de pedidos de alterações realizados pelos usuários. Todos os pedidos de alteração já realizados podem ser consultados (ou novos submetidos) através de nossas interfaces PR WWW para submeter (novos pedidos) e consultar (já submetidos). O comando send-pr(1) também pode ser usado para submeter relatórios de problema e pedidos de alteração por meio de correio eletrônico.

Antes de enviar um relatório de problema, por favor, leia o artigo Escrevendo Relatórios de Problemas para o FreeBSD, que dá boas dicas de como escrever um bom relatório de problema.

P: Como eu me torno um espelho do sítio WWW FreeBSD?

R: Existam várias formas de espelhar o sítio WWW do FreeBSD.

- Você pode obter os arquivos já formatados a partir de um servidor CVSup FreeBSD usando o aplicativo net/cvsup. O arquivo `/usr/share/examples/cvsup/www-supfile` oferece um exemplo de configuração do CVSup para espelhar o servidor WWW do projeto FreeBSD.
- Você pode obter o código fonte do sítio WWW do projeto FreeBSD a partir de qualquer servidor FTP do projeto usando sua ferramento de espelhamento ftp favorita. Considere que estes fontes devem ser processados para publicá-los em formato WWW tradicional. Você pode começar a espelhar o projeto a partir de ftp://ftp.FreeBSD.org/pub/FreeBSD/FreeBSD-current/www.

P: Que outras fontes de informações sobre o FreeBSD existem?

R: Por gentileza, queira verificar a lista de Documentação no sítio WWW principal FreeBSD.

Capítulo 2. Documentação e Suporte

P: Quais bons livros existem sobre FreeBSD?

R: O projeto FreeBSD produz um grande número de documentos disponíveis em http://www.FreeBSD.org/docs.html. Além destes, outros livros recomendados são referenciados nas Bibliografias disponíveis ao final deste `FAQ` e do Manual do FreeBSD.

P: A documentação está disponível em outros formatos, como texto puro (ASCII) ou Postscript?

R: Claro. A documentação pode ser obtida em uma variedade de formatos e opções de compressão no servidor FTP do FreeBSD, sob o diretório /pub/FreeBSD/doc/.

A documentação é organizada em diversas categorias, como:

- O nome do documento, como `faq` ou `Manual do FreeBSD` .
- A codificação e língua do conteúdo do documento. Tal categorização é baseada nos nomes de localização, que podem ser encontrados sob `/usr/share/locale` no seu FreeBSD. Atualmente existem documentos nas seguintes línguas e codificações:

Nome	Significado
`en_US.ISO8859-1`	Inglês Americano
`de_DE.ISO8859-1`	Alemão
`es_ES.ISO8859-1`	Espanhol
`fr_FR.ISO8859-1`	Francês
`ja_JP.eucJP`	Japonês (codificação EUC)
`ru_RU.KOI8-R`	Russo (codificação KOI8-R)
`zh_TW.Big5`	Chinês (codificação Big5)

Nota

Alguns documentos podem não estar disponíveis em todas as línguas.

- Formato da documentação. A documentação é produzida em vários formatos. Cada qual com suas vantagens e desvantagens. Alguns formatos são mais apropriados para leitura on-line, enquanto outros são mais agradéveis estéticamente em formato impresso. Disponibilizar a documentação em todos estes formatos, garante que os leitores poderão sempre ler os trechos de seu interesse, tanto no monitor do seu computador quanto em papel impresso. Atualmente os formatos disponíveis são:

Formato	Significado
`html-split`	Uma série de pequenos documentos HTML, devidamente ligados.
`html`	Um único grande arquivo HTML contendo todo o documento.
`pdb`	Formato de banco de dados pra Palm Pilot, para ser usado com o visualizador iSilo.
`pdf`	PDF (Formato de Documento Portável) da Adobe
`ps`	Postscript
`rtf`	RTF (Formato de Texto Enriquecido) da Microsoft[a]
`txt`	Texto puro

[a] A númeração de página não é automaticamente atualizada quando este tipo de arquivo é aberto no Word. Digite CTRL+A, CTRL+END, F9 depois de abrir o documento no Word, para atualizar a numeração das páginas.

- As técnicas de compressão e empacotamento dos arquivos. Atualmente, 3 destes formatos estão em uso:

  1. Para o formato `html-split` , os arquivos são todos empacotados com tar(1). O resultado é um arquivo `.tar` que é posteriormente comprimido usando as técnicas de compressão detalhadas a seguir.

  2. Todos os outros formatos geram apenas um arquivo, nomeado `book.formato` (por exemplo, `book.pdb` , `book.html` , e outros).

     Estes arquivos são, então comprimidos utilizando três técnicas de compressão:

Tipo	Descrição
`zip`	Formato Zip. Se você quiser descomprimir este formato no

Tipo	Descrição
	FreeBSD, será necessário instalar o `port` archivers/unzip antes.
`gz`	Formato GNU Zip. Para descomprimir estes arquivos, use o comando gunzip(1) que faz parte do FreeBSD.
`bz2`	Formato BZip2. Esse formato é menos popular que os outros, mas geralmente produz arquivos menores. Instale o `port` archivers/bzip2 para descomprimir arquivos deste formato.

Portanto, o Manual do FreeBSD em formato Postscript comprimido com o BZip2 será armazenado como `book.ps.bz2` sob o diretório `handbook/` .

3. A documentação formatada está disponível ainda como um pacote FreeBSD.

Após escolher o formato e o mecanismo de compressão, você deve decidir se vai ou não pegar o documento em formato de *pacote* FreeBSD.

A vantagem de baixar e instalar os pacotes é que a documentação pode então ser gerenciada usando os comandos de gerenciamento de pacotes FreeBSD, como pkg_add(1) e pkg_delete(1).

Se decidir baixar e instalar o pacote, então você deve conhecer o nome do arquivo antes de começar. Os arquivos de documentação-como-pacotes estão estocados em um diretório chamado `packages` . Cada arquivo de pacote segue o padrão de nome `document-name.lang.encoding.format.tgz` .

Por exemplo, o `FAQ`, em língua Inglesa e formato PDF, estará no pacote de nome `faq.en_US.ISO8859-1.pdf.tgz` .

Sabendo isto, você pode usar o seguinte comando pra instalar o pacote contendo o `FAQ` na língua Inglesa e formato PDF:

```
# pkg_add
      ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/packages/
faq.en_US.ISO8859-1.pdf.tgz
```

Depois disso, você pode usar o pkg_info(1) pra determinar onde o documento foi instalado.

```
# pkg_info --f faq.en_US.ISO8859-1.pdf
Information for faq.en_US.ISO8859-1.pdf:

Packing list:
```

```
        Package name: faq.en_US.ISO8859-1.pdf
        CWD to -/usr/share/doc/en_US.ISO8859-1/books/faq
File: book.pdf
        CWD to -.
File: +COMMENT (ignored)
File: +DESC (ignored)
```

Como pode ver, `book.pdf` terá sido instalado sob `/usr/share/doc/en_US.ISO8859-1/books/faq` .

Se você preferir não usar pacotes, será necessário baixar os arquivos comprimidos, depois descomprimí-los e copiar os documentos apropriados para os lugares corretos.

Por exemplo, a versão do `FAQ` dividido em vários arquivos HTML, comprimido usando gzip(1), pode ser encontrado no arquivo `doc/en_US.ISO8859-1/books/faq/book.html-split.tar.gz` . Para baixar e descomprimir aquele arquivo, você deveria fazer o seguinte.

```
# fetch ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/en_US.ISO8859-1/
books/faq/book.html-split.tar.gz
# gzip --d book.html-split.tar.gz
# tar xvf book.html-split.tar
```

Será criada, então, uma série de arquivos `.html`. O principal arquivo é chamado `index.html` contendo o índice, material introdutório e links para outras partes do documento. Posteriormente, você pode copiar ou mover esses arquivos pra onde você quiser.

P: Onde posso obter informações sobre as listas de discussão FreeBSD?

R: Você pode encontrar uma vasta gama de informações na seção do Manual do FreeBSD sobre listas de discussão.

P: Que grupos de notícias (*news groups*) existem sobre o FreeBSD?

R: Informações completas podem ser encontradas na seção do Manual do FreeBSD sobre grupos de notícias (*newsgroups*).

P: Existem canais de bate-papo retransmitido via Internet (IRC) FreeBSD?

R: Sim, a maioria das grandes redes de bate-papo retransmitido via Internet (IRC) tem um canal de bate-papo sobre FreeBSD:

- O canal `#FreeBSD` na EFNet é essencialmente um fórum sobre FreeBSD, mas não entre no canal se você procura suporte técnico, nem se você está procurando uma maneira de evitar a leitura de páginas de manual ou fazer suas próprias pesquisas. Este é essencialmente um canal de bate-papo geral. Assuntos como sexo, esportes ou armas nucleares são tão comuns quanto FreeBSD no canal. Lembre-se, Você Foi Avisado! Para conectar-se, use o servidor `irc.chat.org` .

- O canal `#FreeBSDhelp` na EFNet é dedicado a suporte e auxilio de usuários de FreeBSD. Os participantes neste canal são bem mais receptivos a perguntas que os do canal `#FreeBSD`.

- O canal `#FreeBSD` na DALNET pode ser acessado em `irc.dal.net` nos Estados Unidos, e `irc.eu.dal.net` na Europa.

- O canal `#FreeBSD` na UNDERNET pode ser acessado em `us.undernet.org` nos Estados Unidos, e `eu.undernet.org` na Europa. Partindo do princípio que esse é um canal de ajuda, esteja preparado para ler todos os documentos a que for referido.

- O canal `#FreeBSD` na HybNet *é* um canal de ajuda. Uma lista de servidores pode ser encontrada no sítio WWW da HybNet.

Cada um destes canais é independente, e exatamente por estarem em redes distintas, não é conectada ou relacionada entre si. Os estilos de bate-papo em cada um dos canais são bastante distintos, pode ser necessário testar cada um para descobrir qual á mais adequado ao seu estilo pessoal de bate-papo. Como em toda rede de bate-papo retransmitido via Internet (IRC), nem considere acessar os canais se você ofende-se facilmente ou se você não se dá bem com muita gente jovem (e alguns bem velhos) que usam as mais irregulares formas de escrita e conversação possível, quase sempre assassinando sem o menor pudor todos os princípios verbais - de qualquer língua que seja.

P: Onde eu consigo treinamento em e suporte comercial ao FreeBSD?

R: DaemonNews oferece treinamento em e suporte comercial ao FreeBSD. Mais informações podem ser obtidas no sítio WWW BSD Mall.

FreeBSD Services oferece suporte comercial ao FreeBSD no Reino Unido (além de vender o FreeBSD em mídia DVD). Veja o sítio WWW deles para maiores informações.

A FreeBSD Mall oferece suporte comercial ao FreeBSD. Maiores informações podem ser obtidas no sítio WWW deles.

Qualquer outra organização que ofereça treinamento em ou suporte ao FreeBSD deve entrar em contato com o projeto para serem listadas aqui.

Capítulo 3. Instalação

Nik Clayton

<*nik@FreeBSD.org* >

P: De qual arquivo eu devo baixar o FreeBSD da rede?

R: Até a versão 3.1 era necessário apenas um disquete para instalação do FreeBSD, o disco era o `floppies/boot.flp` . Contudo, depois que o 3.1 foi lançado, o Projeto adicionou ao *kernel* genérico o suporte a uma grande variedade de dispositivos, de forma que ele passou a consumir mais espaço. Por este motivo, desde a série 3.X são necessários dois disquetes, o `floppies/kernel.flp` e o `floppies/mfsroot.flp` . Essas imagens precisam ser copiadas para disquetes, usando ferramentas como o `fdimage` ou o dd(1).

Caso seja necessário baixar da rede a distribuição do sistema (por exemplo, para uma instalação por meio de um sistema de arquivos DOS), você terá que obter as seguintes estruturas da distribuição padrão:

- bin/
- manpages/
- compat*/
- doc/
- src/ssys.*

Para obter instruções completas sobre o procedimento de instalação do FreeBSD e maiores detalhes sobre os meios de instalação, por gentileza, consulte a seção de instalação no Manual do FreeBSD.

P: O que eu devo fazer se as imagens de disquete não couberem em um único disco?

R: Um disquete de 3.5 polegadas (1,44 MB) armazena até 1474560 bytes de dados. O tamanho da imagem de inicialização é de exatamente 1474560 bytes.

Erros comuns, cometidos na preparação dos discos de inicialização, são:

- Baixar a imagem de disco via FTP sem utilizar o modo de transferência *binário.*

  Alguns clientes de FTP - especialmente navegadores de Internet - costumam usar por padrão o modo de transferência *ascii* nas sessões FTP, e para normalizar o arquivo de acordo com o sistema, eles tentam alterar os caracteres finais de cada linha do arquivo. Invariavelmente esse comportamento resulta em baixar uma imagem de inicialização (*boot*) corrompida. Verifique o tamanho da imagem que

você tem em mãos, se é exatamente do mesmo tamanho da imagem no servidor. Caso o tamanho não seja *exatamente* o mesmo, você pode suspeitar do arquivo que você baixou.

Para garantir que esse problema não ocorra, digite *binary* na prompt de comando do seu cliente FTP, ou defina as preferências do programa para utilizar o modo binário. Aí sim, faça baixe da rede a imagem de inicialização.

- Usar o comando `copy` do DOS (ou simplesmente copiar, por meio da Interface Gráfica do sistema) para transferir a imagem de inicialização para o disquete.

  Programas como o `copy` não vão funcionar para copiar a imagem de inicialização direto para o disquete, exatamente porque a imagem foi criada de forma que ela seja carregada diretamente. A imagem tem o conteúdo completo que o disquete deve ter, com seus dados alocados trilha-a-trilha, e portanto não pode ser copiado para o disquete como um simples arquivo. Você tem que copiar a imagem para o disquete usando alguma ferramenta de "cópia crua" (raw copy, como o `fdimage` ou o `rawrite`) como descrito no guia de instalação do FreeBSD.

P: Onde estão as instruções para instalar o FreeBSD?

R: As instruções de instalação do FreeBSD podem ser encontradas na seção de instalação do FreeBSD no Manual do FreeBSD.

P: O que é preciso para usar o FreeBSD?

R: Você vai precisar, no mínimo de um PC 386 com 5MB de memória RAM e no mínimo 60 MB em disco. Essa configuração permite o uso de uma placa de vídeo MDA simples, mas para usar o X11R6 é necessário uma placa de vídeo VGA ou mais avançada.

Para mais informações consulte Capítulo 4, *Compatibilidade de Hardware*

P: Eu tenho apenas 4 MB de memória RAM. Posso usar o FreeBSD?

R: O FreeBSD 2.1.7 foi a última versão do sistema que rodava com apenas 4MB de memória. A partir do FreeBSD 2.2, é necessário no mínimo 5MB de memória para usar o sistema.

Praticamente todas as versões do FreeBSD podem *rodar* com 4MB de memória RAM, contudo, a instalação do sistema operacional não pode ser feita com apenas 4MB. Você pode colocar mais memória para o processo de instalação do sistema, e depois de instalado, voltar a máquina para apenas 4MB de memória, ou como alternativa, instale o seu disco rígido em uma máquina com mais de 4MB, efetue a instalação do sistema, e depois instale o seu disco de volta na máquina com apenas 4MB.

O FreeBSD 2.1.7 não irá instalar em sistemas que usam 640 Kb de memória base + 3 MB de memória extendida. Se sua placa mãe pode fazer o remapeamento da memória "subutilizada" que vai sobrar dos 640kB da região de 1MB, ai sim,

você vai conseguir usar o FreeBSD 2.1.7. Entre no Setup da sua BIOS, procure a opção ``remap'' e habilite-a. Talvez você tenha que desabilitar a opção de ROM shadowing. Com certeza é mais fácil você conseguir mais 4MB apenas para a instalação, compilar um *kernel* customizado e portanto menor, e ai sim, tirar esses 4MB sobresalentes e usar o sistema com apenas os 4MB originais. Também é possível instalar o FreeBSD 2.0.5 e depois "atualizá-lo" para o 2.1.7 com a opção ``upgrade'' disponível no programa de instalação do FreeBSD 2.1.7.

Depois de instalado o sistema, você pode compilar um *kernel* personalizado, que provavelmente irá permitir que o sistema seja usado com 4MB de memória apenas. Existem relatos de sucesso na utilização do sistema com apenas 2MB de memória, contudo, nesse caso é praticamente impossível usar alguma outra aplicação junto ao sistema operacional.

P: Como eu crio meu próprio disquete de instalação?

R: Atualmente não existe uma forma de *simplesmente criar* um disco de instalação personalizado. Para criar um disquete personalizado você terá que preparar todo um novo `release`, o qual, aí sim, teria instruções de instalação.

Para montar um `release` personalizado siga as instruções do artigo de Engenharia de `Release` article.

P: O Windows 95/98 pode co-existir com o FreeBSD?

R: Sim. Primeiro você deve instalar o seu Windows, e depois instalar o FreeBSD. O gerenciador de inicialização (*boot*) do FreeBSD vai ser instalado na MBR do seu disco, e vai conseguir controlar o inicialização entre o FreeBSD e seu Windows. Se você instalar o Windows depois do FreeBSD, a instalação dele irá sobrescrever o setor de inicialização (*boot*) do seu disco, e conseqüentemente seu gerenciador de inicialização (*boot*), sem avisar ou pedir qualquer confirmação. Se esse for o caso, leia a próxima seção.

P: O Windows 95/98 sobrescreveu meu gerenciador de inicialização (*boot*)! Como eu instalo ele de volta?

R: Você pode reinstalar o gerenciador de inicialização (*boot*) do FreeBSD de uma das 3 maneiras:

- Sob o DOS, entre no diretório tools/ da sua distribuição do FreeBSD (seu CDROM por exemplo) e procure o programa `bootinst.exe`. Depois, execute-o da seguinte forma:

```
...\TOOLS> bootinst.exe boot.bin
```

  e o gerenciador de inicialização (*boot*) será reinstalado.

- Faça a inicialização do FreeBSD pelos disquetes de instalação ou pelo CDROM novamente. Entre na opção "Custom" do menu de instalação, escolha a o ítem de

partições (Partition), selecione o drive do disco que continha o seu gerenciador de inicialização (*boot*) (normalmente, se trata do primeiro disco) e então você entra no editor de partições. Não faça nenhuma alteração, apenas aperte a tecla W (Write). O programa de instalação irá pedir a confirmação, se você quer gravar suas informações mesmo sem ter feito nenhuma alteração. Escolha Sim. O programa irá perguntar se você deseja instalar o gerenciador de inicialização (*boot*) do FreeBSD ou se você deseja deixar o setor de inicialização (*boot*) intacto (ou instalar um setor de inicialização (*boot*) padrão) exatamente como no instante da primeira instalação do FreeBSD. Escolha "Boot Manager". Agora o gerenciador de inicialização (*boot*) será reinstalado no disco. Saia do programa de instalação e reinicie o processo de inicialização pelo HD normalmente.

- Inicie o FreeBSD com o disquete (ou CD) de inicialização tradicional, escolha a opção "Fixit" no menu do sysinstall. Escolha entre o disquete de correção ou o segundo CDROM (a opção "live" na distribuição padrão do FreeBSD) no menu a seguir, e entre na shell de correção do sistema. Em seguida execute o comando:

```
Fixit# fdisk --B --b -/boot/boot0 bootdevice
```

  substituindo *bootdevice* pela device controladora do seu disco, como por exemplo, `ad0` (para o primeiro disco IDE),`ad4` (para o primeiro disco IDE na controladora secundária), `da0` (para o primeiro disco SCSI), etc.

P: O meu IBM Thinkpad série A, T ou X trava sempre, quando eu tento inicializar (*boot*) o FreeBSD. Como eu resolvo isso?

R: Um bug nas primeiras versões da BIOS da IBM nessas máquinas, erroneamente identifica as partições FreeBSD como partições FAT especiais. Quando a BIOS tenta reconhecer a partição FreeBSD, o sistema trava.

De acordo com a IBM[1], os seguintes modelos/BIOS tem esse problema corrigido:

Modelo	Revisão da BIOS
T20	IYET49WW ou posterior
T21	KZET22WW ou posterior
A20p	IVET62WW ou posterior
A20m	IWET54WW ou posterior
A21p	KYET27WW ou posterior
A21m	KXET24WW ou posterior
A21e	KUET30WW

[1]Em um e-mail enviado por Keith Frechettei <kfrechet@us.ibm.com >.

Existem relatos de que as revisões posteriores das BIOS IBM re-introduziram esse bug. Essa mensagem enviada por Jacques Vidrine para a lista de discussão de temas específicos ao uso do FreeBSD em notebooks e demais equipamentos móveis. descreve uma série de procedimentos que podem funcionar no seu laptop IBM, caso seja uma versão um pouco mais nova, e que não consiga inicializar (*boot*) o FreeBSD corretamente. Você pode ainda fazer uma atualização ou desatualização (upgrade ou downgrade) da BIOS.

Se a BIOS é mais antiga, e você não considera sua atualização, existe uma opção que pode sanar seu problema. A instalação do FreeBSD pode ser feita alterando-se a identificação da partição (partition ID) do sistema, e depois instalar novos setores de inicialização (*boot*) que podem controlar uma partition ID diferente.

O primeiro passo é restaurar o seu laptop ao ponto onde ele pode fazer os auto-testes, ou seja, os testes básicos de I/O da BIOS. Para fazer isso, basta ligar o computador de forma que ele não consiga encontrar a partição primária do FreeBSD. A maneira mais simples de faze-lo, é retirando o disco rígido do seu laptop, e temporariamente ligando-o em um ThinkPad mais antigo (como oThinkPad 600) ou em um PC comum, com um cabo de conversão apropriado. Uma vez feito isso, basta apagar a partição FreeBSD e colocar o disco de volta no laptop. Agora sim, o ThinkPad deve estar de volta ao estado onde ele pode reconhecer o disco.

Com a máquina funcionando, basta seguir as próximas instruções para fazer o seu FreeBSD instalar:

1. Baixe da rede os arquivos `boot1` e `boot2` no site http://people.FreeBSD.org/~bmah/ThinkPad/. Coloque esses arquivos em algum lugar onde você possa acessá-los posteriormente.

2. Instale o FreeBSD normalmente no ThinkPad. *Não* use o modo `Dangerously Dedicated`. *Não* reinicie o sistema quando o processo de instalação for concluído.

3. Vá para a “Shell Holográfica de Emergência” (ALT+F4) ou inicie uma shell de recuperação - “fixit”

4. Use o fdisk(8) para alterar a partition ID de `165` para `166` (166 é o ID usado pelo OpenBSD).

5. Coloque os arquivos `boot1` e `boot2` no sistema de arquivos local.

6. Use o disklabel(8) para escrever o `boot1` e o `boot2` na sua partição FreeBSD.

   ```
   # disklabel --B --b boot1 --s boot2 ad0sn
   ```

   n é o número da partição onde o FreeBSD está instalado.

7. Reinicie o sistema. O gerenciador de inicialização (*boot*) oferecerá a opção de iniciar o `OpenBSD`, mas na verdade essa opção estará iniciando o FreeBSD.

Agora, se você quer manter os sistemas operacionais OpenBSD e FreeBSD no mesmo laptop ThinkPad, pode considerar isso um exercício prático que fica a critério do leitor.

P: Posso instalar o FreeBSD em um disco com bad blocks?

R: Até a versão 3.0, o FreeBSD tinha um utilitário chamado `bad144`, que automaticamente remapeava os bad blocks. Atualmente, os discos IDE modernos são capazes de fazer isso sozinhos, portanto o `bad144` foi retirado da árvore do FreeBSD. Se sua intenção é instalar o FreeBSD 3.0 ou alguma versão mais recente, nós sinceramente aconselhamos que você compre um novo disco. Se você não quer comprar um disco novo, então use o FreeBSD 2.X.

Se você esta tendo problemas de bad block com algum disco IDE moderno, provavelmente o disco será perdido em breve, já que ele está tão corrompido que a controladora interna não está conseguindo corrigir e remapear os bad blocks. Sugerimos que você compre um disco novo logo, e realize cópia de segurança (*backup*) dos dados, enquanto o disco ainda funciona.

Se o drive de disco é SCSI e está apresentando bad blocks, leia essa resposta.

P: Eu acabei de atualizar o sistema da série 3.X para 4.X, e a minha primeira inicialização (*boot*) falhou com a mensagem bad sector table not supported

R: O FreeBSD 3.X e anteriores suportavam o programa `bad144`, que automaticamente remapeava bad blocks. O FreeBSD 4.X e posteriores não suportam mais esse programa, devido ao fato que os controladores de discos IDE atuais conseguem remapear bad blocks automaticamente. Leia essa pergunta para mais informações.

Para corrigir esse problema depois de uma atualização, é necessário mover fisicamente o disco com problemas para um outro sistema FreeBSD funcional e usar o disklabel(8) da forma discutida a seguir.

P: Como eu faço se um disco tem informações criadas pelo `bad144` antes de atualizar o sistema, e depois de atualizado para o FreeBSD 4.0 ou posterior, a inicialização falha?

R: Use o disklabel(8) para identificar esse ambiente. `disklabel -r drive device` vai te mostrar o conteúdo do disco. Procure o campo `flags`. Se encontrar a informação `flags: badsect` é porque esse disco está usando o bad144. Por exemplo, o disco a seguir tem o `bad144` habilitado:

```
# disklabel --r wd0
# -/dev/rwd0c:
type: ESDI
disk: wd0s1
label:
```

```
flags: badsect
bytes/sector: 512
sectors/track: 63
```

P: Como eu removo o `bad144` do meu sistema anterior ao 4.X de forma que eu possa atualizá-lo com segurança?

R: Use o comando `disklabel -e -rwd0` para editar as informações do seu disco. Basta retirar a palavra `badsect` do seu campo flags, salvar a alteração e sair do programa. O bad144 ainda estará ocupando algum espaço no seu disco, mas ele estará funcional para série 4.X e posteriores.

Caso seu disco tenha um número muito alto de bad blocks, é recomendado a troca do disco.

P: Coisas estranhas acontecem quando inicio o sistema com o disco de instalação! O que está acontecendo?

R: Se sua máquina está desligando ou espontâneamente reiniciando sempre que você tenta iniciar o sistema com o disco de instalação, aqui vão algumas perguntas que você deveria fazer a si mesmo:

1. O disco de instalação foi feito a partir de um disquete novo, recém formatado e completamente livre de erros (de preferência algum disco que acabou de sair da caixa, ao contrário desse seu disco que estava perdido há quase 3 anos debaixo da cama)?

2. Você baixou da rede a imagem em modo binário? (não se envergonhe, até o melhor de nós já baixou um arquivo binário em modo ASCII ao menos uma vez na vida!)

3. No Windows 95 ou 98, você usou o `fdimage` ou o `rawrite` em modo DOS? Esses sistemas operacionais as vezes interferem na forma com que os programas escrevem dados diretamente no hardware, exatamente o que o processo de criação da imagem de disco faz, mesmo que você execute um prompt do DOS no ambiente gráfico o problema pode ocorrer.

Ainda existem notícias de arquivos de imagens sendo corrompidos pelo Netscape, durante o *download*, por isso é mais seguro utilizar um cliente de FTP diferente.

P: Eu inicializei o FreeBSD a partir do meu CDROM ATAPI, mas o programa de instalação diz que o CDROM não foi encontrado. Para onde ele foi?

R: A causa desse problema curioso é a configuração errada do seu drive de CDROM. Hoje em dia muitos PCs vem com o CDROM instalado como escravo na segunda controladora IDE, sem nenhum disco ou drive óptico do tipo mestre na mesma controladora. De acordo com as especificações ATAPI esse tipo de configuração é incorreta e ilegal. Alguns sistemas, como o Windows, simplesmente ignoram uma

série de especificações legais na arquitetura de computadores pessoais, e acabam oferecendo suporte a essa configuração errônea - o que mais tarde pode causar outros conflitos. Depois que o sistema inicia, a BIOS passa a ignorar esse drive, e por isso o FreeBSD não consegue encontrá-lo, para completar a instalação.

Reconfigure o seu computador de forma que o CDROM esteja como mestre na sua controladora IDE, ou que exista um outro periférico como mestre na controladora onde o CD estiver como escravo.

P: Posso instalar o FreeBSD no meu laptop via PLIP (IP em Linha Paralela)?

R: Claro. Use o cabo laplink padrão. Caso necessário, verifique a seção de PLIP do Manual do FreeBSD para obter detalhes sobre a instalação do FreeBSD via rede em porta paralela.

Se você está usando o FreeBSD 3.X ou anterior, dê uma olhada na página de Computação Móvel.

P: Qual geometria eu devo utilizar para um disco?

R:

Nota

Por “geometria”, nós entendemos o número de cilindros, cabeças e setores/trilhas de um disco. Por conveniência, vamos nos referir à esses dados como C/H/S (Cylinders/Heads/Sectores). É a partir dessa informação que a BIOS dos PCs definem quais áreas de um disco podem ser usadas para leitura/escrita.

A geometria de disco costuma causar uma série de confusões entre administradores de sistemas menos experientes. Para começar, a geometria *física* de um disco SCSI é totalmente irrelevante, pois o FreeBSD trabalha com blocos de discos. Na verdade, não existe exatamente “a” geometria física de um disco, visto que a densidade de um setor varia de acordo com os discos. Os fabricantes chamam de “geometria física” as especificações que eles definem para que o menor espaço possível em disco seja desperdiçado. Em discos IDE, o FreeBSD trabalha com as informações de C/H/S, mas todos os dispositivos modernos, internamente convertem essa informações em referências a blocos de disco.

O que importa, portanto, é a geometria *lógica*. O valor lógico é a resposta que a BIOS obtém quando pergunto “qual sua geometria?” ao disco. É esse valor, então, que é usado para definir a forma de acesso ao dispositivo de armazenamento. O FreeBSD usa as informações da BIOS quando inicializa (*boot*), e por isso é extremamente importante obter essa informação de maneira correta. No geral, se você tem mais de

um sistema operacional no mesmo disco, eles devem concordar no valor lógico da geometria do disco, caso contrário você terá sérios problemas ao iniciar o sistema.

Em discos SCSI, a geometria à ser utilizada depende do suporte à tradução extendida definido na sua controladora de disco (normalmente esse suporte é chamado de "support for DOS disks >1GB", que identifica o suporte à discos DOS cuja capacidade de armazenamento é maior que 1GB - ou alguma identificação similar.). Se essa opção está desabilitada, então o C/H/S do disco será de ***N*** cilindros, 64 cabeças e 32 setores/trilhas, onde o valor ***N*** equivale a capacidade (em MB) do disco. Por exemplo, um disco de 2GB teria 2048 cilindros, 64 cabeças e 32 setores/trilhas.

Se a opção *estiver* habilitada (normalmente ela é habilitada por padrão, para sanar algumas limitações de sistemas baseados em MSDOS), e a capacidade do disco forma maior que 1GB, os valores C/H/S do disco serão M cilindros, 63 setores por trilha (*não* 64) e 255 cabeças, sendo 'M' a capacidade do disco, em MB, dividido por 7.844238 (!). Então, por exemplo, o mesmo disco de 2GB nessa situação teria 261 cilindros, 63 setores por trilha e 255 cabeças.

Se você não entendeu o porque disso, ou se o seu FreeBSD falha no momento de reconhecer a geometria correta do seu disco durante a instalação, existe uma forma de tentar resolver esse problema. Crie uma pequena partição do tipo DOS no seu disco, e verifique se a BIOS consegue identificar corretamente a geometria do mesmo. Caso consiga, a instalação vai se completar com tranqüilidade, e a pequena partição DOS pode sempre ser deletada, com o editor de partições do FreeBSD.

Como alternativa, existe uma aplicação gratuitamente disponível com a distribuição do FreeBSD, chamada de `pfdisk.exe` . Ela pode ser encontrada sob o diretório `tools` no CDROM do FreeBSD ou nos servidores FTP do projeto. Esse programa serve para descobrir qual a geometria usada por outros sistemas operacionais no disco local. Nesse caso, esse valor pode ser definido no editor de partições do FreeBSD.

P: Existem restrições quanto ao particionamento de discos?

R: Sim, existem. A principal delas, é que a partição "root" não pode ter mais de 1024 cilindros, senão a BIOS não consegue iniciar o *kernel* do sistema a partir dessa partição. (Note que essa é uma limitação das BIOS dos computadores pessoais, e não do FreeBSD).

Em um disco SCSI, essa limitação implica que a partição raiz (root) deve estar alocada nos primeiros 1024MB do disco (ou nos primeiros 4096MB, caso o suporte a tradução extendida esteja habilitada - veja pergunta anterior). Em discos IDE, o valor correspondente equivale a 504MB para partição raiz (root).

P: O FreeBSD suporta programas gerenciadores de discos?

R: O FreeBSD reconhece apenas o "Ontrack Disk Manager". Outros gerenciadores de discos não são suportados.

Se sua intenção é usar o disco com FreeBSD, você não precisa de um gerenciador de discos. Basta configurar o disco para o total de espaço que a BIOS reconhece (normalmente 504MB) e o FreeBSD vai conseguir identificar o tamanho real do disco. Se você estiver usando um disco antigo com uma controladora MFM, será necessário avisar ao FreeBSD quantos cilindros o disco tem.

Caso queira usar o disco com FreeBSD e algum outro sistema operacional, provavelmente também não será necessário um gerenciador de discos. Certifique-se apenas que a partição de inicialização (*boot*) do FreeBSD e a partição do outro sistema operacional estejam nos primeiros 1024 cilindros do disco. Normalmente, para administradores de sistemas que tomam decisões racionais, 20MB de espaço em uma partição de inicialização (*boot*) é mais que o suficiente.

P: Quando eu inicio o FreeBSD, eu obtenho a mensagem Missing Operating System. O que está acontecendo?

R: Esse é um caso tópico do FreeBSD e o DOS ou qualquer outro sistema operacional discordando de suas definições em relação a geometria do disco. Provavelmente você terá que reinstalar o FreeBSD, mas se seguir as instruções citadas nas perguntas anteriores, raramente esse problema vai acontecer.

P: Porque eu não consigo passar da tela `F?` do gerenciador de inicialização (*boot*)?

R: Esse é mais um sintoma do problema descrito na pergunta anterior. A geometria que a sua BIOS reconhece não equivale ao valor definido no FreeBSD! Se a sua controladora de disco ou sua BIOS suportam o modo de tradução de cilindros (normalmente chamado de “>1GB drive support”), tente alterar essa opção e reinstalar o FreeBSD.

P: Eu preciso instalar todos os fontes do sistema operacional?

R: Geralmente não, mas é altamente recomendável que você instale ao menos os fontes `base`, que incluem inúmeros arquivos mencionados ao longo desse documento, como as fontes do sistema, `sys` que inclui as fontes do *kernel* do FreeBSD, sem os quais não se pode criar um *kernel* personalizado. Não existe qualquer dependência do sistema operacional em relação aos seus fontes; com a única exceção do programa config(8), o resto do sistema operacional não precisa dos fontes para funcionar. Os outros fontes do sistema operacional - exceto os fontes do *kernel* - podem ser montados remotamente (via NFS, por exemplo) em qualquer lugar, e ainda assim novos binários podem ser compilados a partir dos mesmos. Devido a restrição única dos fontes do *kernel*, é recomendável que os outros fontes não sejam diretamente montados sob `/usr/src` mas sim, que sejam montados separadamente e depois interligados com links simbólicos apropriados.

Tendo todos os fontes disponíveis, e sabendo reconstruir o sistema a partir dos mesmos, será muito mais fácil manter o FreeBSD sincronizado e atualizado com futuros releases.

Para escolher um subconjunto dos fontes do sistema, escolha a opção `Custom` quando estiver na opção `Distributions` do programa de instalação do sistema.

P: Eu preciso criar um *kernel* personalizado?

R: Construir um novo *kernel* costumava ser uma obrigação na instalação do FreeBSD, mas hoje em dia existe uma interface de configuração do *kernel* muito mais amigável, que permite a redefinição de recursos do sistema. Para acessar essa ferramenta, basta inicializar (*boot*) o sistema com a opção `-c` no prompt de (`boot:`). Em especial, os principais periféricos ISA - normalmente os mais problemáticos - podem ser facilmente configurados com essa opção.

Ainda é recomendável que se construa um *kernel* personalizado, apenas com suporte aos equipamentos e características do sistema que você precisa, de forma a economizar recursos no sistema (especialmente memória RAM), mas essa recompilação não é mais uma obrigação na maioria dos sistemas - mas é sem dúvida um hábito saudável.

P: Eu devo usar criptografia DES, Blowfish, ou MD5 para senhas do sistema? Como eu defino qual delas o usuário deve usar?

R: O formato padrão para senhas no FreeBSD é a criptografia *MD5*. Esse padrão é considerado mais seguro do que os formatos tradicionais de senhas Unix, que normalmente eram baseados no algorítimo *DES*. O FreeBSD ainda pode trabalhar com senhas em formato DES caso você precise compartilhá-las com sistemas que ainda armazenam suas senhas no formato antigo - e menos seguro - dos sistemas Unix originais (para isso você terá que instalar a distribuição "crypto" via sysinstall ou apartir do código fonte). Instalando as bibliotecas crypto será possivel utilizar outros tipos de criptografia, como o formato Blowfish, que é ainda mais seguro do que o MD5. A definição de qual codificação utilizar é definida no campo "passwd_format" do arquivo de configurações de login, o `/etc/login.conf` . Esse campo deve ter o valor "des", "blf" (caso suas bibliotecas estejam disponíveis) ou "md5". Veja a página de manuais do login.conf(5) para maiores informações.

P: Por que o disco de inicialização (*boot*) inicia, mas trava na tela `Probing Devices...` ?

R: Se você tem um drive Zip IDE ou um Jaz conectado ao seu computador, remova-o e tente de novo. A inicialização (*boot*) de instalação do sistema se confunde as vezes quando esses dispositivos estão disponíveis no computador. Depois da instalação os drives são reconhecidos e controlados normalmente. Provavelmente - esperamos - esse problema será corrigido nas próximas versões.

P: Por que ocorre o erro panic: can't mount root, quando eu reinicio o sistema, depois de tê-lo instalado.

R: Esse problema costuma ocorrer por conta de uma pequena confusão entre os blocos do setor de inicialização (*boot*) do disco, e as definições de disco no *kernel*. É um erro

típico apenas de sistemas com dois discos IDE, quando os mesmos estão definidos como disco mestre e escravo, mas em controladoras distintas, e com o FreeBSD instalado na controladora secundária. Os blocos de inicialização (*boot*) acham que o sistema está instalado no segundo disco IDE (o segundo disco reconhecido pela BIOS) enquanto o *kernel* assume o primeiro disco na segunda controladora IDE. Depois do reconhecimento dos equipamentos do sistema o *kernel* tenta montar a partição raiz no disco que o bloco de inicialização (*boot*) acredita ser o disco de inicialização (*boot*), wd1, ao invés do disco correto na segunda controladora, wd2, e por isso o processo de inicialização falha.

Para corrigir esse problema, você tem três opções:

1. No FreeBSD 3.3 e posteriores, reincie o sistema e aperte Enter na tela `Booting kernel in 10 seconds; hit [Enter] to interrupt` . Você será direcionado ao `boot loader` .

   Depois, digite `set root_disk_unit="disk_number"` . *`disk_number`* deverá ser `0` se o FreeBSD estiver instalado como mestre na primeira controladora IDE, `1` se for o escravo na primeira controladora, `2` se for o mestre da segunda controladora IDE, e `3` se for o escravo na segunda controladora.

   Depois digite `boot`, e seu sistema deve ser iniciado corretamente.

   Para tornar essa alteração permanente, (para que você não tenha que digitar isso na mão toda vez que seu FreeBSD tiver que reiniciar) basta colocar a linha `root_disk_unit="disk_number"` no arquivo `/boot/loader.conf.local` .

2. Se você estiver usando o FreeBSD 3.2 ou alguma versão anterior, digite `1:wd(2,a)kernel` na prompt de inicialização do sistema e aperte Enter. Se o sistema iniciar normalmente, execute o comando `echo "1:wd(2,a)kernel" > /boot.config` para tornar essa alteração permanente.

3. Mude o disco com o FreeBSD para primeira controladora IDE.

4. Recompile o *kernel*, altere as linhas de configuração wd para:

```
controller      wdc0    at isa? port -"IO_WD1" bio irq 14 ↺
vector wdintr
disk            wd0     at wdc0 drive 0
# disk            wd1     at wdc0 drive 1 # comment out this ↺
line

controller      wdc1    at isa? port -"IO_WD2" bio irq 15 ↺
vector wdintr
disk            wd1     at wdc1 drive 0 # change from wd2 to ↺
wd1
disk            wd2     at wdc1 drive 1 # change from wd3 to ↺
wd2
```

E instale o novo *kernel*. Se você mudou seu disco e quer voltar ele para configuração original, mude a ordem deles no PC e reinicie o sistema. Seu sistema deve iniciar com sucesso.

P: Quais são as limitações de memória?

R: A limitação de memória é de 4 gigabytes. Essa definição foi testada, veja a configuração do wcarchive para obter mais detalhes. Se você pretende instalar essa quantidade de memória no FreeBSD, seja cuidadoso. Dê preferência para memórias ECC e reduza a capacidade de carga usando modules de memória de 9 chips, ai invés dos módulos de 18 chips.

P: Qual a limitação para o sistema de arquivos FFS?

R: Para o sistema de arquivos FFS, o limite máximo, na teoria é de 8 terabytes (para blocos de 2K), ou 16TB para o tamanho padrão dos blocos, que é de 8K. Na prática os limites variam de 1TB a 4TB de acordo com algumas modificações no sistema de arquivos.

O tamanho máximo para um arquivo no sistema FFS é de 1G de blocos (4TB) caso os blocos sejam de 4K.

Tabela 3.1. Tamanho máximo dos arquivos.

Tamanho do bloco	2.2.7-stable	3.0-current	Funciona com	Deve funcionar
4K	4T-1	4T-1	4T-1	>4T
8K	>32G	8T-1	>32G	32T-1
16K	>128G	16T-1	>128G	32T-1
32K	>512G	32T-1	>512G	64T-1
64K	>2048G	64T-1	>2048G	128T-1

Quando o sistema de arquivos possui blocos de 4K, o triplo de blocos indiretores funcionam, e o limite máximo do sistema de arquivos deveria ser atingido, mas a triplicação dos blocos indiretores (representados aproximadamente pelo resultado de 1K^3 + 1K^2 + 1K) se limita ao valor (errôneo) de 1G-1 no número de blocos do sistema de arquivos. O limite do número de blocos deveria ser 2G-1. Mas por causa de alguns problemas com o número dos blocos no sistema de arquivos, esse valor não pode ser alcançado quando o tamanho dos blocos no sistema de arquivos é 4K.

Em blocos com tamanho de 8K ou maiores, o limite geral é de 2G-1 no número de blocos do sistema de arquivos, exceto no FreeBSD -STABLE onde o triplo indireto do número de blocos pode ser alcançado, de forma que o limite máximo do sistema de arquivos seja representado pela equação ((blocksize/4)^2 + (blocksize/4)), e sob o -CURRENT onde a exceção desse limite pode causar problemas.

P: Por que a mensagem de erro archsw.readin.failed me perturba sempre, depois que eu recompilo e carrego um *kernel* novo?

R: Você pode carregar um novo *kernel* ao especifica-lo diretamente no segundo estágio do processo de inicialização, simplesmente apertando qualquer tecla quando o pipe (|) aparecer, antes que o loader seja carregado. Provavelmente você atualizou todo o sistema operacional, mas recompilou apenas o *kernel*, *sem dar um make world*. Essa ação é arriscada e não é suportada. Faça um Make World!!!!

P: Como eu atualizo meu sistema da série 3.X para 4.X?

R: É *altamente* recomendável que você use *snapshots* binários para fazer isso. *Snapshots* binário do 4-STABLE podem ser encontrados em ftp://releng4.FreeBSD.org/.

Devido às inúmeras alterações da série 3.X para série 4-STABLE, uma atualização direta, a partir dos fontes, corre grande riscos de falhar. A atualização dos fontes pode ser feita, inclusive desde as primórdias versões 2.X até as mais recentes 4-STABLE ou até mesmo 5-CURRENT, mas essa atualização deve ser realizada em vários estágios. Primeiro, atualize a sua série 3.X pra versão mais recente, a 3-STABLE (`RELENG_3`). Depois atualize para o 4.1.1-RELEASE (`RELENG_4_1_1_RELEASE`). Finalmente, tente atualizar para o 4-STABLE (`RELENG_4`).

Se você pretende atualizar seu sistema a partir dos fontes, por gentileza, refira-se ao Manual do FreeBSD para maiores informações.

Cuidado

A atualização direta por meio dos fontes nunca é aconselhável para usuários inexperientes, a atualização da série 3.X para 4.X portanto é menos aconselhável ainda, portanto, caso você não tenha experiências com esse processo de atualização, leia todas as instruções disponíveis no Manual do FreeBSD com cuidado.

P: Onde estão essas “especificações de segurança”?

R: Uma “especificação de segurança” se refere a um conjunto de configurações e de opções no sistema, que tendem a garantir um nível desejável de segurança, por meio de definir ou desabilitar algumas opções e programas no FreeBSD. Para maiores detalhes, veja a seção de Especificação de Segurança no capítulo de pós-instalação do Manual do FreeBSD.

Capítulo 4. Compatibilidade de Hardware

P: O FreeBSD suporta outras arquiteturas além da x86?

R: Sim. Atualmente o FreeBSD tem suporte para arquiteturas Intel x86 e DEC (agora Compaq) Alpha. Também existe um interesse conhecido no `port` FreeBSD para plataforma SPARC. Caso exista interesse em participar desse projeto ou saber mais informações sobre `port` para esta arquitetura, queira juntar-se à lista de discussão do lista de discussão sobre o *port* do FreeBSD para a plataforma SPARC. As plataformas IA-64 e Power-PC foram recentemente adicionadas à lista de arquiteturas que serão futuramente suportadas; entre na lista do lista de discussão sobre o *port* do FreeBSD para a plataforma IA64 e/ou lista de discussão sobre o *port* do FreeBSD para PowerPC para mais informações sobre tais arquiteturas. Para discussões gerais sobre outras arquiteturas, entre na lista de discussão lista de discussão sobre o *port* do FreeBSD para plataformas que não sejam Intel.

Caso seu computador seja de uma arquitetura não suportada pelo FreeBSD e precise de uma solução imediata, nós sugerimos uma olhada no NetBSD ou OpenBSD.

P: Preciso adquirir um novo hardware para um sistema com FreeBSD. Qual o melhor modelo/marca/tipo?

R: Essa é uma discussão tradicional nas listas do FreeBSD. Partindo do princípio que os tipos de equipamentos e suas características alteram-se de forma muita rápida, e que nós tentamos suportar essas mudanças e torná-las suportadas, é *fortemente recomendado* que você sempre leia as Notas de Hardware e faça uma busca nos histórico das listas de discussão antes de perguntar sobre os melhores e mais novos equipamentos disponíveis. É muito provável que as informações que você quer sobre determinado equipamento tenham sido discutidas há menos de uma semana.

Caso você esteja procurando informações sobre *laptops*, verifique o histórico da lista FreeBSD-mobile. Do contrário, o histórico da FreeBSD-questions será o mais indicado, ou de alguma lista específica sobre o tipo de hardware em questão.

P: Que tipos de discos rígido o FreeBSD suporta?

R: O FreeBSD suporta discos EIDE e SCSI (com alguma controladora compatível; veja a próxima pergunta) e todos os outros discos que usam a interface de controle original da "Western Digital" (MFM, RLL, ESDI, e é claro IDE). Algumas controladoras ESDI que usam interfaces de controle proprietária não funcionarão no FreeBSD: mude para controladoras do tipo WD1002/3/6/7 ou algum clone dessa interface.

P: Quais controladoras SCSI são suportadas pelo FreeBSD?

R: Veja a lista completa de equipamentos suportados nas Notas de Hardware atuais.

P: Quais drives de CDROM são suportados pelo FreeBSD?

R: Quaisquer drives SCSI ligados à controladoras suportadas são controladas pelo FreeBSD.

As seguintes interfaces proprietárias de CDROM também são suportadas:

- Mitsumi LU002 (8bits), LU005 (16bits) e FX001D (16bits velocidade 2x (2x Speed)).
- Sony CDU 31/33A.
- CDROM Sound Blaster não-SCSI.
- CDROM Matsushita/Panasonic.
- CDROMs IDE compatíveis com o padrão ATAPI.

Todo equipamento não-SCSI é reconhecidamente mais lento do que os SCSI, e alguns drives de CDROM ATAPI podem não funcionar corretamente.

A partir da versão 2.2, todos os CDROM do FreeBSD distribuídos pela FreeBSD Mall podem ser iniciados (booting) diretamente pela unidade de CD.

P: Quais drives de CD-RW são suportados pelo FreeBSD?

R: O FreeBSD suporta qualquer tipo de unidade CD-RW ou CD-R IDE compatíveis com o padrão ATAPI. No FreeBSD 4.0 e posteriores, leia a página de manual do burncd(8).Em versões anteriores, veja os exemplos de utilização desses equipamentos em `/usr/share/examples/atapi` .

O FreeBSD também suporta qualquer drive de CD-R ou CD-RW do tipo SCSI. Instale o aplicativo `cdrecord` a partir da coleção de `ports` ou como pacote, e tenha certeza de ter o device `pass` compilado no seu *kernel*.

P: O FreeBSD suporta ZIP Drives?

R: O FreeBSD suporta ZIP Drives do tipo SCSI, é claro. Essa unidade deve ser configurada apenas nos SCSI ID números 5 ou 6, mas se a sua BIOS tem suporte à inicializãço(boot) pela unidade SCSI, essa característica pode ser usada sem problemas. Não está claro exatamente quais adaptadores SCSI suportam a característica de inicializãço(boot) em IDs diferentes de 0 ou 1, portanto será necessário consultar o manual do seu equipamento para obter informações mais precisas sobre esse recurso.

Os ZIP Drives padrão ATAPI (IDE) são suportados pelo FreeBSD desde a versão 2.2.6 e em todas as posteriores.

O FreeBSD tem suporte ainda a ZIP Drives de Porta Paralela desde a versão 3.0. Caso seu sistema seja dessa versão ou superior, verifique o suporte a `scbus0`, `da0`, `ppbus0`,

`vp0` no seu *kernel* (o *kernel* GENERIC tem todos esses suportes, exceto à device `vp0`). Com esses suportes presentes no *kernel*, o drive de Porta Paralela deve estar disponível em `/dev/da0s4` . Os discos ZIP podem ser montados usando o comando `mount /dev/da0s4 /mnt` OU (discos formatados como DOS) `mount_msdos /dev/da0s4 /mnt` , como é de costume.

Verifique também o FAQ sobre discos removíveis disponível ainda nesse capítulo, e as notas sobre "formatação" no capítulo de Administração.

P: O FreeBSD suporta discos JAZ, EZ ou outras unidades removíveis?

R: Fora a versão IDE dos discos EZ, os outros discos são todos do tipo SCSI, e portanto devem todos ser reconhecidos como discos SCSI no FreeBSD. O drive EZ tipo IDE deve ser reconhecido como disco IDE.

Não há uma certeza quanto à forma que o FreeBSD trata uma alteração de mídia enquanto o sistema está em pleno uso, então é necessário desmontar a unidade antes de trocar de disco e garantir que qualquer unidade externa esteja ligada quando o sistema for bootado, de forma que o FreeBSD possa reconhecê-las.

Veja essa nota sobre "formatação".

P: Que dispositivos seriais de múltiplas portas são suportados pelo FreeBSD?

R: Existe uma lista dessas unidades na seção de Dispositivos Diversos do Manual do FreeBSD.

Alguns dispositivos clones parecem também funcionar normalmente no sistema, em especial equipamentos que se dizem ser AST compatíveis.

Verifique a página de manual do sio(4) para obter mais informações quanto à configuração desses dispositivos.

P: O FreeBSD suporta meu teclado USB?

R: O suporte a dispositivos USB foi adicionado no FreeBSD desde a versão 3.1. Contudo, na versão 3.1, o suporte ainda é muito preliminar, e alguns equipamentos podem não funcionar antes da versão 3.2. Caso você queira usar o suporte a teclados USB, tente o seguinte.

1. No FreeBSD 3.2 ou posterior.

2. Adicione as seguintes linhas no arquivo de configuração do seu *kernel*, e recompile-o.

```
device  uhci
device  ohci
device  usb
device  ukbd
```

```
options KBD_INSTALL_CDEV
```

Em versões anteriores à 4.0, use:

```
controller      uhci0
controller      ohci0
controller      usb0
controller      ukbd0
options         KBD_INSTALL_CDEV
```

3. No diretório `/dev`, crie os seguintes devices:

```
# cd -/dev
# ./MAKEDEV kbd0 kbd1
```

4. Edite o `/etc/rc.conf` e adicione as seguintes linhas:

```
usbd_enable="YES"
usbd_flags=""
```

Depois de reiniciado(rebooting) o sistema, o teclado AT aparece como `/dev/kbd0i` e o teclado USB aparece como `/dev/kbd1` , se ambos estiverem conectados ao sistema. Se estiver somente o teclado USB, ele estará como `/dev/ukbd0` .

Caso queira usar o teclado USB no console, é necessário informar explicitamente ao driver do console que ele deve usar esse teclado. Isso pode ser feito com o seguinte comando em tempo de inicialização do sistema:

```
# kbdcontrol --k -/dev/kbd1 < -/dev/ttyv0 > -/dev/null
```

Note que se o teclado USB for o único teclado disponível, ele será acessado via `/dev/kbd0` , portanto a linha de comando deve-se parecer com:

```
# kbdcontrol --k -/dev/kbd0 < -/dev/ttyv0 > -/dev/null
```

O arquivo `/etc/rc.i386` é um bom lugar para colocar o comando acima.

Depois de configurado, o teclado USB deve funcionar também no ambiente X, sem nenhuma outra configuração especial.

Conectar e desconectar o teclado USB com o sistema ligado ainda não é um comportamento completamente suportado, portando é aconselhável ligar o teclado antes de iniciar o sistema e apenas desligá-lo depois que o computador estiver desligado, para evitar possíveis problemas.

Veja a página de manual do ukbd(4) para maiores informações.

P: Eu tenho um mouse de barramento não tradicional. Como o configuro?

R: O FreeBSD suporta o barramento de mouse tradicional do tipo InPort fabricados pela Microsoft, Logitech e ATI. O suporte a esse periférico é compilado por padrão

no *kernel* GENERIC do FreeBSD na versão 2.X, mas não é suportado por padrão na versão 3.0 e posteriores. Se você quer recompilar um *kernel* com suporte ao barramento de mouse, adicionando a linha ao seu arquivo de configuração:

No FreeBSD 3.0 e anteriores, adicione:

```
device mse0 at isa? port 0x23c tty irq5 vector mseintr
```

Na série 3.X do FreeBSD, a linha deve ser:

```
device mse0 at isa? port 0x23c tty irq5
```

E na série 4.X e posteriores, a linha deve ser:

```
device mse0 at isa? port 0x23c irq5
```

Barramentos de mouse costumam ter uma interface dedicada que permite definir o endereço de memória e a IRQ que a placa vai funcionar. Nesse caso, refira-se ao manual do seu equipamento e à página de manual do mse(4) para maiores informações.

P: Como eu uso o meu mouse (“mouse port” ou “keyboard”) PS/2?

R: Em versões posteriores ao 2.2.5, o *kernel* do FreeBSD inclui por padrão o suporte ao device `psm`, que controlará seu mouse PS/2 desde o momento de inicialização do sistema.

Caso seu FreeBSD seja 2.1.X ou similar, o suporte a PS/2 pode ser incluído no *kernel*, no momento da instalação, ou mesmo depois desse processo, com a opção `-c` na tela de `boot:` do sistema. O suporte nesse caso é desabilitado por padrão e por isso deve ser explicitamente habilitado.

Caso sua versão de FreeBSD seja antiga, adicione a seguinte linha ao seu *kernel* e recompile-o:

No FreeBSD 3.0 e anteriores, a linha é:

```
device psm0 at isa? port -"IO_KBD" conflicts tty irq 12 vector ↺
psmintr
```

No FreeBSD 3.1 e posteriores da mesma série, a linha deve ser:

```
device psm0 at isa? tty irq 12
```

No FreeBSD 4.0 e posteriores, a linha é:

```
device psm0 at atkbdc? irq 12
```

Veja a seção de configuração do *kernel* no Manual do FreeBSD caso você não tenha experiência com a compilação do *kernel*.

Uma vez detectado o `psm0` durante a inicialização (*boot*) do seu sistema, tenha certeza de que existe uma entrada `psm0` no `/dev`. Faça o seguinte:

```
# cd -/dev; sh MAKEDEV psm0
```

logado como usuário `root`.

P: É possível usar mouse de alguma forma, fora do sistema X Windows?

R: Se estiver utilizando o driver padrão de console, o syscons, pode-se usar o mouse para copiar & colar texto. Execute o mouse daemon, `moused`, para habilitar o mouse nos consoles virtuais da seguinte forma:

```
# moused --p -/dev/xxxx --t yyyy
# vidcontrol --m on
```

Onde *xxxx* deve ser substituído pelo nome de device do seu mouse e *yyyy* pelo tipo de protocolo do mesmo. Veja a página de manual do moused(8) para maiores informações quanto aos tipos de protocolos suportados.

É provável que você queira usar o mouse daemon automaticamente, sempre que o FreeBSD for iniciado. Na versão 2.2.1, defina as seguintes variáveis, no arquivo `/etc/sysconfig`.

```
mousedtype="yyyy"
mousedport="xxxx"
mousedflags=""
```

Da versão 2.2.2 até a 3.0, defina as seguintes variáveis no `/etc/rc.conf`.

```
moused_type="yyyy"
moused_port="xxxx"
moused_flags=""
```

Da versão 3.1 em diante, caso você tenha um mouse PS/2 é necessário apenas adicionar a opção `moused_enable="YES"` no arquivo `/etc/rc.conf`.

E se a intenção é usar o mouse em todos os terminais virtuais ao invés de apenas no console, insira a seguinte linha no `/etc/rc.conf`.

```
allscreens_flags="-m on"
```

Desde a versão 2.2.6 do FreeBSD, o mouse daemon é capaz de detectar o tipo de protocolo do mouse automaticamente, a não ser que o dispositivo em questão seja um mouse serial muito velho. Defina `auto` para que o programa identifique o protocolo do mouse automaticamente.

Quando o daemon está rodando, o acesso ao dispositivo deve ser coordenado entre ele e qualquer outra aplicação, como o X-Windows, por exemplo. Leia uma outra pergunta sobre esse assunto.

P: Como eu copio e colo com o mouse em um console modo texto?

R: Uma vez configurado o mouse (veja a pergunta anterior), aperte o botão 1 (botão esquerdo) do mouse e mova o cursor por toda a região desejada, selecionando o texto em questão. Depois, basta apertar o botão 2 (do meio) ou o botão 3 (direito) para colar o conteúdo selecionado anteriormente.

A partir da versão 2.2.6 o botão 2 cola o texto copiado, enquanto o botão 3 ``extende'' a região selecionada. Caso seu mouse não tenha o botão do meio, é possível remapear (ou emular) os botões do periférico usando algumas opções específicas do mouse daemon. Veja a página de manual do moused(8) para maiores detalhes.

P: O FreeBSD suporta mouse USB?

R: No FreeBSD 3.1 existe um suporte preliminar à recursos USB que não funciona muito bem dependendo da situação. A partir da versão 4.0 o FreeBSD suporta dispositivos USB por padrão. Caso queira usar um mouse USB no FreeBSD 3.X, siga as seguintes instruções.

1. Atualize seu sistema para FreeBSD 3.2 ou posterior.

2. Adicione o seguinte suporte ao seu *kernel*, e recompile-o:

```
device  uhci
device  ohci
device  usb
device  ums
```

Em versões anteriores à 4.0 o suporte à ser adicionado é:

```
controller        uhci0
controller        ohci0
controller        usb0
device            ums0
```

3. Entre no diretório `/dev` e crie os devices necessários:

```
# cd -/dev
# ./MAKEDEV ums0
```

4. Edite o `/etc/rc.conf` e adicione as linhas:

```
moused_enable="YES"
moused_type="auto"
moused_port="/dev/ums0"
moused_flags=""
```

```
usbd_enable="YES"
usbd_flags=""
```

Veja a seção anterior para uma discussão mais detalhada sobre o moused.

5. Para configurar o mouse USB no X, edite o `XF86Config` e, caso esteja usando o XFree86 3.3.2 ou posterior, adicione as seguintes linhas na seção *Pointer*:

```
Device    -"/dev/sysmouse"
Protocol        -"Auto"
```

Caso esteja usando uma versão anterior do Xfree86, adicione também na seção *Pointer* as seguintes linhas:

```
Device    -"/dev/sysmouse"
Protocol        -"SysMouse"
```

Leia também uma outra pergunta sobre o uso do mouse em ambiente X.

Conectar e desconectar o teclado USB com o sistema ligado ainda não é um comportamento completamente suportado, portando é aconselhável ligar o teclado antes de iniciar o sistema e apenas desligá-lo depois que o computador estiver desligado, para evitar possíveis problemas.

P: Eu tenho um mouse do tipo Wheel com uma rodinha e botões adicionais. Posso usá-lo no FreeBSD?

R: A resposta, infelizmente é, "Depende". Esse tipo de mouse tem algumas características especiais que requerem o uso de *drivers* especiais na maioria dos casos. A não ser que o device do seu mouse tenha suporte específico, ou se a aplicação em questão reconhecer esse tipo de equipamento, ele irá funcionar simplesmente como um mouse tradicional de dois ou três botões.

Mais informações sobre o uso de mouse do tipo Wheel em ambiente X Windows, refira-se a essa seção.

P: Por que meu mouse PS/2 do tipo Wheel fica louco, pulando pela tela?

R: O suporte ao mouse PS/2 no FreeBSD 3.2 e anteriores é falho quanto a mouses do tipo Wheel, incluindo o modelo M-S48 da Logitech e seus similares OEM. Aplique o seguinte patch no arquivo `/sys/i386/isa/psm.c` e recompile seu *kernel*:

```
Index: psm.c
===================================================================
RCS file: -/src/CVS/src/sys/i386/isa/Attic/psm.c,v
retrieving revision 1.60.2.1
retrieving revision 1.60.2.2
diff --u --r1.60.2.1 --r1.60.2.2
--- psm.c        1999/06/03 12:41:13 1.60.2.1
+++ psm.c        1999/07/12 13:40:52 1.60.2.2
@@ --959,14 +959,28 @@
```

```
     sc->mode.packetsize = vendortype[i].packetsize;

     -/* set mouse parameters */
+#if 0
+    -/*
+     * A version of Logitech FirstMouse+ won't report wheel ↺
movement,
+     * if SET_DEFAULTS is sent...  Don't use this command.
+     * This fix was found by Takashi Nishida.
+     */
     i = send_aux_command(sc->kbdc, PSMC_SET_DEFAULTS);
     if (verbose >= 2)
         printf("psm%d: SET_DEFAULTS return code:%04x\n", unit, ↺
i);
+#endif
     if (sc->config & PSM_CONFIG_RESOLUTION) {
         sc->mode.resolution
             = set_mouse_resolution(sc->kbdc,
-                (sc->config & PSM_CONFIG_RESOLUTION) -- 1);
+                       (sc->config & PSM_CONFIG_RESOLUTION) -- ↺
1);
+    -} else if (sc->mode.resolution >= 0) {
+        sc->mode.resolution
+            = set_mouse_resolution(sc->kbdc, sc-
>dflt_mode.resolution);
+    -}
+    if (sc->mode.rate > 0) {
+        sc->mode.rate = set_mouse_sampling_rate(sc->kbdc, sc-
>dflt_mode.rate);
     -}
+    set_mouse_scaling(sc->kbdc, 1);

     -/* request a data packet and extract sync. bits */
     if (get_mouse_status(sc->kbdc, stat, 1, 3) < 3) {
```

Em versões posteriores à 3.2, o suporte deve funcionar.

P: Como eu uso o mouse/bolinha-de-rolagem/touchpad no meu laptop?

R: Por gentileza, leia a pergunta anterior. Verifique também a página de Computação Móvel do Projeto.

P: Que tipos de dispositivos de fitas são suportados pelo FreeBSD?

R: O FreeBSD suporta dispositivos de fitas do tipo SCSI e QIC-36 (com interface QIC-02). Tal suporte inclui drives 8-mm (também conhecidos como Exabyte) e unidades de fita DAT.

Alguns dispositivos 8-mm mais antigos não são compatíveis com o padrão SCSI-2 e por isso podem não funcionar bem no FreeBSD.

P: O FreeBSD suporta bibliotecas de fitas?

R: O FreeBSD suporta alternadores (também conhecidos com carrosséis) SCSI, usando o device ch(4) e o comando chio(1). Os detalhes relativos a como controlar o alternador de fitas podem ser encontrados na página de manual do chio(1).

Caso você não esteja usando o AMANDA ou qualquer outro produto que entenda o funcionamento dos alternadores, lembre-se que tal equipamento simplesmente alterna a posição da fita, de um compartimento para outro, e portanto deve-se saber em qual compartimento a fita está e para qual ela deve voltar.

P: Quais placas de som são suportadas pelo FreeBSD?

R: O FreeBSD suporta as placas SoundBlaster, SoundBlaster Pro, SoundBlaster 16, Pro Audio Spectrum 16, AdLib e Gravis UltraSound. Existe ainda um suporte - limitado, é verdade - para as placas MPU-401 e placas MIDI compatíveis. Placas de som que estiverem em conformidade com a especificação MSS (Microsoft Sound System) também são suportadas pela controladora pcm do *kernel*.

Nota

Esse suporte é específico para apenas som! Exceto no caso das placas SoundBlaster, o suporte não inclui controle de joysticks, CDROMs ou SCSI. A interface SCSI da SoundBlaster e alguns CDROMs não-SCSI são suportados, mas o sistema não pode iniciar(booting) a partir desses dispositivos.

P: Qual a solução para falta de som da minha placa es1370 com o controlador pcm?

R: Basta aumentar o volume do seu som ;-) Use os seguintes comandos, sempre que o sistema iniciar:

```
# mixer pcm 100 vol 100 cd 100
```

P: Quais placas de rede o FreeBSD suporta?

R: Veja a seção de Placas Ethernet do Manual do FreeBSD para uma lista detalhada dos dispostivos suportados.

P: Eu não tenho um co-processador matemático. Isso é ruim?

R:

Nota

Vale apenas para proprietários de 386/486SX/486SLC - outras máquinas terão um co-processador integrado à CPU.

No geral, a falta de um co-processador matemático não traz nenhum problema, mas existem algumas circunstâncias onde você encontrará sérias limitações, seja no desempenho ou na precisão da emulação dos seus cálculos (veja a seção de emulação FP). Por exemplo, a renderização de círculos e arcos no ambiente gráfico será uma tarefa muito lenta. É recomendável comprar um co-processador matemático; vale a pena.

Nota

Alguns co-processadores matemáticos são melhores que outros. É estranho ter que dizer isso, mas ninguém nunca se deu mal ao comprar co-processadores Intel, portanto tenha certeza absoluta que o produto vai funcionar com o FreeBSD antes de comprar um clone.

P: Que outros dispositivos o FreeBSD suporta?

R: Veja o Manual do FreeBSD para obter uma listagem dos outros dispostivos suportados.

P: O FreeBSD suporta gerenciamento de energia no meu laptop?

R: O FreeBSD suporta APM em alguns computadores. Por gentileza, refira-se ao arquivo `LINT` de configuração do *kernel*; procure pela palavra APM. Mais informações na página de manual do apm(4).

P: Por que placas Micron travam na inicialização(boot)?

R: Algumas placas-mãe Micron não tem conformidade na implementação de sua BIOS e por isso confundem o FreeBSD no momento da inicialização(boot), pois os equipamentos em questão não foram configurados nos endereços que a BIOS reportou.

Procure a opção “Plug and Play Operating System” - ou algo parecido - na sua BIOS e desabilite-a para corrigir o problema. Mais informações sobre esse

problema podem ser encontradas em http://cesdis.gsfc.nasa.gov/linux/drivers/vortex.html#micron.

P: Por que o FreeBSD não reconhece nenhuma controladora SCSI Adaptec?

R: As séries mais novas (AIC789x) dos chips Adaptec tem suporte no modo CAM SCSI, que será redefinido na versão 3.0 do FreeBSD. Na versão 2.2-STABLE, você pode aplicar as correções disponíveis em ftp://ftp.FreeBSD.org/pub/FreeBSD/development/cam/. Caso você precise instalar um sistema com essas controladoras, existe um disquete de inicialização(boot) com suporta a CAM, disponível em http://people.FreeBSD.org/~abial/cam-boot/. Nos dois casos leia o arquivo README antes de tomar qualquer ação.

P: Por que o FreeBSD não encontra o meu Modem Plug & Play interno?

R: Será necessário adicionar o ID PnP do modem na lista de drivers seriais do sistema para que ele reconheça-o normalmente. Isso requer hackear um pouco o sistema. Pra habilitar o suporte Plug & Play, compile um novo *kernel* com a opção `controller pnp0` e reinicie o seu FreeBSD. O *kernel* irá mostrar os IDs PnP de todos os dispositivos que ele encontrar, no momento da inicialização(boot). Copie o ID PnP do modem em questão para a tabela no arquivo `/sys/i386/isa/sio.c` , por volta da linha 2777. Procure a expressão `SUP1310` na estrutura `siopnp_ids[]` para encontrar essa tabela. Recompile o seu *kernel*, instale-o e reinicie o sistema. Seu modem deve ser encontrado.

Provavelmente será necessário configurar o dispositivo PnP manualmente, usando o comando `pnp` no momento do boot, como a seguir:

```
pnp 1 0 enable os irq0 3 drq0 0 port0 0x2f8
```

para forçar detecção do modem.

P: O FreeBSD suporta software modems, como os Winmodems?

R: O FreeBSD suporta alguns software modems por meio de programas adicionais. A aplicação comms/ltmdm disponível na coleção de `Ports` do FreeBSD suporta os modems baseados no popular chipset Lucent LT. A aplicação comms/mwavem suporta o modem em laptops IBM Thinkpad 600 e 700.

Não é possível instalar o FreeBSD via software modem, visto que os programas adicionais para controlar esse equipamento só podem ser configurados depois que o sistema operacional já está instalado.

P: Como eu faço para o interpretador(prompt) de inicialização(boot): aparecer no console serial?

R:
1. Construa um *kernel* com a opção `options COMCONSOLE` .
2. Crie o arquivo /boot.config e coloque os caracteres `-P` como o único texto no arquivo.

3. Desligue o teclado do computador.

Leia o arquivo `/usr/src/sys/i386/boot/biosboot/README.serial` para mais informações.

P: Por que a placa de rede PCI da 3Com não funciona com motherboards Micron?

R: Algumas placas-mãe Micron não tem conformidade na implementação de sua BIOS e por isso confundem o FreeBSD no momento do boot, pois os equipamentos em questão não foram configurados nos endereços que a BIOS reportou.

Procure a opção "Plug and Play Operating System" - ou algo parecido - na sua BIOS e desabilite-a para corrigir o problema.

Mais informações sobre esse problema podem ser encontradas em http://cesdis.gsfc.nasa.gov/linux/drivers/vortex.html#micron

P: O FreeBSD suporta Multiprocessamento Simétrico (SMP)?

R: SMP é suportado a partir do FreeBSD 3.0-STABLE. O suporte ao SMP (multiprocessamento simétrico) não está disponível por padrão no *kernel GENERIC*, portanto é necessário compilar um novo *kernel* para habilitar o suporte SMP. Veja o arquivo `/sys/i386/conf/LINT` para descobrir quais opções são necessárias adicionar ao seu *kernel*.

P: O disquete de inicialização(boot) trava em um computador cuja placa-mãe é a ASUS K7V. O que eu faço?

R: Entre na configuração da BIOS da sua placa e desligue a opção "boot virus protection".

Capítulo 5. Resolução de Problemas

P: O que fazer quando meu disco rígido tiver bad blocks?

R: Com controladoras SCSI, o drive (HD) deveria ser capaz de remapear blocos ruins e corrigí-los automaticamente. Porém, muitos desses discos mantém essa função desabilitada por alguma razão misteriosa...

Para habilitar essa função é necessário editar o primeiro modo de página do dispositivo, o qual pode ser feito com o comando abaixo (como `root`)

```
# scsi --f -/dev/rsd0c --m 1 --e --P 3
```

E mudando os valores de AWRE e ARRE de 0 para 1:-

```
AWRE (Auto Write Reallocation Enbld):  1
ARRE (Auto Read Reallocation Enbld):  1
```

Os parágrafos seguintes foram enviados por Ted Mittelstaedt <tedm@toybox.placo.com >:

Para os discos IDE, qualquer bad block é considerado um sinal de dificuldade em potencial. Todos os discos IDE modernos já vêm com um remapeador interno que realoca bad blocks por outros blocos em bom estado, automaticamente. Todos os disco rígido IDE fabricados hoje em dia oferecem garantias extensas.

Se ainda quiser tentar salvar um drive IDE com bad blocks, pode fazer um download do programa de diagnóstico e correção do próprio fabricante do disco rígido. Às vezes estes programas podem fixar e forçar eletronicamente o disco a marcar estes blocos ruins e desativá-los.

Em discos ESDI, RLL e MFM, a existência de bad blocks é normal e não representa nenhum sinal de dificuldade,geralmente. Em um PC, a placa controladora das unidades de disco e, a BIOS se encarregam da tarefa de re-mapear os bad blocks. Isso funciona em sistemas operacionais como DOS que usa código da BIOS para acessar o disco. Porém, o driver (software controlador) do FreeBSD não trabalha ou acessa comandos da BIOS para para interagir com o drive (HD), então um mecanismo chamado bad144, existente no FreeBSD, acaba substituindo esta funcionalidade. O bad144 só trabalha com o drive wd (portanto, não é suportado no FreeBSD 4.0), e não pode ser usado com drive SCSI. O bad144 trabalha marcando e organizando setores ruins encontrados no HD, em um arquivo especial no disco.

Uma característica do bad144 - o bloco danificado é colocado em um arquivo especial situado na última trilha do disco. Como este arquivo contém uma lista de setores defeituosos que pode incluir valores perto do início do disco, onde o /

kernel pode estar alocado, esse arquivo deverá ser acessível ao bootstrap para que o programa - por meio da BIOS - leia o *kernel*; isso significa que o disco usado com bad144 não deve exceder 1024 cilindros, 16 cabeças, e 63 setores, logo, temos um limite efetivo de 500MB para discos mapeados com o bad144..

Para ativar o uso do bad144, simplesmente defina a opção de procurar por "Bad Block" como ON na tela do fdisk do FreeBSD, durante a instalação. Essas instruções funcionam a partir do FreeBSD 2.2.7, e o disco deve ter menos que 1024 cilindros. Geralmente recomenda-se que a unidade de disco esteja em operação durante pelo menos 4 horas antes de executar o bad144, permitindo assim a expansão térmica do HD.

Se o disco tem mais de 1024 cilindros (como um disco ESDI grande) a controladora ESDI usa um tipo de tradução especial em modo DOS. A device wd também entende esses mesmos modos de tradução e conversão, isso se o "tradutor" de geometria for ativado como "geometria fixa", quando particionado pelo fdisk. Você também não deve usar o modo "dangerously dedicated" para criar partições do FreeBSD, pois isso ignora o tipo de geometria. Embora o fdisk use a geometria definida pelo usuário, ele continua reconhecendo o tamanho verdadeiro do disco, e tentará criar uma partição maior para o FreeBSD. Se a geometria de disco for alterada para geometria traduzida(translated geometry), a partição DEVE ser criada manualmente, informando os números de blocos do HD.

Um truque rápido é usar um disco grande ESDI com uma controladora ESDI, iniciar(booting) o sistema com um disco DOS e formatar uma partição DOS. Depois reiniciar o sistema com um disco de instalação do FreeBSD, e anotar o número e tamanho dos blocos que serão apresentados na tela do fdisk para a partição DOS. Redefina a geometria do disco com os valores anotados, apague a partição DOS e crie uma partição FreeBSD "cooperativa". Defina essa partição como bootável e habilite o reconhecimento de bad blocks. Na instalação, o bad144 é carregado antes que qualquer outro sistema de arquivos seja criado (você pode ver isso com um Alt+F2). Se houver problemas na criação do arquivo de definições de setor danificado (o arquivo de badsector) é porque a geometria definida é maior do que o seu valor real - reinicie o sistema e comece todos os procedimentos novamente, inclusive o particionamento e formatação da partição DOS.

Se o remapeamento já estiver habilitado e os problemas com bad block continuarem, considere a substituição imediata do disco, pois os danos e os bad blocks aumentarão consideravelmente com o passar do tempo.

P: Por que o FreeBSD não reconhece a minha controladora SCSI EISA Bustek 742a?

R: As informações a seguir são para o modelo 742a, mas provavelmente também servem para as placas Buslogic. (Bustek = Buslogic).

Existem duas "versões" tradicionais da placa 742a. São os equipamentos de revisão A-G e de revisão H; as letras de cada revisão são colocadas depois do número de fabricação, ao lado das placas. A placa 742a possui 2 chips ROM acoplados, o

primeiro é o chip da BIOS e o segundo é o do Firmware. Para o FreeBSD a versão da BIOS é irrelevante, mas a versão do Firmware é uma informação fundamental. É interessante dizer que, se você fizer uma chamada ao departamento de suporte da Buslogic, eles irão te enviar um upgrade desses ROMs, e é muito bom sempre manter a versão mais recente do ROM do seu Firmware para a versão de revisão do seu equipamento.

As placas cuja letra de revisão é A-G aceitam apenas atualizações da BIOS/Firmware de versão 2.41/2.21 respectivamente. A revisão H (REV H) aceita as versões mais recentes da BIOS/Firmware até a versão 4.70/3.37. A principal diferença entre as versões do Firmware é que a versão 3.37 tem suporte a"round robin".

As placas Buslogic também tem um número serial. Caso seu equipamento seja antigo, tente abrir uma chamada no departamento de RMA da Buslogic e informe-os o número de série da sua placa. Se ela estiver entre os seriais de abrangência, a Buslogic vai aceitar seu equipamento para revisão.

O FreeBSD 2.1 aceita apenas as revisões de Firmware até o 2.21. Caso o seu Firmware seja mais antigo do que o 2.21 sua placa não será reconhecida como Buslogic. Contudo, é possível que o equipamento seja reconhecido como Adaptec 1540, já que os Firmware mais antigos da Buslogic possuem um modo de "emulação" da AHA1540, o que não é uma boa coisa, para uma placa EISA.

Caso seu Firmware seja antigo e você conseguiu obter uma revisão para a versão 2.21, não se esqueça que é necessário alterar o jumper W1 da posição A-B (padrão) para posição B-C ao atualizar a placa.

P: Por que o FreeBSD não detecta a controladora SCSI do Netserver HP??

R: Isso já é um problema conhecido. A controladora SCSI on-board EISA dos servidores HP Netserver estão no slot EISA número 11, portanto todos os "verdadeiros" slots EISA estão na sua frente. O endereço definido para os slots EISA >= 10 ocupa um endereço compartilhado com o barramento PCI, e portanto entra em conflito com seus recursos. Essa é uma situação onde a configuração automática do FreeBSD não se comporta muito bem.

Portanto o que você deve fazer, é fingir que não existe limitação quanto ao intervalo de endereços, definindo a opção option `EISA_SLOTS` do *kernel* para o valor 12. Configure e compile um novo *kernel*, conforme descrito no capítulo de configuração do *kernel* no Manual do FreeBSD.

Obviamente esse problema é ainda maior quando se trata de uma nova instalação. Para corrigir esse problema é necessário uma pequena alteração no modo *UserConfig*. Não use a interface de configuração "visual" do *kernel*, use a Interface de Linha de Comando (CLI), simplesmente digitando:

```
eisa 12
quit
```

na tela do modo CLI, e continue a instalação do FreeBSD como de costume. De qualquer forma, é recomendável recompilar e instalar um novo *kernel* depois da instalação do sistema..

Futuras versões do FreeBSD terão esse problema corrigido automaticamente.

Nota

Não use discos em modo `dangerously dedicated` com um HP Netserver. Veja essa nota para maiores informações.

P: O que está havendo com minha controladora IDE CMD640?

R: Está com defeito! Não suporta mais comandos nos dois canais de forma simultânea.

Existe uma correção disponível e automaticamente habilitada, se você usa uma controladora com esse chip. Para maiores detalhes, refira-se a página de manual da controladora de disco (wd(4)).

Se o FreeBSD em questão é o FreeBSD 2.2.1 ou 2.2.2 com essa controladora em questão, e você quer usar o segundo canal, compile um novo *kernel* com a opção `options "CMD640"` habilitada. Essa configuração é padrão para o FreeBSD 2.2.5 e posteriores.

P: Eu sempre vejo mensagens como ed1: timeout. O que elas significam?

R: Normalmente esse problema é causado por um conflito de interrupções (por exemplo, duas placas usando a mesma IRQ). O FreeBSD até a versão 2.0.5R costumava ser tolerante quanto a esse problema e a placa de rede continuava funcionando mesmo com IRQ conflitantes. Contudo desde a versão 2.0.5R os conflitos de interrupções não são mais tolerados. Inicie o sistema com a opção de boot -c e mude as device ed0/de0/... para o valor correspondente ao da placa.

Caso esteja usando um conector BNC na sua placa de rede, é provável que existam device timeout por causa de má terminação do barramento. Pra tirar isso a limpo coloque um terminador direto na placa (sem cabos) e veja se as mensagens de erro param.

Algumas placas compatíveis NE2000 apresentarão esse problema caso a porta UTP não receba sinal de link, ou se o cabo estiver desconectado.

P: Por que minha placa 3COM 3C509 parou de funcionar sem motivo aparente?

R: Esse cartão tem um hábito horrível de perder suas informações de configuração. Redefina as informações da placa usando o programa de DOS chamado `3c5x9.exe` .

P: Minha impressora paralela está ridiculamente lenta. O que devo fazer?

R: Se o único problema é a lerdeza terrível da sua impressora, tente mudar seu modo da porta de impressão conforme discutido na seção de Configuração de Impressoras no Manual do FreeBSD.

P: Por que alguns programas ocasionalmente morrem com erro de Signal 11 ?

R: Erros de sinal 11 são fruto de tentativas de acesso indevido a memória. Esse acesso normalmente é controlado pelo sistema operacional, e quando o sistema não permite acessar determinados endereços, o processo é morto com signal 11. Se isso estiver acontecendo em intervalos aleatórios de tempo, é preciso investigar as causas com cuidado.

Esse problema normalmente é atribuído a:

1. Se o problema ocorre apenas em um programa específico que você mesmo esta desenvolvendo, se trata de um bug no código do seu programa.

2. Se o problema é com algum programa que faz parte da base do FreeBSD, é possível que também seja um problema de bug no código em questão. Contudo, esses problemas costumam ser corrigidos antes que os usuários tradicionais percebam sua existência - e necessitem ler este `FAQ` - afinal, é para isso que o -CURRENT existe.

Em especial, uma indicação que esse problema *não* é um bug do FreeBSD, é um erro repetitivo no mesmo instante da compilação, mas o problema que o compilador apresenta muda de linha a cada nova compilação.

Por exemplo, suponha que você esteja executando um “make buildworld”, e a compilação falha na hora de compilar o `ls.c` em `ls.o`. Se você rodar o “make buildworld” de novo e a compilação falha exatamente no mesmo trecho do código, então o problema realmente é com o fonte da aplicação, nesse caso atualize os fontes do FreeBSD e tente novamente. Agora se a compilação falhar em um trecho diferente do código, é quase certo que o problema seja físico, ou seja, com o seu equipamento.

O que deve ser feito:

Em primeiro lugar, deve-se usar um debugador, como o gdb, por exemplo, para encontrar o ponto exato do código que está tentando acessar um endereço problemático de memória, e corrigi-lo.

Em segundo lugar, verifique se a culpa não é do seu equipamento.

As causas mais comuns para o problema incluem::

1. Os seus discos rígidos podem estar superaquecidos: Verifique se o sistema de ventilação do seu PC está funcionando. Verifique coolers internos (da fonte) e externos, e verifique se não existe superaquecimento de outros componentes do computador.

2. O processador está superaquecido: Pode ser porque foi feito um overclock no processador em questão, ou no caso mais tradicional, pode ser que o cooler tenha parado de funcionar ou que esteja sujo e portanto funcionando em rotação baixa. Em ambos os casos, o primeiro passo é garantir que o processador esteja rodando sob as mesmas condições que ele foi construído para funcionar - por exemplo, com a velocidade do clock original e com a ventilação adequada.

   Caso tenha sido feito overclock no processador, lembre-se que é mais barato usar um computador um pouco mais lento, do que trocar o processador da máquina por causa de um chip fritado ;-) Além do que a maioria das pessoas não são simpatizantes de overclock, mesmo que você considere a ação segura ou não.

3. Caso tenha múltiplos pentes de memória SIMM/DIMM, tente desligá-los e experimente usar cada pente de uma vez, indiviualmente. Com isso é possível descobrir se o problema é com algum chip DIMM/SIMM ou se o problema é a combinação entre os pentes.

4. Configurações super otimistas na BIOS da sua placa mãe são outra causa provável. Algumas BIOS tem opções que permitem alterar a velocidade e frequência de vários recursos. Normalmente os valores padrão na BIOS são os mais conservadores, e portanto devem ser o bastante para controlar corretamente o equipamento; contudo algumas opções como por exemplo "RAM Speed: Turbo" ou alguma opção parecida coloca o estado de espera para o acesso a memória em um valor muito baixo, e as vezes, por mais otimista que você seja, sua memória pode não ser rápida o bastante. O ideal é definir os valores padrão da sua BIOS, mas é interessante anotar os valores atuais primeiro!

5. Alimentação insuficiente de energia na placa-mãe. Caso exista alguma placa que não esteja sendo utilizada, algum disco rígido ou CDROM, é interessante desliga-los temporariamente do computador, ou simplesmente remover o cabo de energia desses equipamentos. Mesmo em sub utilização, essas placas e discos estão sob constante alimentação e talvez sua fonte consiga suprir uma carga menor. Ou tente trocar a fonte do seu PC, de preferência por uma com maior poder de alimentação (por exemplo, se a sua fonte é de 250 Watts troque por uma de 300 Watts).

Leia ainda o FAQ SIG11 (disponível a seguir) que tem outras boas explicações sobre esses problemas. O FAQ também discute como alguns programas de teste de memória podem pensar que um pente problemático está funcionando corretamente.

Finalmente, se nenhum dos casos acima ajudou a solucionar seu problema, pode ser que exista um bug no FreeBSD. Você deve seguir as intruçõoes para enviar um relatório de problemas para o Projeto FreeBSD.

Existe um FAQ extenso que cobre esse assunto, disponível no FAQ dos problemas com SIG11.

P: O meu sistema trava com o erro Fatal trap 12: page fault in kernel mode, ou panic:, e sai mostrando uma quantidade enorme de informações. O que eu faço?

R: A equipe de desenvolvimento do FreeBSD tem muito interesse nesse tipo de erro, mas é necessário obter algumas informações suplementares, do que apenas o erro que você está tendo. Copie sua mensagem de erro inteira, consulte o FAQ sobre *kernel* panics, compile um *kernel* em modo de debugação e tente analisar o problema. Parece uma tarefa difícil, mas não é necessário conhecimento de programação; basta seguir as instruções.

P: Por que a tela fica preta e perde sincronia quando eu inicio o sistema?

R: Esse é um problema conhecido da placa de vídeo ATI Mach 64. O problema é que essa placa usa o endereço `2e8`, o mesmo utilizado pela quarta porta serial dos computadores pessoais. Devido a um bug (ou uma vantagem?) da sio(4) ,essa porta será sempre reconhecida, ainda que não exista a quarta porta serial no seu computador, ou *mesmo* se o sio3 (a quarta porta) estiver desabilitado.

Até que o bug seja corrigido, você pode usar essa solução:

1. Entre no modo de configuração do *kernel* com opção `-c` na tela de inicialização(boot). (Isto colocara o *kernel* no modo de configuração).

2. Desabilite a `sio0`, `sio1`, `sio2` e `sio3` (todas elas). Dessa forma será ativada, logo, você não terá -> problemas.

3. Digite exit para continuar o boot.

Caso queira usar as portas seriais, será necessário construir um *kernel* customizado, com as seguintes alterações: no fonte `/usr/src/sys/i386/isa/sio.c` encontre a ocorrência da expressão `0x2e8` e apague essa expressão e a vírgula que a antecede (mantenha a outra). Depois compile o novo *kernel* normalmente.

Mesmo depois dessa correção, é provável que o X Windows ainda não funcione como esperado. Se for o caso, garanta que a versão do Xfree86 em questão seja ao menos o XFree86 3.3.3 ou uma versão superior. Esse XFree86 e os posteriores tem suporte nativo às placas de vídeo Mach64, e tem inclusive um X Server dedicado para tal equipamento.

P: Por que o FreeBSD só detecta 64MB de memória RAM se eu tenho 128MB instalados?

R: Devido à maneira que o FreeBSD obtém as informações quanto ao tamanho da memória disponível por intermédio da BIOS, pode acontecer de apenas 16 bits válidos serem detectados (65535 Kbytes = 64MB) ou até menos, dependendo da BIOS (em alguns casos, apenas 16MB). Mesmo nessa situação o FreeBSD tenta detectar mais que 64MB de memória, mas esse reconhecimento pode falhar.

Pra corrigir esse problema pode ser usada a opção do *kernel* descrita a seguir. Existe uma forma de obter informações completas quanto ao tamanho da memória, a partir da BIOS, mas devido a algumas limitações isso nem sempre é possível hoje em dia. Futuramente será. De qualquer forma, temos ainda a opção do *kernel* para situações onde toda a memória não puder ser reconhecida.

```
options "MAXMEM=n"
```

Onde *n* equivale à memória (em Kilobytes) disponível no sistema. Para 128 MB de memória, use o valor `131072`.

P: Por que o FreeBSD 2.0 entre em pânico com a mensagem kmem_map too small!?

R:

Nota

A mensagem em questão também pode ser `mb_map too small!`

Essa mensagem de pânico indica que o sistema ficou sem memória suficiente pros buffers de rede (especificamente, os mbuf clusters). A quantidade de memória virtual disponível para os clusters mbuf pode ser elevada com a opção::

```
options "NMBCLUSTERS=n"
```

no arquivo de configuração do seu *kernel*, onde *n* equivale ao valor entre 512-4096, dependendo do número de conexões TCP simultâneas que você espera poder suportar. O valor 2048 é recomendável, e provavelmente será o bastante para sanar o problema que causa o pânico em questão O número de clusters mbuf em uso pode ser monitorado com o comando `netstat -m` (veja netstat(1)). O valor padrão para a variável NMBCLUSTERS no *kernel* do FreeBSD é `512 + MAXUSERS * 16`.

P: Por que o erro /kernel: proc: table is fullocorre?

R: O *kernel* do FreeBSD limita o número máximo de processos simultâneos existentes no sistema. O número em questão é baseado na opção `MAXUSERS`. do sistema. A opção `MAXUSERS` afeta ainda inúmeros outros limites do *kernel* do FreeBSD, como por exemplo os buffers disponíveis para o stack de rede do sistema (veja esta resposta anterior). Se o computador estiver sob grande carga, provavelmente será

necessário aumentar o `MAXUSERS` . Essa alteração aumentará os limites do sistema em adição ao número de processos permitido.

Desde a versão 4.4 do FreeBSD, o valor para `MAXUSERS` se tornou configurável, não sendo mais necessário recompilar o *kernel* para alterá-lo, bastando definir a variável `kern.maxusers` no arquivo `/boot/loader.conf` . Em versõoes mais recentes do FreeBSD, deve-se ajustar o `MAXUSERS` em sua configuração do *kernel*.

Caso seu sistema não esteja muito carregado, mas o número de processos simultâneos ainda assim é alto basta definir a variável `kern.maxproc` com o sysctl. Em casos especiais, onde esses inúmeros processos estão sendo executados por um único usuário, será preciso alterar ainda alterar o valor da variável `kern.maxprocperuid` para um a menos do que o valor de `kern.maxproc` . (deve ser ao menos 1 processo a menos, visto que ao menos o init(8) do sistema vai estar sempre em execução.)

Para tornar uma alteração de variável do sysctl permanente, defina-a no arquivo `/etc/sysctl.conf` nas versões mais recentes do FreeBSD, ou então no arquivo `/etc/rc.local` em versões mais antigas do sistema.

P: Por que acontece o erro CMAP busy quando eu reinicio com um novo *kernel*?

R: A lógica que tenta detectar uma data errada nos arquivos `/var/db/kvm_*.db` as vezes é falha, o que leva o sistema a entrar em pânico.

Se for o caso, reinicie seu sistema em modo monousuário e faça:

```
# rm -/var/db/kvm_*.db
```

P: O que a mensagem ahc0: brkadrint, Illegal Host Access at seqaddr 0x0 significa??

R: Trata-se de um conflito com o Ultrastor SCSI Host Adapter.

Durante o processo de inicialização(boot), entre no menu de configuração do *kernel* e desabilite a `uha0`, que esta causando o problema.

P: Quando eu inicio o sistema, encontro o erro ahc0: illegal cable configuration, mas o meu cabo está certo. O que está havendo?

R: A placa-mãe em questão não consegue se dar bem com o suporte a terminação automática do barramento. Altere sua BIOS SCSI para a terminação correta, de acordo com a configuração do equipamento, ao invés de usar terminação automática. O driver AIC7XXX não consegue descobrir se o reconhecimento externo dos cabos (e conseqüente auto-terminação) está disponível, e portanto ele simplesmente assume que o suporte existe, caso a configuração da EEPROM serial esteja definida como automatic termination. Sem o reconhecimento de cabo externo o driver irá sempre configurar a terminação de forma incorreta, o que compromete a confiabilidade do barramento SCSI.

P: Por que o Sendmail retorna um erro que diz “mail loops back to myself”?

R: Essa pergunta é respondida no FAQ do próprio Sendmail, e diz:-

> * Eu estou tendo problemas de configurações local "Local configuration error" como essas:
>
> 553 relay.domain.net config error: mail loops back to myself
> 554 <user@domain.net>... Local configuration error
>
> Como posso resolver esse problema?
>
> Você definiu que as mensagens enviadas para o domínio
> em questão (domain.net) devem ser repassadas para uma outra
> estação específica (nesse caso para
> relay.domain.net) usando um registro MX, mas essa máquina de
> relay não se reconhece como a estação
> responsável pelas mensagens do domínio domain.net.
> Adicione domain.net no arquivo /etc/mail/local-host-names
> (caso você esteja usando FEATURE(use_cw_file)) ou então
> adicione a linha "Cw domain.net" em /etc/mail/sendmail.cf.

Atualmente a versão mais recente do FAQ do sendmail é mantida em sincronia com as versões mais atuais do MTA, mas ela ainda é enviada regularmente para os grupos de notícias comp.mail.sendmail, comp.mail.misc, comp.mail.smail, comp.answers, e news.answers. Ainda é possível receber um cópia por e-mail do FAQ, enviando uma mensagem para <mail-server@rtfm.mit.edu > com o comando send usenet/news.answers/mail/sendmail-faq no corpo da mensagem.

P: Porque algumas aplicações que usam tela inteira não se comportam muito bem em estações remotas?

R: A estação remota deve estar definindo o terminal como algum tipo diferente do cons25 que é o tipo de terminal usado pelo console do FreeBSD.

Existem inúmeras correções para esse problema:

- Depois de logar-se na estação remota, defina a variável de ambiente TERM como ansi ou sco caso a máquina em questão tenha informações quanto a esse tipo de terminal.

- Use um emulador VT100 como o screen no console do FreeBSD. O screen oferece a possibilidade de usar múltiplas sessões concorrentes em um mesmo terminal, e é um grande programa. Cada janela do screen se comporta como um terminal VT100, portanto a variável TERM deve ser definida como vt100.

- Instale a base do cons25 na estação remota. A maneira correta de faze-lo depende do sistema operacional em questão na estação remota. Consulte os manuais de administração do sistema remoto em questão para descobrir como faze-lo.

- Levante um X server do lado FreeBSD da coisa e acesse a estação remota usando um terminal baseado no ambiente X, como o `xterm` ou o `rxvt`. A variável TERM deve ser definida como`xterm` ou `vt100` no lado remoto.

P: Por que meu computador apresenta a mensagem calcru: negative time...?

R: Esse comportamento pode ser causado por diversos motivos relacionados a interrupções de *hardware* e/ou software. Pode ser devido a algum bug, mas também pode acontecer por causa da natureza de alguns devices. Por exemplo, usar TCP/IP via porta paralela com uma MTU muito grande é uma boa forma de provocar esse comportamento. Aceleradores gráficos também são eficientes para criar esse tipo de problema, nesse caso, sendo necessário analisar as configurações de interrupções do software.

Um efeito colateral desse problema são processos que morrem"SIGXCPU exceeded cpu time limit".

No FreeBSD 3.0 e sistemas posteriores a 29 de Novembro de 1998, caso o problema não possa ser solucionado de outra forma, uma correção pode ser definir a seguinte variável do sysctl:

```
# sysctl --w kern.timecounter.method=1
```

Isso causa um impacto de performance, mas dependendo do problema que causava esse comportamento, é provável que nem consiga-se notar a mudança nessa performance. Se o problema continuar, mantenha a variável do sysctl habilitada e defina a opção `NTIMECOUNTER` no *kernel* para valores crescentes. Se chegar a um ponto em que foi necessário definir `NTIMECOUNTER=20` e o problema ainda não tiver sido resolvido, as interrupções são serias demais e seu comportamento não é confiável.

P: Acontece da pcm não ser encontrada, com a mensagem pcm0 not found ou então minha placa de som é encontrada na `pcm1` mas no meu *kernel* a entrada se refere a `device pcm0`. O que está havendo?

R: Isso acontece no FreeBSD 3.X com placas de som PCI. A `pcm0` é reservada exclusivamente para placas de som ISA e por isso se a placa em questão é PCI, ela será reconhecida como `pcm1` e a mensagem em questão pode acontecer.

Nota

Não é possível evitar a mensagem de advertência simplesmente alterando o seu *kernel* e definindo `device pcm1` pois isso resultará na `pcm1` sendo reservada para placas ISA,

e o seu equipamento PCI será reconhecido na `pcm2` (e a mensagem de advertência pcm1 not found continuará).

Caso sua placa de som seja PCI ainda será preciso criar a device `snd1` ao invés da `snd0`:

```
# cd -/dev
# ./MAKEDEV snd1
```

Esse comportamento não ocorre na série 4.X do FreeBSD, muito trabalho foi feito para tornar o sistema mais *PnP-centric* e a device `pcm0` não é mais reservada exclusivamente para placas ISA.

P: Porque a minha placa PnP não é mais encontrada (ou é encontrada como `unknown`) desde a atualização para o FreeBSD 4.X?

R: O FreeBSD 4.X é muito mais *PnP-centric* do que as versões anteriores, e isso causou alguns efeitos distintos em algumas placas PnP (como algumas placas de som e alguns modems interno por exemplo) que não funcionam mais da forma como funcionavam no FreeBSD 3.X.

O motivo para esse comportamento é explicado no seguinte e-mail, que foi enviada na lista freebsd-questions pelo Reter Wemm, respondendo uma pergunta sobre um modem interno que não era mais reconhecido no FreeBSD depois de atualizar o sistema para versão 4.X (os comentários entre `[]`foram adicionados com a intenção de explicar melhor o contexto da mensagem).

Nota

Os índices dessa citação foram atualizados de seu texto original

> A bios PNP configurou ele [o modem] e o deixou conectado na porta em questão, por isso o estilo antigo [no 3.X] “reconhece” o equipamento ISA.
>
> No FreeBSD 4 o código ISA é bem mais PnP-centric. Era possível [no 3.X] encontrar uma placa ISA que funcionava com “determinada” device e depois, o id PNP da mesma device encontrava a mesma placa novamente, como se fosse uma outra usando os mesmos recursos do sistema, e por isso ele falhava, como se fosse um conflito de recursos. Portanto, agora ele desabilita o suporta às placas programáveis de forma que essa

> confusão e dupla detecção de *hardware* não ocorra. Essa mudança implica também na necessidade de se saber previamente os ids PnP para cada tipo de equipamento suportado, aumentando um pouco mais a lista de TODO do suporte PnP no sistema.

Para fazer o equipamento voltar a funcionar, é necessário encontrar seu PnP id e adicioná-lo a lista de devices ISA reconhecidas como PnP. Essa informação pode ser obtida usando pnpinfo(8) que detecta a configuração dos equipamentos. Por exemplo, veja a saída do pnpinfo(8) de um modem interno:

```
# pnpinfo
Checking for Plug-n-Play devices...

Card assigned CSN #1
Vendor ID PMC2430 (0x3024a341), Serial Number 0xffffffff
PnP Version 1.0, Vendor Version 0
Device Description: Pace 56 Voice Internal Plug & Play Modem

Logical Device ID: PMC2430 0x3024a341 #0
        Device supports I/O Range Check
TAG Start DF
    I/O Range 0x3f8 -.. 0x3f8, alignment 0x8, len 0x8
        [16-bit addr]
    IRQ: 4  -- only one type (true/edge)
```

[algumas linhas com TAG foram eliminadas]

```
TAG End DF
End Tag

Successfully got 31 resources, 1 logical fdevs
-- card select # 0x0001

CSN PMC2430 (0x3024a341), Serial Number 0xffffffff

Logical device #0
IO:  0x03e8 0x03e8 0x03e8 0x03e8 0x03e8 0x03e8 0x03e8 0x03e8
IRQ 5 0
DMA 4 0
IO range check 0x00 activate 0x01
```

A informação que você quer é a linha "Vendor ID" no começo da saída do comando. O valor hexadecimal entre parênteses (0x3024a341 esse caso) é PnP id e o conjunto de caracteres que o antecede (PMC2430) é a identificação ASCII única.

Alternativamente, se o pnpinfo(8) não listou sua placa em questão, o pciconf(8) pode ser usado preferivelmente. Esta é a saída do comando `pciconf -vl` de uma placa de som onboard:

```
# pciconf --vl
```

```
chip1@pci0:31:5:        class=0x040100 card=0x00931028 ↺
chip=0x24158086 rev=0x02 hdr=0x00
    vendor   = -'Intel Corporation'
    device   = -'82801AA 8xx Chipset AC'97 Audio Controller'
    class    = multimedia
    subclass = audio
```

Aqui deve-se usar o valor do `chip`, "0x24158086".

Tais informações (Vendor ID ou valor do chip) precisam ser adicionadas ao arquivo `/usr/src/sys/isa/sio.c` .

Primeiro faça uma cópia de segurança do `sio.c` no caso de algo dar errado e também para que você possa fazer um patch para enviar junto com o seu Relatório de Problemas (você vai enviar um PR, não vai?) e depois edite o `sio.c` e procure a linha

```
static struct isa_pnp_id sio_ids[] = {
```

Depois analise as linhas logo abaixo para encontrar o lugar apropriado para sua placa. As entradas na tabela ficam todas parecidas com essa logo abaixo, e são ordenadas de acordo com a identificação ASCII do fabricante do produto a qual deve ser incluída como comentário na coluna do lado do código em questão, e junto com a *descrição da placa* ou parte dela, conforme identificada na saída do pnpinfo(8):

```
{0x0f804f3f, NULL},     -/* OZO800f -- Zoom 2812 (56k Modem) */
{0x39804f3f, NULL},     -/* OZO8039 -- Zoom 56k flex */
{0x3024a341, NULL},     -/* PMC2430 -- Pace 56 Voice Internal ↺
Modem */
{0x1000eb49, NULL},     -/* ROK0010 -- Rockwell -? */
{0x5002734a, NULL},     -/* RSS0250 -- 5614Jx3(G) Internal ↺
Modem */
```

Adicione o ID hexadecimal do fabricante da placa no local apropriado, salve o arquivo e recompile o *kernel*, depois reinicie o sistema. Agora a sua placa deve ter sido encontrada como uma device `sio` exatamente como era encontrada no FreeBSD 3.X

P: Qual a causa do erro nlist failed quando eu executo, por exemplo, o `top` ou o `systat` ?

R: O problema é que o programa que você está tentando executar tenta ler alguma informação específica do *kernel*, baseando-se no *kernel* symbol em questão, mas por algum motivo, essa informação não pode ser encontrada; esse erro é causado por um dos seguintes problemas:

- O *kernel* e o userland do sistema não estão em sincronia (por exemplo, você compilou um *kernel* novo e o instalou sem de dar um `installworld` , ou vice-versa), e por isso a tabela de informações dos *kernel* symbols é diferente do que o programa pensa que é. Se esse for o caso basta completar os procedimentos

de atualização do sistema (veja o arquivo `/usr/src/UPDATING` para a correta seqüência de ações).

- O `/boot/loader` não está sendo usado para carregar o *kernel* dessa estação, ao invés dele, o boot2 (veja boot(8)) está sendo usado diretamente. Apesar de não ter problema algum deixar de usar o `/boot/loader` , ele costuma se comportar melhor na hora de tornar os *kernel* symbols disponíveis para aplicações em nível de usuário.

P: Porque demora tanto para a conexão se estabelecer via `ssh` ou `telnet` ?

R: O sintoma: existe um atraso muito grande entre o estabelecimento da conexão TCP e o momento que o programa cliente pede a senha (ou no caso do telnet(1), quando a tela de login aparece).

O problema: o programa servidor dessa transação leva muito tempo tentando resolver o nome da estação cliente que está se conectando. A maioria dos servidores, incluindo os servidores Telnet e SSH que vem junto com o FreeBSD tentam resolver o número IP do cliente no nome da estação, para, entre outras coisas, gravar essa informação em um arquivo de log para referências futuras por parte do administrador.

A solução: Se o problema acontece apenas quando você (o cliente) tenta se conectar no servidor, o problema é com o lado cliente da transação; se o problema acontece com qualquer estação que tente se conectar ao computador (servidor) então o problema é do lado servidor.

Se o problema é com o cliente, a única maneira de corrigir o problema é configurar corretamente o servidor DNS que responde autoritativamente pelo endereço da estação. Se for uma rede local considere esse comportamento um problema do servidor, e continue lendo; se a conexão deve ser estabelecida na rede global (internet) , então entre em contato com o seu Provedor de Serviços Internet e solicite que eles corrijam o problema.

Se o problema é do lado servidor, e a rede em questão, se trata de uma rede local, será necessário configurar o servidor de forma que ele consiga resolver os endereços dos clientes em nomes. Veja as páginas de manual do hosts(5) e named(8) para obter mais informações. Se a conexão é na Internet, provavelmente o resolvedor (lado cliente do serviço de nomes) do seu servidor não está funcionando corretamente. Pra fazer o teste, tente descobrir o endereço IP do site `www.yahoo.com` por exemplo. Se não funcionar, esta ai o problema.

P: O que a mensagem stray IRQ quer dizer?

R: Stray IRQs são sintomas de *hardware* que interrompe o pedido de interrupção no meio de um ciclo de autorização de interrupção.

Existem três formas de tratar o problema:

- Aprenda a conviver com as mensagens de advertência. De qualquer forma, todas as mensagens exceto as 5 primeiras para cada IRQ são suprimidas pelo sistema mesmo.

- Evite o inconveniente alterando de 5 para 0 na função `isa_strayintr()` o número de mensagens antes de suprimir as advertências.

- Evite as advertências instalando algum equipamento de porta paralela que use a IRQ 7 e o driver PPP (é o usual, na maioria dos sistemas) e instale algum driver IDE ou qualquer outro dispositivo que use a IRQ 15 e seu respectivo suporte.

P: Por que a mensagem file: table is full aparece repetidas vezes no dmesg?

R: Esse erro indica que você excedeu o número máximo de descrevedores (descriptors) de arquivos no sistema. Leia a seção kern.maxfiles da capítulo de Ajuste de Limites do *Kernel* no Manual do FreeBSD do FreeBSD para obter mais informações sobre o problema.

P: Por que o relógio do meu laptop mantem a hora incorreta?

R: Rode o comando dmesg(8), e procure algumas linhas com a expressão `Timecounter`. A última linha encontrada será o relógio que o FreeBSD escolheu, e com certeza ele será `TSC`.

```
# dmesg -| grep Timecounter
Timecounter -"i8254"  frequency 1193182 Hz
Timecounter -"TSC"  frequency 595573479 Hz
```

Essa informação pode ser confirmada ao verificar a variável `kern.timecounter.hardware` do sysctl(3).

```
# sysctl kern.timecounter.hardware
kern.timecounter.hardware: TSC
```

A BIOS do laptop altera a frequência do relógio TSC de forma a modificar a velocidade do processador quando o computador estiver sendo utilizado com baterias, ou se o mesmo entrar em modo de economia de energia. O FreeBSD não faz distinção entre frequência do clock e modos especiais de trabalho, e por isso pode atrasar ou adiantar a hora do sistema.

Esse exemplo, o laptop em questão tem dois relógios; portanto o `i8254` pode ser definido como padrão na variável `kern.timecounter.hardware` do sysctl(3).

```
# sysctl --w kern.timecounter.hardware=i8254
kern.timecounter.hardware: TSC --> i8254
```

Agora o seu laptop deve conseguir manter a data e hora de forma mais precisa.

Pra tornar essa alteração automática, adicione a seguinte linha no arquivo `/etc/sysctl.conf` .

```
kern.timecounter.hardware=i8254
```

P: Por que o meu laptop não funciona muito bem na hora de identificar cartões PCMCIA?

R: Esse problema é comum em laptops que tem mais de um sistema operacional instalado. Alguns sistemas não-BSD fazem os cartões PCMCIA ficarem em um estado inconsistente, causando um reconhecimento problemático dos dispositivos por parte do `pccardd` , como por exemplo, detectando os cartões como "(null)""(null)" ao invés da sua marca e modelo verdadeiros.

É necessário desligar completamente a alimentação de energia do equipamento para garantir que o mesmo seja completamente resetado. Desligue completamente o laptop (não suspenda seu funcionamento, não o deixe entrar em modo de espera, conhecido como standby, a alimentação deve ser completamente interrompida), espere alguns - poucos - minutos e reinicie o laptop. Tudo deve correr bem.

Alguns laptops são grandes mentirosos quando afirmam estar desligados. Se o procedimento acima não funcionar, tire a bateria do laptop, espere alguns minutos e ligue o sistema novamente.

P: Por que o bootloader do FreeBSD mostra a mensagem Read error e pára completamente logo após a tela da BIOS?

R: O inicializador do FreeBSD reconheceu a geometria de disco de forma incorreta e por isso esse valor deve ser definido manualmente com o fdisk(8) ao criar ou modificar uma partição FreeBSD.

A geometria correta do disco pode ser verificada na BIOS do computador. Procure pelo número de cilindros, cabeças e de setores do disco em questão.

No fdisk do sysinstall(8), aperte a tecla G para definir a geometria do disco manualmente.

Irá aparecer uma janela de diálogo perguntando o número de cilindros, cabeças e setores do disco. Defina esses valores, conforme anotados da BIOS do sistema e separados por barras.

5000 cilindros, 250 cabeças e 60 setores, por exemplo, seria definido como **`5000/250/60.`**

Aperte ENTER para confirmar os valores e depois aperte a tecla W para escrever as novas informações na tabela de partições do disco.

P: Outro sistema operacional destruiu o meu gerenciador de inicialização(Boot Manager). Como eu o recupero?

R: Entre no sysinstall(8) e escolha o menu Configure, seguido do Fdisk. Escolha o disco onde o gerenciador de boot costumava ficar e aperte a barra de espaços(space). Depois aperte a tecla W para escrever as novas informações no disco. Vai aparecer uma tela, perguntando o que deve ser instalado na MBR do disco. Escolha o Gerenciador de inicialização(Boot Manager), e ele será reinstalado.

P: O que o erro swap_pager: indefinite wait buffer: quer dizer?

R: Quer dizer que existe um processo tentando paginar uma área da memória para o disco, e que esse processo demorou mais de 20 segundos; portanto falhou. É provável que a causa desse erro sejam blocos defeituosos no disco, falha nos cabos, ou até mesmo algum outro erro de I/O relacionado ao *hardware*. Se o disco estiver danificado, serão apresentadas mensagens de erro referentes ao mesmo em `/var/log/messages` e também na saída do `dmesg`. Do contrário, verifique seus cabos e conectores.

Capítulo 6. Aplicações Comerciais

Nota

Esta seção se encontra ainda muito escassa, embora esperamos, naturalmente, que as empresas façam adições a ela! :) Os desenvolvedores do FreeBSD não tem interesses financeiros em nenhuma das empresas listadas aqui, mas apenas as listam como um serviço público (e sente que o interesse comercial no FreeBSD pode ter muitos efeitos positivos na viabilidade do uso do sistema a longo prazo). Nós encorajamos que os vendedores de softwares comerciais mandem seus softwares para inclusão. Consulte a página de Fabricantes para obter uma lista maior.

P: Onde eu posso conseguir *Office Suite*) para o FreeBSD?

R:

- A FreeBSD Mall oferece uma versão nativa do VistaSource ApplixWare 5 para o FreeBSD.

  O ApplixWare é uma poderosa suíte comercial de aplicações para escritório no FreeBSD. Ela contém um processador de texto, planilha de cálculos, um programa de apresentação e um pacote para desenho vetorial e outros aplicativos.

  O ApplixWare é vendido com parte integrante da edição de Desktops BSD da FreeBSD Mall.

- A versão para Linux do StarOffice funciona sem problemas no FreeBSD. A maneira mais fácil de instalar a versão para Linux do StarOffice é pela Coleção de `Ports` do FreeBSD. Versões futuras da suíte Open-Source de escritório,OpenOffice deverão funcionar também.

P: Onde posso conseguir o Motif para o FreeBSD?

R: O Open Group lançou o código fonte do Motif 2.1.30, que pode ser instalado como o pacote `open-motif` , ou então ser compilado pelo `Ports`. Consulte a seção sobre o `Ports` no Manual do FreeBSD para obter mais informações sobre o assunto.

Nota

O Open Motif pode ser redistribuído apenas se sua distribuição estiver sendo usada em sistemas operacionais open source.

Em contrapartida, existem distribuições comerciais do Motif disponíveis. Tais distribuições contudo não são gratuitas, mas suas licenças permitem que ele seja utilizando em softwares de código fechado. Contate a Apps2go para obter informações quanto a versão mais barata do ELF Motif 2.1.20 para FreeBSD (tanto para i386 quanto para Alpha).

Existem duas distribuições, a “development edition” e a “runtime edition” (bem mais barata). Tais distribuições incluem:

- OSF/Motif manager, xmbind, panner, wsm.
- Kit de Desenvolvimento com uil, mrm, xm, xmcxx, arquivos include e arquivos.
- Bibliotecas ELF estáticas e dinâmicas (para serem usadas com FreeBSD 3.0 e superiores).
- Applets de demonstração.

Lembre-se de especificar que você quer a versão para FreeBSD do Motif quando encomendado (não esqueça de mencionar a arquitetura que você quer também)! Versões para NetBSD e OpenBSD também são vendidas pela *Apps2go*. Atualmente o produto é apenas disponível para download via FTP.

Mais informações:
Página WWW da Apps2go

ou
<sales@apps2go.com > ou <support@apps2go.com >

ou
fone [EUA] (817) 431 8775 ou +1 817 431-8775

Contate Metro Link para obter informações quanto a versão ELF ou a versão a.out do Motif 2.1 para o FreeBSD.

Tal distribuição inclui:

- Gerenciador OSF/Motif, xmbind, panner, wsm.

- Kit de desenvolvimento com uil, mrm, xm, xmcxx, arquivos include e arquivos Imake.
- Bibliotecas estáticas e dinâmicas (não se esqueça de especificar que você quer o formato ELF, caso queira usar com o FreeBSD 3.0 e posteriores; ou o formato a.out para usar com FreeBSD 2.2.8 e anteriores).
- Applets de Demonstração.
- Páginas de manual previamente formatadas.

Certifique-se de especificar que você quer a versão para FreeBSD do Motif quando encomendá-lo! Versões para Linux também são vendidas pela *Metro Link*. O produto está disponível em CDROM ou download via FTP.

Contate a Xi Graphics para obter informações quanto a versão a.out do Motif 2.0 para o FreeBSD.

A distribuição inclui:

- Gerenciador OSF/Motif, xmbind, panner, wsm.
- Kit de desenvolvimento com uil, mrm, xm, xmcxx, arquivos include e arquivos Imake.
- Bibliotecas estáticas e dinâmicas (para o FreeBSD 2.2.8 e anteriores).
- Applets de Demonstração.
- Páginas de manual previamente formatadas.

certifique-se de especificar que você quer a versão para FreeBSD do Motif quando encomendá-lo! Versões para BSDI e para Linux também são vendidas pela *Xi Graphics*. Atualmente o produto se trata de um conjunto de 4 disquetes... futuramente se tornará uma distribuição única em CD, como o CDE da mesma empresa.

P: Onde posso adquirir o CDE para o FreeBSD?

R: A Xi Graphics costumava vender o CDE para o FreeBSD, mas não o faz mais.

O KDE é um Desktop de código fonte aberto para X11 similar ao CDE em muitos aspectos. Talvez você também aprecie o visual e as características do xfce. KDE e xfce estão ambos disponíveis no sistema de `ports` do FreeBSD.

P: Existe algum servidor X comercial de alta-performance?

R: Sim, a Xi Graphics e a Metro Link vendem produtos da Accelerated-X para o FreeBSD e para outros sistemas baseados em Intel.

O que a Metro Link oferece é um servidor X de alta-performance que possui um esquema de configuração extremamente fácil, fazendo uso da suíte de ferramentas de gerenciamento de pacotes do FreeBSD, com suporte a múltiplas placas de vídeo simultâneas e é distribuído apenas em forma binária, por meio conveniente de um download via FTP. Sem esquecer que, o produto oferecido pela Metro Link está disponível a um preço muito razoável, $39 dólares.

A Metro Link vende também o Motif e formato ELF e a.out para o FreeBSD (veja pergunta anterior).

informações:
Página WWW da Metro Link

ou
<sales@metrolink.com > ou <tech@metrolink.com >

ou
fone [EUA] (954) 938-0283 ou +1 954 938-0283

O produto oferecido pela Xi Graphics é um servidor X de alta performance, que oferece uma interface fácil de configuração e com suporte a múltiplas placas de vídeo simultâneas, e é distribuído apenas de forma binária em uma distribuição única para FreeBSD e Linux. A Xi Graphics oferece ainda um servidor X de alta performance com suporte desenvolvido especificamente para laptops.

Existe uma versão de"demonstração de compatibilidade" disponível na versão 5.0 do servidor gráfico.

A Xi Graphics vende ainda o Motif e o CDE para o FreeBSD (veja acima).

Mais informações:
Página WWW da Xi Graphics

ou
<sales@xig.com > ou <support@xig.com >

ou
fone [EUA] (800) 946 7433 ou +1 303 298-7478.

P: Existe algum sistema de Banco de Dados para o FreeBSD?

R: Sim! Veja a seção de Fabricantes Comerciais do Web site do FreeBSD.

Dê uma olhada também na seção de Databases da Coleção de `Ports` do FreeBSD.

P: Posso rodar o Oracle no FreeBSD?

R: Pode. As páginas a seguir descrevem exatamente como configurar o Oracle para Linux no FreeBSD:

- http://www.scc.nl/~marcel/howto-oracle.html
- http://www.lf.net/lf/pi/oracle/install-linux-oracle-on-freebsd

Capítulo 7. Aplicações de Usuário

P: Então, onde estão todas as aplicações de usuários?

R: Por gentileza, dê uma olhada na página do `Ports` para informações sobre pacotes de programas disponíveis na Coleção de `Ports` do FreeBSD. A lista atualmente ultrapassa 24,000 aplicações e está crescendo diariamente, então retorne à página e verifique freqüentemente as aplicações, ou entã inscreva-se na lista de discussão `freebsd-announce` mailing list para atualizações periódicas ou novas adições.

A maioria dos `ports` devem estar disponíveis para as versões 2.2, 3.x e 4.x, e muitos deles devem funcionar também em sistemas 2.1.x. Cada vez que um lançamento do FreeBSD é produzido, um *snapshot* da árvore do `ports` do momento da lançamento também é incluída no diretório `ports/`.

O FreeBSD também suporta o conceito de "pacote", que essencialmente nada mais é do que uma distribuição binária compactada com o gzip e com um pouco de inteligência extra embutido nesse pacote, para fazer o trabalho que é requerido para uma instalação customizada. Um pacote pode ser instalado e desinstalado repetidas vezes de forma fácil, sem ter que se conhecer os detalhes horrendos dos arquivos que ele inclui.

Use o menu de instalação de pacotes em `/stand/sysinstall` (sobre a opção do pos-configuration menu) ou invoque o comando pkg_add(1) nos arquivos de pacotes específicos que voc;ê quer instalar. Os pacotes podem ser identificados normalmente pelo sufixo `.tgz` e o pessoal da distribuição em CDROM tem um diretório `/packages/All` no cd que conté esses arquivos. Eles podem também ser baixados pela rede para várias versões do FreeBSD nos seguintes endereços: do (sobre a opção PostConfiguration do menu) ou invoque o comando pkg_add(1)

para o 2.2.8-RELEASE/2.2.8-STABLE
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-2.2.8/

para o 3.X-RELEASE/3.X-STABLE
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-3-stable/

para o 4.X-RELEASE/4-STABLE
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-4-stable/

para o 5.X-CURRENT
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/i386/packages-5-current

ou em um sítio espelho mais perto de você.

Note que nem todos os `Ports` podem estar disponíveis em formato de pacotes, visto que a atualização da Coleção de `Ports` do FreeBSD é muito freqüente, e novos programas são constantemente adicionados, e outros são atualizados. É sempre bom verificar periodicamente quais pacotes estão disponíveis no servidor FTP mestre do projeto FreeBSD, o ftp.FreeBSD.org.

P: Onde eu encontro a libc.so.3.0?

R: Você está tentando usar um pacote construído para o FreeBSD 2.2 ou para versões posteriores, em um sistema 2.1.X. Por gentileza, dê uma olhada na seção anterior e pegue o `port`/pacote correto para o seu sistema.

P: Por que eu estou tendo problemas cuja mensagem de erro mostra “Error: can't find libc.so.4.0”?.

R: Acidentalmente você pegou um pacote construído para FreeBSD 4.X ou para o 5.X e está tentando instala-lo no seu FreeBSD 2.X ou 3.X. Por favor, pegue a versão correta dos pacotes.

P: Por que o ghostscript gera um monte de erros no meu 386/486SX?

R: Deixe-me adivinhar. Você não tem um co-processador matemático, certo? Será necessário adicionar um co-processador matemático alternativo ao seu *kernel*; você pode fazer isso adicionando a seguinte linha no arquivo de configuração do seu *kernel*, e depois recompilá-lo:

```
options GPL_MATH_EMULATE
```

Nota

Quando você fizer isto, será necessário remover a opção `MATH_EMULATE`

P: Por que aplicações do SCO/iBCS2 bombardeiam o ``socksys''? (FreeBSD 3.0 e anteriores, apenas).

R: Primeiro, é necessário editar o arquivo `/etc/sysconfig` (ou `/etc/rc.conf` , veja o rc.conf(5)) e modificar a última seção, alterando para `YES` a seguinte variável:

```
# Set to YES if you want ibcs2 (SCO) emulation loaded at startup
ibcs2=NO
```

Essa alteração fará o sistema carregar os módulos de *kernel* do ibcs2 na inicialização.

Depois, será necessário alterar o /compat/ibcs2/dev para parecer com:

```
lrwxr-xr-x  1 root  wheel         9 Oct 15 22:20 X0R@ --> -/dev/
null
lrwxr-xr-x  1 root  wheel         7 Oct 15 22:20 nfsd@ --> ↺
socksys
-rw-rw-r--  1 root  wheel         0 Oct 28 12:02 null
lrwxr-xr-x  1 root  wheel         9 Oct 15 22:20 socksys@ --> -/
dev/null
crw-rw-rw-  1 root  wheel   41,   1 Oct 15 22:14 spx
```

É necessário que o socksys aponte para `/dev/null` (veja null(4)) para fingir o processo de abertura e fechamento do device. O código mais recente (`-CURRENT`) se encarregará dos outros detalhes. Essa maneira de trabalhar o socksys é bem mais limpa do que a forma que era usada anteriormente. Se você quer que o driver `spx` fique disponível para um socket X local, defina a opção `SPX_HACK` no *kernel* do FreeBSD quando você o recompilar.

P: Como eu configuro um sistema de INN (Internet News) na minha estação?

R: Depois de instalar o pacote news/inn ou o `port`, um excelente lugar para iniciar é Dave Barr's INN Page onde voçe encontrará o `FAQ` do INN.

P: Qual versão do Microsoft FrontPage eu devo usar?

R: Use os `Ports`, Luke! Uma versão previamente corrigida do Apache,apache13-fp, está disponível na árvore do `Ports`.

P: O FreeBSD suporta Java?

R: Veja a página http://www.FreeBSD.org/java/.

P: Por que eu não consigo compilar determinado `port` na minha estação 3.X-STABLE?

R: Caso esteja usando uma versão do FreeBSD significativamente mais velha do que o -CURRENT ou o -STABLE, será necessário usar o kit de atualização dos `Ports`, disponível em http://www.FreeBSD.org/ports/. Caso esteja atualizado mas ainda assim você tem dificuldades, é provável que alguém disponibilizou uma versão do programa que funciona perfeitamente no -CURRENT mas que não compila corretamente no -STABLE. Por gentileza, envie um Relatório de Problemas com o comando send-pr(1) porque a coleção de `Ports` deve funcionar tanto no -STABLE quanto no -CURRENT.

P: Onde posso encontrar o arquivo ld.so?

R: Algumas aplicações cujo formato binário é o a.out, como o Netscape Navigator, necessitam das bibliotecas a.out. Uma versão do FreeBSD nativamente construído com bibliotecas ELF não instala as bibliotecas a.out por padrão. Você terá problemas por não ter o `/usr/libexec/ld.so`, se esse for o caso do seu sistema. Estas bibliotecas estão disponíveis embutidas na distribuição compat22. Utilize o sysinstall(8) para instalá-los. Pode-se instalá-lo apartir do código fonte do FreeBSD:

```
# cd -/usr/src/lib/compat/compat22
# make install clean
```

Se quiser instalar as bibliotecas mais recentes do compat22 sempre que executar o `make world`, edite `/etc/make.conf` para incluir `COMPAT22=YES`. Bibliotecas de compatibilidade antigas raramente sofrem mudanças, as vezes nunca, então geralmente não é necessário.

Também veja as páginas de ERRATAS para o 3.1-RELEASE e 3.2-RELEASE.

P: Eu atualizei meus fontes. E agora, como eu atualizo meus `Ports` instalados?

R: FreeBSD não inclui uma ferramenta de atualização do `ports`, mas existem algumas ferramentas que tornam o processo de atualização dos `Ports` uma tarefa, digamos, fácil. É possível ainda instalar algumas ferramentas adicionais que facilitam o gerenciamento dos `Ports` instalados

O comando pkg_version(1) pode gerar um script que atualizará os `ports` instalados para as últimas versões da árvore de `Ports`.

```
# pkg_version --c > -/tmp/myscript
```

O script de saída *deve* ser editado manualmente antes de ser usado. As versões mais recentes do pkg_version(1) forçam a edição do arquivo, colocando um exit(1) no começo do script.

A saída do script deve ser salva pois ela gera informações sobre os pacotes que são dependências dos que estão sendo atualizados. Tais dependências podem precisar ser atualizadas ou não, dependendo de cada uma delas. Os casos comuns onde as dependências precisam ser atualizadas é quando a versão das bibliotecas compartilhadas foram alteradas, portanto o `port` que usava aquela biblioteca precisa ser atualizado para que a nova versão seja usada.

Caso tenha espaço o bastante em disco, pode ser interessante usar a ferramenta `portupgrade` para automatizar o processo de atualização das aplicações instalados por meio de `ports` ou pacotes. Posto que ele foi programado em Ruby, o `portupgrade` é um candidato improvável à se tornar parte da árvore principal do FreeBSD. Mas isso não evita que qualquer pessoa use o programa. Alias, ele é uma ótima ferramenta. Ele está disponível em sysutils/portupgrade.

Se a estação fica constantemente conectada, é interessante usar o sistema periodic(8) para gerar um relatório semanal sobre as versões do `Ports` que podem ser atualizadas. Pra configurar o sistema para isso, insira a linha `weekly_status_pkg_enable="YES"` no `/etc/periodic.conf`.

P: Por que o `/bin/sh` é tão pequeno? Por que o FreeBSD não usa o `bash` ou outro interpretador de comandos (*shell*)?

R: Porque o POSIX diz que é assim que deve ser um interpretador de comandos (*shell*)?

A reposta mais complicada: muitas pessoas precisam escrever scripts *shell* que sejam portáveis através de muitos sistemas. É por isso que o POSIX especifica o interpretador de comandos (*shell*) e comandos utilitários com tanto detalhe. A maioria dos scripts são escritos para o interpretador de comandos Bourne (Bourne *shell*), e várias interfaces importantes de programação (make(1), system(3), popen(3) e análogos em linguagens de alto-nível como Perl e Tcl) o usam como interpretador de comandos (*shell*) padrão. Por ser tão amplamente utilizado é importante que o interpretador de comandos (*shell*) Bourne seja rápido para carregar, seja determinístico em seu comportamento, e que tenha uma pequena alocação de memória.

A implementação atual é o nosso melhor esforço para encontrar a maior parte destes requerimentos simultaneamente. Como forma de manter o `/bin/sh` do menor tamanho possível, não incluímos muitas das características convenientes que outros interpretadores de comando (*shell*) possuem. É por isso que a Coleção de `Ports` disponibiliza outros interpretadores de comandos (*shell*) com características mais abrangentes, como o bash, o csh, o tcsh, e o zsh. (Você pode comparar a utilização de memória entre todos esses interpretadores de comandos (*shell*), analisando as colunas "VSZ" e "RSS" na saída do comando `ps -u`.)

P: Por que o Netscape e o Opera demoram tanto para iniciar?

R: A resposta tradicional é que o DNS no seu computador está mal configurado. O Netscape e o Opera fazem verificação de DNS ao iniciar, e por isso não se tornarão disponíveis até que obtenham uma resposta do servidor DNS ou até que eles determinem que a estação não está conectada na rede.

Capítulo 8. Configuraçãao do Kernel

P: Eu gostaria de configurar meu *kernel*. É difícil?

R: De modo algum! Veja a seção "kernel config" do Manual do FreeBSD.

Nota

Recomenda-se que você faça uma cópia datada do seu `kernel` na forma `/kernel.AAMMDD` e o diretório `/modules` para `/modules.AAMMDD` depois que estiver tudo funcionando. Desta forma se você fizer alguma bobagem quando mexer com a sua configuração, pode-se iniciar aquele *kernel* ao invés de ter que desfazer tudo novamente no `kernel.GENERIC`. Isso é particularmente importante se você estiver dando boot em um equipamento não suportado pelo *kernel* genérico (`GENERIC`).

P: A compilação do meu *kernel* falha porque está faltando o `_hw_float`. Como eu resolvo o problema?

R: Deixa eu adivinhar. Você removeu o `npx0` (veja npx(4)) do arquivo de configuração do *kernel* porque você não possui um co-processador aritmético, certo? Errado! :-) O `npx0` é *OBRIGATÓRIO*. Mesmo que você não tenha um co-processador aritmético, o dispositivo `npx0` *deve* ser incluido.

P: Por que meu *kernel* é tão grande (cerca de 10MB)?

R: Provavelmente, seu *kernel* foi compilado em *modo de depuração (`debug`)*. Um *kernel* construído em modo de depuração (`debug`) contêm muitos símbolos usados para depuração que aumentam muito o seu tamanho. Note que se você está executando um FreeBSD 3.0 ou superior, terá pouca, ou nenhuma, perda de performance por usar um *kernel* em modo de depuração (`debug`), sendo útil ter um para o caso de pane no sistema.

Entretanto, se você possui pouco espaço em disco, ou simplesmente não quer executar um *kernel* para depuração, certifique-se que os dois itens abaixo sejam verdadeiros:

- Não existe a seguinte linha no arquivo de configuração do *kernel*

```
makeoptions DEBUG=-g
```

- Você não está executando config(8) com a opção `-g`.

Ambas as situações acima fazem com que o *kernel* seja compilado no modo de depuração (`debug`). Tão logo você tenha certeza que não se enquadra naqueles itens, o *kernel* poderá ser compilado normalmente e notadamente diminuirá o tamanho; a maioria dos *kernels* tendem a ficar em torno de 1.5MB a 2MB.

P: Porque estão ocorrendo conflitos de interrupção com portas multi-seriais?

R: Quando se compila um *kernel* com suporte a porta multi-serial, ele avisa que somente a primeira porta é testada e as demais são ignoradas devido a conflitos de interrupção. Como eu conserto isto?

O problema, neste caso, é que o FreeBSD possui código para evitar que o *kernel* fique com lixo (*trashed kernel*) por causa de conflitos de hardware ou software. A maneira de corrigir isto é excluindo as definições de IRQ em todas as portas exceto uma. Veja o exemplo:

```
#
# Multiport high-speed serial line -- 16550 UARTS
#
device sio2 at isa? port 0x2a0 tty irq 5 flags 0x501 vector ↺
siointr
device sio3 at isa? port 0x2a8 tty flags 0x501 vector siointr
device sio4 at isa? port 0x2b0 tty flags 0x501 vector siointr
device sio5 at isa? port 0x2b8 tty flags 0x501 vector siointr
```

P: Porque todo *kernel* que eu tento construir falha na compilação, mesmo o GENERIC?

R: Existem várias causas possíveis para esse problema. Elas são, sem uma ordem particular:

- Você não está usando os comandos `make buildkernel` e `make installkernel`, e seus fontes estão estruturados de forma diferente daqueles usados para construir o sistema atual (por exemplo, está sendo um 4.3-RELEASE em um sistema 4.0-RELEASE). Se estiver sendo feita uma atualização, leia o arquivo `/usr/src/UPDATING`, prestando atenção ao final da seção "COMMON ITEMS".

- Você está usando os comandos `make buildkernel` e `make installkernel`, mas não garantiu a correta finalização do comando `make buildworld`. O `make buildkernel` depende de arquivos gerados pelo `make buildworld` para fazer seu trabalho corretamente.

- Mesmo que você esteja tentando construir um FreeBSD-STABLE, é possível que os fontes tenham sido pegos quando estavam sendo modificados, ou inconsistentes por alguma outra razão; somente os releases são absolutamente garantidos de

serem compilados, embora o FreeBSD-STABLE possa ser compilado com sucesso na grande maioria das vezes. Caso já não tenha conseguido, tente buscar os fontes novamente e veja se o problema já não foi resolvido. Tente um servidor diferente, para o caso daquele que está sendo usado estar com problemas.

Capítulo 9. Discos, Sistemas de Arquivos e Carregadores de Inicialização (Boot Loaders)

P: Como eu mudo todo o meu sistema operacional para um disco novo?

R: A melhor maneira de migrar seu sistema de um disco para o outro é reinstalar completamente o SO na nova unidade de armazenamento e depois migrar os dados dos usuários do disco antigo para nova instalação. Essa forma é a mais recomendada, caso o sistema tenha sido -STABLE por mais de uma versão, ou caso tenha atualizado um Release ao invés de ter instalado um novo sistema. O booteasy pode ser facilmente instalado com o comando boot0cfg(8), de forma a permitir que o sistema possa iniciar dois sistemas distintos, até que você esteja satisfeito com o novo sistema. Pule o próximo parágrafo para saber algumas formas seguras de migrar os dados dos usuários para o novo disco.

Você pode ter decidido por não refazer uma nova instalação; nesse caso será necessário reparticionar o novo disco com o `/stand/sysinstall` , ou fdisk(8) ou disklabel(8). Também é necessário instalar o booteasy em ambas as unidades com o comando boot0cfg(8), de forma que você possa alternar a inicialização entre o novo sistema e a configuração atual do mesmo, até que a cópia dos dados tenha sido efetuada.

Agora, com um novo disco configurado, você está pronto para começar a mover os dados da antiga para nova unidade de armazenamento. Infelizmente os dados não podem simplesmente ser copiados ao acaso. Existem alguns arquivos especiais (como os arquivos de dispositivos `/dev`), flags, e links que tendem a não funcionar no novo sistema, visto que esses arquivos ocupam inodes ou tem informações específicas da unidade, e por isso não podem ser copiados como um arquivo comum. É necessário usar ferramentas que entendam esse comportamento. Isso significa que você terá que usar o dump(8). É sempre uma boa idéia realizar esse processo de cópia de dados em modo mono-usuário, contudo tal precaução não é obrigatória - é apenas sinal de cuidado.

Não é aconselhável usar nenhuma outra ferramenta a não ser o dump(8) e restore(8) para copiar o sistema de arquivos da partição raiz ("/"). O tar(1) pode funcionar de forma satisfatória, mas pode ser que não. Também é ótima idéia usar o dump(8) e restore(8) para copiar (ou mover completamente) os dados em uma partição para uma outra partição vazia. Os passos necessários para usar o dump para copiar os dados de uma partição existente para uma nova partição são:

1. Crie um novo sistema de arquivos com o newfs(8) na nova partição.

2. Monte a partição em um ponto de montagem temporário.

3. Entre (cd) no ponto de montagem em questão.

4. Finalmente mova os dados da partição existente para a nova partição com o dump(8).

Por exemplo, se a intenção é copiar os dados da partição raiz para a partição `/dev/ad1s1a`, cujo ponto de montagem temporário é o `/mnt`, faça o seguinte:

```
# newfs -/dev/ad1s1a
# mount -/dev/ad1s1a -/mnt
# cd -/mnt
# dump 0af -- -/ -| restore xf --
```

Redefinir a estrutura das partições com o dump(8) é um processo um pouco mais trabalhoso. Caso você queira, por exemplo, unir o conteúdo da partição `/var` com as partições de nível acima, crie uma partição que seja grande o bastante para alocar o conteúdo de ambas, copie a partição principal como no exemplo descrito acima e depois copie as sub-partições para os diretórios vazios que o primeiro comando deve ter criado:

```
# newfs -/dev/ad1s1a
# mount -/dev/ad1s1a -/mnt
# cd -/mnt
# dump 0af -- -/ -| restore xf --
# cd var
# dump 0af -- -/var -| restore xf --
```

Para separar um diretório de sua estrutura atual, ou seja, no mesmo exemplo ainda, alocar os dados de `/var` em uma partição própria quando na definição atual o `/var` é apenas um diretório comum, é necessário montar a sub partição no diretório apropriado do ponto de montagem temporário, simulando assim o sistema de arquivos a ser criado a partir da raiz (montada no diretório temporário), depois basta copiar os dados do diretório antigo para nova partição:

```
# newfs -/dev/ad1s1a
# newfs -/dev/ad1s1d
# mount -/dev/ad1s1a -/mnt
# mkdir -/mnt/var
# mount -/dev/ad1s1d -/mnt/var
# cd -/mnt
# dump 0af -- -/ -| restore xf --
```

Talvez você prefira usar cpio(1), pax(1), tar(1) ao invés de dump(8) na hora de copiar os dados de usuários. Quando este `FAQ` foi escrito, esses comandos costumavam

perder atributos especiais dos arquivos ou mesmo alterar algumas permissões ou autoridade (dono dos arquivos), portanto use esses comandos com cuidado.

P: Um disco no modo "dangerously dedicated" pode ser prejudicial a minha saúde?

R: O processo de instalação permite a escolha entre dois modos distintos de particionar o(s) disco(s) rígido. A maneira tradicional permite que outros sistemas operacionais na mesma estação possam acessar essas partições, criando entradas na tabela do fdisk (entradas chamadas de "slices" no FreeBSD) sob uma partição FreeBSD própria. Uma característica desse particionamento é permitir múltiplos sistemas operacionais e permitir a instalação de um gerenciador de inicialização (*boot*) para alternar entre esses sistemas. A segunda maneira, a alternativa ao modo tradicional faz uso do disco todo para o FreeBSD e não faz esforços para se tornar compatível com outros sistemas operacionais.

Então, por que esse modo é chamado de "modo perigosamente dedicado"? Um disco particionado dessa forma não tem algumas características tradicionais que os PCs poderiam considerar entradas válidas do fdisk. Dependendo das circunstâncias o PC pode reclamar e gerar advertências sobre o disco em questão, assim que o primeiro contato com essa unidade seja feito, ou pior, o PC pode ainda danificar o processo de inicialização (*boot*) do BSD (*bootstrap*) sem pedir confirmação da alteração ou até mesmo sem avisar o usuário dessa mudança. O "modo perigosamente dedicado" ainda costuma confundir várias BIOS, inclusive as BIOS AWARD (encontradas, por exemplo no HP Netserver e em sistemas Micronics assim como em muitos outros) e as BIOS Symbios/NCR (da popular série 53C8xx de controladoras SCSI). Não são apenas esses dois modelos que podem apresentar dificuldades com esse modo de particionamento de disco, a lista completa é ainda maior. O principal sintoma desse tipo de confusão é a presença de mensagens de read error apresentada pelo *bootstrap* do FreeBSD quando ele tem dificuldades de encontrar-se; outro sintoma é a falta de sistema operacional no momento da inicialização do PC.

Então porque esse modo de particionamento de disco existe? Esse modo economiza o uso de alguns poucos kbytes de espaço em disco, e pode, em contrapartida gerar grandes problemas em uma nova instalação do sistema. O modo "perigosamente dedicado" em sua origem é um desejo antigo dos usuários do FreeBSD, especialmente durante a instalação, que é simplesmente poder ignorar a "geometria" de disco reconhecida pela BIOS e usar o disco todo de forma independente, sem prestar satisfação ao sistema básico de entrada/saída do PC.

O conceito de "Geometria" é ultrapassado, mas infelizmente ainda faz parte do coração da BIOS dos Computadores Pessoais, sendo extremamente necessário para a interação do computador com seus discos. Quando o FreeBSD cria as partições, ele tem que gravar sua localização de forma correspondente a maneira que a BIOS irá procurá-la. Se essa informação não é acessível à BIOS, pode ser que o sistema não consiga iniciar-se.

"O modo Dangerously dedicated" tenta evitar esse desconforto fazendo a operação se tornar mais simples. Em muitos casos essa forma de particionamento funciona,

mas ela foi criada para ser usada como última alternativa à necessidade de definir a geometria do disco - em 99% dos casos existem formas mais vantajosas de resolver problemas com geometria.

Mas então, como evitar a necessidade do modo "DD" na instalação do sistema? Comece anotando os valores pra geometria que a BIOS diz estar usando pros discos locais. Esses valores podem ser apresentados pelo kernel do FreeBSD, especificando a opção `-v` no prompt de `boot:` usando `boot -v` nas configurações do loader do sistema operacional. Antes do programa de instalação ser carregado o kernel apresenta a listagem dos valores para a Geometria de discos reconhecida pela BIOS do computador; não se precipite nem se preocupe, essas informações podem ser visualizadas paginando a tela para as notações anteriores, sendo possível assim verificar esses valores. Normalmente as unidades de disco da BIOS são apresentadas na mesma ordem que o FreeBSD as encontra, sendo primeiro as unidades IDE seguido das SCSI.

No momento do particionamento do disco, verifique se a geometria apresentada pelo FDISK corresponde ao valor apresentado pela BIOS; caso não corresponda use a tecla g para redefinir esses valores. Também pode ser necessário definir a geometria manualmente em casos onde o disco em questão está vazio e sem nenhum outro tipo de partição criada, ou se o disco foi instalado em um outro computador e foi colocado na estação atual recentemente. Note que esse tipo de complicação não é comum, e quando acontece, acontece apenas com o disco onde o FreeBSD está iniciando; Qualquer outro disco existente no computador será controlado perfeitamente pelo FreeBSD em qualquer situação.

Uma vez existindo a concordância de Geometria entre a BIOS e o FreeBSD, com certeza seus problemas terão acabado, e não existia a necessidade de usar o modo "DD". Contudo, se em casos extremos ainda ocorrerem erros de read error então pode cruzar os dedos e usar o modo dedicado (DD), afinal não há nada a perder, visto que das formas tradicionais sua BIOS insiste em não cooperar de forma correta.

Para voltar um disco particionado em modo "dangerously dedicated" para uso normal em um PC existem duas alternativas. A primeira é alocar dados nulos (NULL bytes) o bastante na MBR do disco, de forma que qualquer instalação posterior acredite que o disco está vazio. Isso pode ser feito, por exemplo, da seguinte maneira:

```
# dd if=/dev/zero of=/dev/rda0 count=15
```

E a segunda forma, a "opção" não documentada do DOS:

```
C:\> fdisk -/mbr
```

instalará um novo registro mestre de inicialização no disco em questão, sobrepondo inclusive o *bootstrap* do BSD.

P: Em quais partições podemos seguramente usar o softupdates? Eu tenho ouvido falar de problemas em se usar o softupdates na partição `/`?

R: A resposta breve: o softupdates pode ser usado com segurança em todas as partições.

A resposta mais completa: existiam algumas restrições quanto ao uso do softupdates na partição raiz do sistema. Tais restrições se deviam a duas características causadas pelo softupdates, que eram a pouquíssima chance de perder alguns dados da partição raiz se acontecesse algum system crash; apesar das probabilidades de perda de dados serem mínimas, elas existem; não existe uma danificação ou corrosão do sistema de arquivos, apenas alguns arquivos podem ser simplesmente perdidos. E a segunda característica é a diminuição temporária de espaço em disco.

Quando o softupdates está ativado o kernel pode levar até trinta segundos para realmente escrever alguns dados fisicamente em disco. Caso algum arquivo grande seja apagado, ela permanece temporariamente em disco até que o kernel realmente o apague, o que pode causar uma condição de corrida simples(simple race condition). Imagine, apenas para ilustrar, que você apague um arquivo enorme do disco, e logo em seguida crie um outro arquivo tão grande quanto o primeiro; a gravação de metadados no disco pode não ter sido realizada ainda quando o segundo arquivo é criado, ou seja o primeiro não foi realmente definido como um arquivo apagado. Nesse caso é provável que você receba uma mensagem dizendo que não existe espaço o bastante em disco para o segundo arquivo, quando você tem certeza absoluta que o espaço que acabou de ser liberado ao apagar o primeiro arquivo é o suficiente para alocar o segundo! Aí você tenta gravar o segundo arquivo mais uma vez, alguns meros segundos depois, e o processo de criação do mesmo, simplesmente funciona como esperado. Esse tipo de comportamento já levou mais de um usuário a balançar sua cabeça e duvidar de sua própria sanidade, ou mesmo da sanidade do sistema de arquivos do FreeBSD; em caso extremos, de dúvidas do bom estado de ambos, a sanidade do sistema de arquivos e do próprio usuário ;-)

Se o sistema falhar antes do kernel aceitar um conjunto de dados que tem que ser escrito no disco antes do mesmo ser gravado, é provável que exista perda dos dados em questão. Esse risco é extremamente baixo, e geralmente é contornável. O uso, por exemplo, das opções de cache de gravação das unidades de disco IDE é um fator que causa a possibilidade desse tipo de desconforto, portanto é altamente recomendável que essa opção seja desativada nos discos IDE quando for usar o softupdates.

Esse comportamento afeta todas as partições que estiverem com softupdates. Portanto, o que isso implica para a partição raiz?

A maioria das informações vitais da partição raiz mudam com pouquíssima freqüência. Arquivos como o `/kernel` e o conteúdo do `/etc` são alterados apenas durante a manutenção do sistema operacional, ou quando os usuários alteram suas senhas. Caso o sistema falhe durante essa janela de trinta segundos, depois que uma

dessas alterações foi feita, é possível que alguns dados sejam perdidos. Esse risco é baixíssimo e praticamente indiferente para maioria das aplicações, mas deve-se atentar que o risco existe. Caso seu sistema não tolere nem uma possibilidade tão baixa de riscos, não use o softupdates na partição raiz!

A `/` normalmente é uma das menores partições do sistema. Por padrão o FreeBSD coloca o diretório `/tmp` na partição `/`, comportamento este que pode ser modificado por administradores de sistemas mais experientes. Caso seu `/tmp` costume ocupar muito espaço, pode ser criado um link simbólico para o `/var/tmp`, o comportamento inverso também é válido e bastante seguro, caso o /tmp seja uma partição grande o bastante para alocar os dados do /var/tmp também.

P: O que é inapropriado no meu ccd?

R: Os sintomas desse tipo de dúvida:

```
# ccdconfig --C
ccdconfig: ioctl (CCDIOCSET): -/dev/ccd0c: Inappropriate file ↺
type or format
```

Esse problema normalmente ocorre quando se tenta concatenar as partições `c` cujo tipo padrão é `unused`. O ccd requer que a partição que ele esteja usando seja do tipo FS_BSDFFS. Edite o disklabel dos discos em questão e altere o tipo das partições para `4.2BSD`.

P: Porque eu não posso editar o disklabel do meu ccd?

R: Os sintomas desse tipo de dúvida:

```
# disklabel ccd0
(it prints something sensible here, so let us try to edit it)
# disklabel --e ccd0
(edit, save, quit)
disklabel: ioctl DIOCWDINFO: No disk label on disk;
use -"disklabel --r" to install initial label
```

Isso acontece por que o disklabel que o ccd retorna é na verdade um valor “falso”, que não está realmente gravado no disco. Esse problema pode ser resolvido ao reescrever explicitamente esse dado, da seguinte forma:

```
# disklabel ccd0 > -/tmp/disklabel.tmp
# disklabel --Rr ccd0 -/tmp/disklabel.tmp
# disklabel --e ccd0
(agora irá funcionar)
```

P: Posso montar outros tipos de partições externas sobre FreeBSD?

R: Digital UNIX

CDROMs do tipo UFS podem ser montados diretamente no FreeBSD, já a montagem de partições de disco do Digital UNIX que usam o UFS pode ser um

pouco mais complexa, dependendo dos detalhes do particionamento de disco para o sistema operacional em questão.

Linux
: A partir do 2.2, o FreeBSD suporta partições do tipo `ext2fs`. Veja o mount_ext2fs(8) para obter mais informações.

NT
: Existe suporte somente-leitura as partições NTFS no FreeBSD. Para obter mais informações leia esse tutorial, escrito por Mark Ovens em http://ukug.uk.FreeBSD.org/~mark/ntfs_install.html.

Mais informações sobre esse assunto seriam bem vindas :-)

P: Como montar uma partição DOS secundária?

R: As partições secundárias do DOS são encontradas depois que TODAS as partições primárias foram definidas. Por exemplo, se você tem a partição "E" como a segunda partição DOS no segundo disco SCSI, será necessário criar um arquivo especial para essa "quinta partição" (slice 5) no /dev, depois montar /dev/da1s5:

```
# cd -/dev
# sh MAKEDEV da1s5
# mount --t msdos -/dev/da1s5 -/dos/e
```

P: Existe um sistema de arquivos criptografado para o FreeBSD?

R: Sim, veja o `port` security/cfs.

P: Como posso usar o carregador do NT para iniciar o FreeBSD?

R: Esse procedimento é um pouco diferente entre o FreeBSD 2.2.X e o 3.X (que tem um sistema de inicío (*boot*) de 3 fases).

A idéia geral se consiste em copiar os primeiros setores da partição raiz nativa do FreeBSD e transforma-los em um arquivo, para ser colocado dentro da partição DOS/NT. Se assumir-mos que você vai chamar esse arquivo de `c:\bootsect.bsd` (inspirado por `c:\bootsect.dos`), pode-se então editar o arquivo `c:\boot.ini` de forma a carregá, mais ou menos assim:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows NT"
C:\BOOTSECT.BSD="FreeBSD"
C:\="DOS"
```

No 2.2.X esse procedimento assume que o DOS, o NT, o FreeBSD, ou o que quer que seja, tenha sido instalado em suas partições respectivas do fdisk no *mesmo* disco. Esse exemplo foi testado em um sistema onde o DOS & NT estavam instalados

na primeira partição do fdisk, e o FreeBSD na segunda. O FreeBSD havia sido configurado para iniciar a partir de sua partição nativa, e não pela MBR do disco.

Monte um disquete formatado em DOS (caso ele tenha sido convertido para NTFS) ou em FAT, por exemplo, sob o ponto de montagem `/mnt`.

```
# dd if=/dev/rda0a of=/mnt/bootsect.bsd bs=512 count=1
```

Reinicie o computador no DOS ou NT. Sobre o NTFS, copie o arquivo `bootsect.bsd` e/ou arquivo `bootsect.lnx` do disquete para `C:\`. Modifique os atributos originais(permissões) do `boot.ini` com:

```
C:\> attrib --s --r c:\boot.ini
```

Edite e adicione as entradas apropriadas no `boot.ini` seguindo o exemplo anterior e volte os atributos originais:

```
C:\> attrib +s +r c:\boot.ini
```

Se o FreeBSD estiver inicializando pela MBR, reconstrua-a com o comando `fdisk` do DOS depois de configurar os sistemas para iniciar a partir de suas partições nativas.

No FreeBSD 3.X esse procedimento é mais simples.

Se o FreeBSD estiver instalado no mesmo disco que a partição de inicialização (*boot*) do NT está instalada, copie o `/boot/boot1` para `C:\BOOTSECT.BSD`. Contundo, se o FreeBSD estiver em uma partição distinta, o `/boot/boot1` não irá funcionar, nesse caso, o `/boot/boot0` será necessário.

Atenção

NÃO COPIE SIMPLESMENTE O `/boot/boot0` NO LUGAR DO `/boot/boot1`, POIS A TABELA DE PARTIÇÃO SERÁ REESCRITA, E O COMPUTADOR SE TORNARÁ NÃO INICIALIZÁVEL!!

O `/boot/boot0` precisa ser instalado com o sysinstall, selecionando o gerenciador de inicialização (*boot*) do FreeBSD na tela que o programa pergunta se você deseja usar um gerenciador de inicialização(boot manager). Isso se deve ao fato que o `/boot/boot0` contém informações sobre a área da tabela de partições definidas como caracteres nulos, mas o sysinstall copia a tabela de partições antes de copiar o `/boot/boot0` para a MBR.

Quando o gerenciador de inicialização (*boot*) do FreeBSD é executado, ele grava qual último sistema operacional foi carregado, definindo uma flag de sistema ativo na tabela de partição referente aquele sistema, e depois ele escreve todos os 512 bytes

do próprio gerenciador de inicialização (*boot*) de volta na MBR, portanto se o `/boot/boot0` simplesmente for copiado para `C:\BOOTSECT.BSD` será definida uma tabela de partição vazia com a flag de partição ativa, na MBR.

P: Como posso iniciar o FreeBSD e o Linux com o LILO?

R: Caso o FreeBSD e o Linux estejam no mesmo disco, basta seguir as instruções de instalação do LILO para carregar sistemas operacionais não-Linux. De forma breve, tais instruções são

Carregue o Linux e adicione as seguintes linhas no `/etc/lilo.conf` :

```
other=/dev/hda2
 table=/dev/hda
 label=FreeBSD
```

(A definição acima assume que a sua partição FreeBSD é conhecida pelo Linux como `/dev/hda2` ; altere esse valor para sua necessidade). Depois basta executar o comando `lilo` como usuário `root` e deve estar pronto.

Caso o FreeBSD esteja em um outro disco, será preciso criar uma entrada `loader=/boot/chain.b` no LILO. Por exemplo:

```
other=/dev/dab4
        table=/dev/dab
        loader=/boot/chain.b
        label=FreeBSD
```

Em alguns casos é necessário especificar o número da unidade da BIOS para que o loader do FreeBSD consiga carregar o sistema com sucesso, a partir do segundo disco. Por exemplo, caso o disco SCSI com o FreeBSD seja reconhecido pela BIOS como o disco 1, na tela do carregadir de inicialização (*boot loader*) do FreeBSD será necessário definir:

```
Boot: 1:da(0,a)/kernel
```

No FreeBSD 2.2.5 e posteriores, o boot(8) pode ser configurado para que ele automaticamente defina essa informação na hora da inicialização (*boot*).

The Linux+FreeBSD mini-HOWTO é uma boa referência sobre a utilização do FreeBSD com o Linux.

P: Como eu inicio o FreeBSD e o Linux usando o BootEasy?

R: Instale o LILO no começo da sua partição de inicialização (*boot*) do Linux ao invés de instala-lo na MBR. Agora o LILO pode ser carregado a partir do BootEasy.

Com Windows 9x e Linux essa é uma ação recomendada sempre, para garantir que o Linux possa ser iniciado de forma mais fácil caso seja necessário reinstalar o

Windows (que é um sistema operacional que não espera nenhum outro sistema na MBR).

P: Como eu altero a tela de inicialização (*boot*) de `???` para algo mais significativo?

R: Você não pode alterar esse comportamento no gerenciador de inicialização (*boot*) padrão sem reescreve-lo. Existem inúmeros outros gerenciadores de inicialização (*boot*) na categoria `sysutils` da da coleção de `ports`, que oferecem esse recurso.

P: Possuo uma nova unidade de disco removível, como posso usa-la?

R: Ainda que seja uma unidade removível como um ZIP Drive, um EZ Drive (ou até um disquete, caso queira usa-lo dessa maneira), ou um novo disco, uma vez instalado e reconhecido pelo sistema e assim que o cartucho/disquete/outra-coisa esteja ligado a unidade em questão, as coisas passam a funcionar da mesma forma para qualquer tipo de dispositivo.

(essa seção é baseada no ZIP `FAQ` de Mark Mayo)

Caso o dispositivo seja um ZIP Drive ou um disquete cujo sistema de arquivos já tenha sido criado como sendo do tipo DOS, o seguinte comando é o bastante para montá-lo:

```
# mount --t msdos -/dev/fd0c -/floppy
```

caso seja um disquete, ou então:

```
# mount --t msdos -/dev/da2s4 -/zip
```

caso seja um disco ZIP com as configurações de fábrica.

Para outros tipos de disco, veja como eles são tratados, usando o fdisk(8) ou sysinstall(8).

Os próximos exemplos serão para um ZIP Drive controlado pela device da2, ou seja, correspondente ao terceiro disco SCSI.

A não ser que se trate de um disquete ou de um disco removível que se planeje compartilhar com outras pessoas, provavelmente é mais sensata a idéia de colocar um sistema de arquivos do tipo BSD nesse disco. Um sistema de arquivos desse tipo possibilita usar o suporte a nomes longos de arquivos, ter uma performance de, pelo menos 2x, e oferece muito mais estabilidade e confiança. O primeiro passo é redefinir a partição DOS da unidade removível de forma que a mesma passe a ser do tipo BSD; o fdisk(8) ou `/stand/sysinstall` podem ser usados para esse fim, em um disco pequeno onde não se queira preocupações quanto a possibilidade de manter suporte a múltiplos sistemas operacionais. Nesse caso simplesmente elimine toda a partição FAT do disco e use o particionador do BSD:

```
# dd if=/dev/zero of=/dev/rda2 count=2
```

```
# disklabel --Brw da2 auto
```

O disklabel ou `/stand/sysinstall` podem ser usados para criar múltiplas partições do tipo BSD. Certamente o usuário vai querer criar mais de uma partição BSD se a intenção for criar espaço de Swap. De qualquer forma, esse tipo de ação é irrelevante em um disco removível do tipo ZIP.

Finalmente, crie um novo sistema de arquivos como esse, que usa todo o disco em um ZIP Drive:

```
# newfs -/dev/rda2c
```

e monte-o:

```
# mount -/dev/da2c -/zip
```

provavelmente é uma boa idéia adicionar uma entrada no `/etc/fstab` para facilitar o processo de montagem dessa unidade(veja fstab(5))de forma que seja apenas necessário digitar o comando `mount /zip` no futuro:

```
/dev/da2c -/zip ffs rw,noauto 0 0
```

P: Por que eu tive o erro Incorrect super block na hora de montar um CDROM?

R: É necessário avisar ao mount(8) que tipo de unidade está sendo montada. Essa definição é tratada com mais detalhes em na seção de mídias ópticas do Manual do FreeBSD , mais precisamente na seção Usando CDs de Dados.

P: Device not configured ocorre na hora de montar o CDROM?

R: Geralmente esse tipo de comportamento indica que não existe nenhum CD na unidade de CDROM, ou então, que o CD em questão não é visível ao barramento de dados do seu PC, comum quando um CD-RW não pode ser lido por um drive tradicional. Por gentileza, queira referir-se a Usando CDs de Dados da seção do Manual do FreeBSD para uma discussão mais detalhada sobre esse assunto.

P: Porque todos caracteres não-Inglês são apresentados como “?” no sistema de arquivos do CD que acabou de ser montado no FreeBSD?

R: Provavelmente seu CDROM usa extensões “Joliet” para armazenar informações sobre tipos de arquivos e diretórios. Esse assunto é discutido no capítulo de Criação e Uso de CDROMs do Manual do FreeBSD, mais precisamente na seção de Uso de CDs de Dados.

P: Eu queimei um CD no FreeBSD e agora esse disco não pode ser lido em nenhum outro sistema operacional. Por que?

R: Provavelmente você queimou um CD de forma crua (usando raw mode) no seu sistema, ao invés de criar um sistema de arquivos do tipo ISO 9660. Dê uma olhada

no Capítulo de Criação de CDROMs do Manual do FreeBSD, em específico na seção criando CDS de dados puros.

P: Como posso criar uma imagem de um CD de dados?

R: A criação de imagens de CDs de dados é discutida na seção duplicando CDs de dados do Manual do FreeBSD. Para mais informações sobre unidades ópticas, por gentileza, queira referir-se a seção Criação de CDs no capítulo de Armazenamento do Manual do FreeBSD.

P: Por que um CD de áudio não pode ser montado com o comando `mount`?

R: Ao tentar montar um CD de áudio, o mais provável é que ocorra um erro do tipo cd9660: /dev/acd0c: Invalid argument. Isso se deve ao fato que o comando `mount` trabalha exclusivamente com sistema de arquivos, o que não é o caso de CDs de áudio. CDs de áudio contém apenas dados, e por isso é necessário alguma aplicação capaz de ler tais CDs. Use algum programa como o audio/xmcdna coleção de `Ports` do FreeBSD para ler dados em CDs desse tipo.

P: Como eu monto um CD com múltiplas sessões de gravação?

R: Por padrão, o mount(8) tenta montar a última trilha de dados (sessão) de um CD. Caso queira montar uma sessão anterior pode-se usar o argumento `-s` na linha de comando em questão. Por gentileza, refira-se a página de manual do comando mount_cd9660(8) para obter exemplos específicos do uso dessa e de outras opções.

P: Como posso permitir que usuários comuns montem disquetes, CDROMs e outros tipos de mídia removível?

R: Usuários comuns podem ter a permissão de montar dispositivos. Veja como:

1. Como `root` defina a variável `vfs.usermount` do sysctl para `1`.

```
# sysctl --w vfs.usermount=1
```

2. Como `root` defina as permissões apropriadas nos dispositivos associados ao controle da mídia removível.

   Por exemplo, para permitir que os usuários possam montar a primeira unidade de disquete, use:

```
# chmod 666 -/dev/fd0
```

   Para permitir que os usuários do grupo `operator` montem a unidade de CDROM, use:

```
# chgrp operator -/dev/cd0c
# chmod 640 -/dev/cd0c
```

3. Finalmente, para tornar essa alteração permanente, adicione a linha `vfs.usermount=1` no arquivo `/etc/sysctl.conf` .

Agora todos os usuários podem montar qualquer disquete no `/dev/fd0` em algum ponto de montagem que lhes pertença:

```
% mkdir ~/my-mount-point
% mount --t msdos -/dev/fd0 ~/my-mount-point
```

Os usuários no grupo `operator` agora tem permissão para montar os CDs no `/dev/cd0c` em qualquer ponto de montagem que lhes pertença:

```
% mkdir ~/my-mount-point
% mount --t msdos -/dev/cd0c ~/my-mount-point
```

Desmontar o dispositivo é extremamente simples:

```
% umount ~/my-mount-point
```

Contudo, habilitar a opção `vfs.usermount` , contudo, causa algumas implicâncias quanto ao quesito segurança;. A forma mais racional de acessar mídia do tipo MSDOS no FreeBSD é usando a aplicação mtools , disponível na Coleção de `Ports`.

P: Os comandos `du` e `df` apresentam quantias distintas de espaço em disco disponível. O que está havendo?

R: É necessário entender o que os comandos `du` e `df` realmente fazem. O `du` percorre a árvore de diretórios medindo o tamanho de cada arquivo e apresentando o total da soma de todos os arquivos encontrados em um dado diretório, e posteriormente apresentando a soma de ambos subdiretórios no diretório de nível seguinte, `df` simplesmente pergunta ao sistema de arquivos quanto espaço ainda lhe resta. Parece o mesmo comportamento, mas um arquivo alocado fora de um diretório, por exemplo, afeta os dados apresentados pelo `df`, mas não afeta o `du`.

Quando um programa está usando algum arquivo, e este é apagado, o arquivo não é retirado do sistema de arquivos até que o programa em questão pare de usá-lo. Contudo, esse arquivo é imediatamente deletado da listagem de diretórios; isso pode ser mais bem observado com algum paginador como o `more`. Por exemplo, assuma que existe um arquivo grande o bastante que sua presença seja perceptível claramente na saída dos comandos `du` e `df`. (Considerando que hoje em dia os discos são bem grandes, o arquivo em questão deve ser um arquivo de tamanho *extremamente considerável!*) Caso o arquivo seja apagado em quanto ele esteja sendo paginado com o `more`, pode-se perceber que o `more` não passa a reclamar que o arquivo não pode mais ser visualizado no mesmo instante. Contudo, a listagem, do diretório em questão não apresentará mais esse arquivo, de forma que nenhum usuário ou outro programa possa acessá-lo. A entrada do arquivo no diretório é simplesmente removida. O `du` vai mostrar que esse arquivo não existe mais — -- afinal, o du percorreu todo o diretório e não encontrou esse arquivo listado -- contudo o `df` mostra que o arquivo continua no disco, pois o sistema de arquivos sabe que o `more` continua usando aquele espaço de dados que se encontram no disco. Uma vez terminada a sessão do `more`,`du` e `df` concordarão entre si.

Note que em sistemas de arquivos com softupdates, a liberação de espaço em disco também pode ser atrasada em até 30 segundos dependendo da situação; apenas depois desse tempo, a alteração em disco será visível!

Esse comportamento é comum e fácil de ser observado em Servidores Web. Muitos usuários configuram algum Servidor Web no FreeBSD e se esquecem de rotacionar os arquivos de log da aplicação, os quais entopem o `/var`. O novo administrador do sistema deleta os arquivos de log em questão, mas o sistema operacional continua reclamando que a partição está cheia, até que o Servidor Web seja desligado e religado, de forma que a aplicação libera o arquivo e permite que o sistema apague o arquivo, recuperando o espaço em disco em questão. Para prevenir que isso ocorra, configure o newsyslog(8).

P: Como eu posso adicionar mais espaço para *swap*?

R: No capítulo Configuração e Ajuste Fino (*Tuning*) do Manual do FreeBSD, pode ser encontrada uma seção descrevendo como fazer isto.

Capítulo 10. Administração do Sistema

P: Onde estão os arquivos que configuram a inicialização do sistema ?

R: Do FreeBSD 2.0.5R até o 2.2.1R, o arquivo de configurações primário é o `/etc/sysconfig`. Todas as opções devem ser definidas nesse arquivo ou então em outros, como o `/etc/rc` (veja o manual para o rc(8)) e o `/etc/netstart`

Dê uma olhada no `/etc/sysconfig` e altere as variáveis de acordo com o que você quer configurar no seu sistema. O arquivo é repleto de comentários que auxiliam a correta definição dos valores a serem definidos.

A partir do 2.2.1 até o 3.0, o `/etc/sysconfig` foi renomeado para rc.conf(5), que é auto-descritivo, e cuja sintaxe foi melhorada no processo de substituição. O `/etc/netstart` agora se chama `/etc/rc.network`, de forma que todos os arquivos possam ser copiados com um simples comando como um `cp /usr/src/etc/rc* /etc`

E depois, a partir do FreeBSD 3.1, o `/etc/rc.conf` foi alterado para o `/etc/defaults/rc.conf`. *Não edite esse arquivo!* Ao invés disso, para todas as entradas que você queira alterar no `/etc/defaults/rc.conf`, basta apenas copiar a linha relativa à essa entrada para o `/etc/rc.conf` e depois modificar seu valor.

Por exemplo, caso deseje iniciar o named, o servidor DNS disponível no FreeBSD, a partir do FreeBSD 3.1 basta fazer isso:

```
# echo named_enable="YES" >> -/etc/rc.conf
```

Para iniciar serviços locais no FreeBSD 3.1 e posteriores, basta colocar os scripts shell de inicialização desses serviços no diretório `/usr/local/etc/rc.d`. Tais shell scripts devem ser executáveis e terminarem com a extensão .sh. No FreeBSD 3.0 ou anteriores, o arquivo `/etc/rc.local` era a única opção para iniciar serviços/processos locais automaticamente.

O arquivo `/etc/rc.serial` é usado para a inicialização de portas seriais (por exemplo, para definir as características das portas, e assim por diante).

O arquivo `/etc/rc.i386` é usado para configurações específicas de sistemas Intel e compatíveis, como por exemplo, emulação iBCS2 ou definições do sistema de console dos PC.

P: Como posso adicionar um usuário de forma simples?

R: Use o comando adduser(8). Caso prefira uma forma mais complexa (e mais completa), use o comando pw(8).

Para remover o usuário do sistema, use o comando rmuser(8). Mais uma vez, o pw(8) também funciona muito bem nesse caso.

P: Depois de editar o crontab, mensagens como root: not found ficam aparecendo sempre. Por que?

R: Normalmente esse é um problema causado ao se editar o crontab do sistema (`/etc/crontab`) e depois usar o crontab(1) para instala-lo:

```
# crontab -/etc/crontab
```

Essa não é a forma correta de fazer as coisas. O crontab do sistema tem um formato distinto do crontab dos usuários, o qual o crontab(1) atualiza (o manual do crontab(5) explica tais diferenças de forma mais detalhada).

Caso você tenha cometido esse engano, o novo crontab é uma simples cópia do `/etc/crontab` , ou seja, com um formato errado. Apague-o com o comando:

```
# crontab --r
```

Da próxima vez que editar o `/etc/crontab` , nenhuma ação precisa ser tomada para avisar o cron(8) das alterações. Ele vai perceber as mudanças automaticamente.

Caso queira executar alguma tarefa diária, semanal ou mensal, é mais indicado adicionar alguns scripts de shell sob o `/usr/local/etc/periodic` e deixar o programa periodic(8), chamado a partir da tabela cron do sistema, cuidar das suas tarefas assim como ele faz com as outras tarefas pertinentes ao sistema.

A única razão para esse erro é que a tabela de cron do sistema tem um campo a mais, que especifica o usuário que deve executar o comando. No crontab do sistema padrão do FreeBSD, esse usuário é o `root`, em todas as entradas. Quando essa crontab é usada como a tabela de cron do `root` (que é diferente da tabela de cron do sistema), o cron(8) assume que a string `root` fosse um primeiro comando, mas esse comando não existe, por isso ocorre o erro.

P: Porque o erro you are not in the correct group to su root ocorre, quando eu tento virar `root` com o su ?

R: Essa é uma característica de segurança do FreeBSD. Para se tornar `root` com o su (ou qualquer outro usuário com privilégios de super usuário), é preciso fazer parte do grupo `wheel`. Sem essa característica, qualquer usuário com uma conta válida no sistema que soubesse a senha de `root` poderia obter privilégios de super usuário. Por causa do comportamento atual, essa afirmação não é verdadeira, uma vez que o su não vai nem permitir que o usuário dê a senha de `root`, caso ele não esteja no grupo `wheel`.

Para permitir que algum usuário se torne `root`, basta que ele faça parte do grupo `wheel`.

P: Cometi um erro no `rc.conf`, ou em algum outro arquivo de inicialização, e agora não posso corrigir essa alteração porque o sistema de arquivos é apenas-leitura. O que devo fazer?

R: Nessa situação, o comportamento esperado é que o sistema entre em modo monousuário e peça o caminho completo para o seu interpretador de comandos (sua shell). Basta confirmar a shell padrão, que ele oferece, com um simples `ENTER`, e depois executar um `mount /` para remontar o sistema de arquivos raiz (`/`) em modo leitura/escrita (rw). Também pode ser necessário executar um `mount -a -t ufs` para montar o sistema de arquivos onde o seu editor de texto preferido vai estar disponível. Caso seu editor esteja em um sistema de arquivos da rede, será necessário configurar a rede manualmente, ou usar um editor disponível localmente, como o ed(1).

Caso queira usar um editor de tela inteira como o vi(1) ou emacs(1), será necessário definir a variável de ambiente TERM como do tipo cons25, bastando um simples export TERM=cons25, de forma que tais editores possam carregar as informações corretas da base de dados do termcap(5).

Depois disso, o `/etc/rc.conf` pode ser editado normalmente, e a sintaxe problemática, corrigida. A mensagem de erro apresentada imediatamente após o carregamento do *kernel* indica o número da linha e o arquivo onde o erro aconteceu.

P: Porque estou tendo problemas ao configurar minha impressora?

R: Por gentileza, dê uma olhada nas páginas sobre impressão do Manual do FreeBSD. O documento deve responder a maioria de suas dúvidas. Veja a entrada sobre Impressão no Manual do FreeBSD.

Algumas impressoras precisam de um driver local, baseado em estações, para prover qualquer tipo de impressão. Essas impressoras são chamadas de “WinPrinters” e não são suportadas nativamente pelo FreeBSD. Se sua impressora não funciona sob DOS ou com Windows NT 4.0, provavelmente ela é uma WinPrinter. A única esperança de se obter uma impressora desse tipo funcionando, é verificar se o `port` print/pnm2ppa tem suporte para ela.

P: Como posso corrigir o mapeamento de teclados do meu sistema?

R: Por gentileza, refira-se à seção usando localização do Manual do FreeBSD, mais precisamente na parte sobre a configuração do console.

P: O que causa mensagens como: unknown: <PNP0303> can't assign resources na inicialização do sistema?

R: O trecho a seguir é citação de uma mensagem enviada na lista freebsd-current.

> A mensagem “can't assign resources” indica que os equipamentos em questão são do tipo ISA, e que não existem

> entradas indicando drivers não-PnP compiladas no *kernel*. Esses equipamentos podem ser controladoras de teclados, controladora de interrupção programável e várias outras peças da infra-estrutura padrão do sistema. Os recursos não podem ser atribuídos por já existirem drivers usando tais endereços.
>
> —Garrett Wollman, 24 Abril 2001

P: Porque eu não consigo fazer as quotas de usuários funcionarem de forma correta?

R: 1. Não habilite quotas na `/`,

2. Coloque o arquivo de quotas indicando o sistema de arquivos onde se deseja estabelecer as quotas, por exemplo:

Sistemas de arquivos	Arquivos de quotas
`/usr`	`/usr/admin/quotas`
`/home`	`/home/admin/quotas`
...	...

P: O FreeBSD suporta as primitivas de IPC do System V?

R: Sim, o FreeBSD suporta IPC ao estilo do System V. Esse suporte inclui compartilhamento de memória, mensagens e semáforos. É necessário adicionar as seguintes linhas no arquivo de configurações do seu *kernel*, para ativar o suporte:

```
options    SYSVSHM   # habilita memória compartilhada
options    SYSVSEM          # habilita semáforos
options    SYSVMSG          # habilita mensagens
```

Nota

No FreeBSD 3.2 e posteriores, tais opções já fazem parte do *kernel GENERIC*, o que significa que tal suporte já deve estar compilado no seu sistema.

Recompile e instale o novo *kernel*.

P: Como posso usar o sendmail para entregar mensagens com UUCP?

R: A configuração do sendmail disponível por padrão no FreeBSD é direcionada para sites que estejam conectados à Internet. Servidores que pretendem entregar suas mensagens via UUCP devem instalar um novo arquivo de configurações do sendmail.

Alterar o `/etc/mail/sendmail.cf` manualmente é considerado tarefa para os mais puristas. A versão 8 do sendmail tem uma nova abordagem de arquivos de configuração por meio de pré processamento com o m4(1), onde os modelos de configuração são manipulados em um nível mais alto de abstração. Use os arquivos de configuração disponíveis sob /usr/src/usr.sbin/sendmail/cf.

Caso seu sistema não tenha sido instalado com os fontes, os arquivos de configuração do sendmail foram divididos em pacotes separados. Assumindo que você tenha o CDROM do FreeBSD montado, faça o seguinte:

```
# cd -/cdrom/src
# cat scontrib.?? -| tar xzf -- --C -/usr/src contrib/sendmail
```

Não se desespere, são apenas algumas centenas de Kilobytes em tamanho. O arquivo README no diretório cf serve de introdução básica ao uso do m4.

Para entregar mensagens via UUCP, o melhor conselho é usar o `mailtertable` . Trata-se de uma base de dados que o sendmail usa para basear suas decisões de roteamento de mensagens.

Primeiro, é necessário criar seu arquivo `.mc`. O diretório `/usr/src/usr.sbin/sendmail/cf/cf` é o diretório home para esse tipo de arquivo. Dê uma olhada, já existem alguns exemplos disponíveis por lá. Se assumirmos que você chamou o arquivo de `foo.mc` , para converte-lo para um arquivo `sendmail.cf` válido basta:

```
# cd -/usr/src/usr.sbin/sendmail/cf/cf
# make foo.cf
# cp foo.cf -/etc/mail/sendmail.cf
```

Um arquivo `.mc` típico, se parece com algo mais ou menos assim:

```
VERSIONID(`Número da sua versão ')
OSTYPE(bsd4.4)

FEATURE(accept_unresolvable_domains)
FEATURE(nocanonify)
FEATURE(mailertable, `hash --o -/etc/mail/mailertable')

define(`UUCP_RELAY', your.uucp.relay )
define(`UUCP_MAX_SIZE', 200000)
define(`confDONT_PROBE_INTERFACES')

MAILER(local)
MAILER(smtp)
MAILER(uucp)

Cw    your.alias.host.name
Cw    youruucpnodename.UUCP
```

As linhas contendo as entradas `accept_unresolvable_domains` , `nocanonify` , e `confDONT_PROBE_INTERFACES` previnem o uso do DNS durante a entrega das mensagens. A cláusula `UUCP_RELAY` é necessária por razões bizarras, nem pergunte quais. Apenas coloque o nome de uma estação que possa manipular endereços com pseudo-domínio .UUCP; normalmente o endereço de relay de e-mail do seu Provedor de Serviço Internet deve servir.

Depois disso, é necessário usar o arquivo `/etc/mail/mailertable` . Caso exista apenas um link para fora, por onde todos os e-mails são roteados, as seguintes definições são o bastante:

```
#
# makemap hash -/etc/mail/mailertable.db < -/etc/mail/mailertable
.  uucp-dom:your.uucp.relay
```

Um exemplo mais complexo, se pareceria com:

```
#
# 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:
```

Como pode-se perceber, se trata de um arquivo usado na vida real. As primeiras três linhas tratam situações especiais onde as mensagens endereçadas aquele domínio não devem ser roteadas pela saída padrão, mas ao invés disso, ser entregues para algum servidor UUCP vizinho, de forma a encurtar o caminho para entrega dos e-mails. A linha seguinte trata mensagens para rede Ethernet local, para domínios onde os mails possam ser entregues via SMTP. Finalmente, os vizinhos UUCP são mencionados na notação do pseudo-domínio .UUCP, que permite um `uucp-neighbor!recipient` sobrescrever as regras padrão. A última linha é sempre um ponto, que indica que todos os e-mails que não foram tratados pelas entradas anteriores cuja entrega seja do tipo UUCP, devem ser tratados por um dos vizinhos UUCP que sirva como gateway universal com o resto do mundo. Todas as estações antecedendo a entrada `uucp-dom:` devem ser nomes de vizinhos UUCP válidos, que podem ser checados com o comando `uuname`.

Para lembrar que esse arquivo precisa ser convertido em base de dados do tipo DBM, o comando necessário para tomar essa ação está comentado no início do arquivo mailertable. Esse comando deve ser executado sempre que o mailertable for alterado.

Dica final: caso tenha dúvidas se uma rota de e-mail em particular irá funcionar, lembre-se que a opção `-bt` do sendmail permite que ele seja iniciado em modo de testes de endereço; simplesmente digite `3,0` seguido do endereço que você quer testar o roteamento de mensagens. A última linha irá indicar o agente de transferência interno que foi usado, a estação de destino com a qual esse agente de entrega irá se comunicar, e o seu endereço. Para sair desse modo, digite Control-D.

```
% sendmail --bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 foo@example.com
canonify           input: foo @ example -. com
...
parse            returns: $# uucp-dom $@ your.uucp.relay  $: foo ↵
< @ example -. com -. >
> ^D
```

P: Como eu configuro e-mail em uma conexão dialup com a rede?

R: Se a sua conexão discada lhe atribui um endereço IP estático, não é necessário configurar nenhuma opção extra. Ajuste o nome da sua estação para o nome que a identifica na Internet, e o sendmail fará o resto.

Mas se a conexão PPP lhe atribui endereços dinâmicos, provavelmente o seu Provedor de Serviço Internet oferece uma conta de correio eletrônico em seus servidores. Vamos assumir que o nome do domínio do seu provedor é `example.net` , e que o nome do seu usuário é `user`. Vamos assumir também que o nome da sua estação seja `bsd.home` e que o Provedor de Serviço Internet defina que o endereço `relay.example.net` deva ser usado para relay de mensagens eletrônicas.

Para acessar as mensagens da sua caixa de correio, é necessário usar um agente de busca. O Fetchmail é uma boa escolha, já que ele suporta vários protocolos distintos. Normalmente o provedor em questão oferece serviço de POP3. Caso sua conexão PPP seja estabelecida à nível de usuário (user-PPP), para acessar suas mensagens automaticamente ao estabelecer-se uma conexão com a rede, basta adicionar a seguinte entrada no arquivo `/etc/ppp/ppp/linkup` :

```
MYADDR:
  -!bg su user --c fetchmail
```

Caso esteja usando o sendmail (como foi descrito anteriormente) para entregar suas mensagens para endereços não-locais, insira o comando:

```
  -!bg su user --c -"sendmail --q"
```

depois da entrada apresentada anteriormente. Esse comando irá forçar o sendmail a processar sua fila de e-mail tão logo uma conexão com a rede seja estabelecida.

Assumindo que exista uma conta para o `user` na máquina `bsd.home`. No diretório home do `user` na estação `bsd.home`, crie um arquivo `.fetchmailrc` com o seguinte conteúdo:

```
poll example.net protocol pop3 fetchall pass MySecret
```

Esse arquivo não deve ter permissão de leitura para nenhum outro usuário, a não ser o `user` já que ele contém a `sua senha`.

Para garantir que o cabeçalho `from:` esteja sempre correto, é necessário indicar ao sendmail que o endereço `user@example.net` deve ser usado ao invés de `user@bsd.home`. Também é interessante configurar o sendmail para entregar suas mensagens via `relay.example.net`, permitindo transmissão de mensagens de forma mais rápida.

O seguinte arquivo `.mc` deve ser o bastante:

```
VERSIONID(`bsd.home.mc version 1.0')
OSTYPE(bsd4.4)dnl
FEATURE(nouucp)dnl
MAILER(local)dnl
MAILER(smtp)dnl
Cwlocalhost
Cwbsd.home
MASQUERADE_AS(`example.net')dnl
FEATURE(allmasquerade)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(nocanonify)dnl
FEATURE(nodns)dnl
define(`SMART_HOST', `relay.example.net')
Dmbsd.home
define(`confDOMAIN_NAME',`bsd.home')dnl
define(`confDELIVERY_MODE',`deferred')dnl
```

Por gentileza, refira-se à seção anterior para obter detalhes sobre como transformar esse arquivo `.mc` em um arquivo `sendmail.cf`. Não se esqueça também de reiniciar o sendmail depois de alterar o `sendmail.cf`.

P: Que outros servidores de correio eletrônico posso usar no lugar do Sendmail?

R: O Sendmail é o programa servidor de correio eletrônico padrão no FreeBSD, mas ele pode ser facilmente substituído por qualquer outro MTA (por instância, um MTA instalado a partir do `ports`).

Existem vários MTA's que servem de alternativa ao Sendmail na Coleção de `Ports` do FreeBSD, sendo o mail/exim, mail/postfix, mail/qmail, mail/zmailer, os mais populares.

A diversidade é sempre uma boa indicação, e o fato de ter vários servidores de e-mail disponíveis é ótimo. Conteúdo, evite perguntas como “O Sendmail é melhor

que o Qmail?" nas listas de discussão. Se você realmente quer saber, procure no histórico das listas. As vantagens e desvantagens de cada MTA já foram discutidas inúmeras vezes.

P: Esqueci a senha de root! O que eu faço?

R: Em primeiro lugar, não entre em pânico! Reinicie o seu FreeBSD, digite **`boot -s`** na tela do Boot: (ou apenas **`-s`** para as versões anteriores à 3.2 do FreeBSD) para entrar e modo monousuário. Quando o sistema perguntar sobre que shell usar, aperte ENTER. Você estará em uma prompt de comandos; digite `mount -u /` para montar o sistema de arquivos raiz com leitura/escrita, e depois `mount -a` para remontar todos os seus sistemas de arquivos. Execute o comando `passwd root` para modificar a senha de root do sistema, e depois digite exit(1) para continuar a inicialização em modo multiusuário.

P: Como posso evitar que a seqüência de teclas Control+Alt+Delete reinicie o sistema?

R: Caso esteja usando o syscons (o driver padrão para o console) em um sistema FreeBSD 2.2.7 ou posterior, construa e instale um novo *kernel* com a opção:

```
options SC_DISABLE_REBOOT
```

Caso use o driver de console PCVT em um FreeBSD 2.2.5 ou posterior, use a seguinte linha:

```
options PCVT_CTRL_ALT_DEL
```

Em versões anteriores às citadas, edite o mapeamento do seu teclado, usado para o console, e substitua a palavra `boot` por `nop`. O mapeamento de teclado padrão está em `/usr/share/syscons/keymaps/us.iso.kbd` . O `/etc/rc.conf` deve ser instruído de forma que esse arquivo seja lido. Se você estiver usando um outro mapa específico para o seu país, edite esse mapa ao invés do padrão.

P: Como posso converter arquivos de texto do DOS para o formato do Unix?

R: Use esse comando do perl:

```
% perl --i.bak --npe -'s/\r\n/\n/g' file...
```

onde `file` indica o arquivo ou arquivos a serem processados. As modificações são feitas no próprio arquivo e o original é salvo com a extensão .bak.

O comando tr(1) também pode ser usado:

```
% tr --d -'\r' < dos-text-file > unix-file
```

Onde *`dos-text-file`* é o arquivo com o texto em formato DOS, enquanto o *`unix-file`* armazenará a saída convertida. Usar o tr(1) é um pouco mais rápido do que usar o perl.

P: Como eu mato processos pelo seu nome?

R: Use o comando killall(1).

P: Por que motivos o su está me atazanando pelo fato de não pertencer à ACL do `root`?

R: Esse erro é proveniente do sistema de autenticação da distrição do Kerberos. O problema não é uma perturbação fatal. Basta executar o su com a opção -K ou então desinstalar o Kerberos, como será descrito na próxima questão.

P: Como eu desinstalo o Kerberos?

R: Para remover o Kerberos do sistema, reinstale a distribuição `bin` da versão que está sendo usada. Caso tenha o CDROM do FreeBSD, monte-o (vamos assumir, em /cdrom) e execute os comandos:

```
# cd -/cdrom/bin
# ./install.sh
```

Ou então, apague todas as opções "MAKE_KERBEROS" do `/etc/make.conf` e recompile todo o sistema com um build world.

P: Como posso adicionar pseudo-terminais ao sistema?

R: Caso tenha inúmeras conexões telnet, ssh, X, ou tela de usuário, é provável que você atingirá o limite dos seus pseudo-terminais. Aqui estão as instruções de como adicionar mais pseudo-terminais:

1. Construa e instale um novo *kernel* com a linha

   ```
   pseudo-device pty 256
   ```

   em seu arquivo de configurações.

2. Execute os comandos

   ```
   # cd -/dev
   # sh MAKEDEV pty{1,2,3,4,5,6,7}
   ```

   de forma a criar 256 novos devices para os novos terminais.

3. Edite o `/etc/ttys` e adicione uma linha para cada um dos 256 terminais. Tais entradas devem ter o formato correspondente às entradas já existentes, por exemplo:

   ```
   ttyqc none network
   ```

   A ordem de definição das letras é expressa como `tty[pqrsPQRS][0-9a-v]` , ao ilustrarmos em expressões regulares.

4. Reinicie o sistema com o novo *kernel*, e pronto.

P: Por que motivo não consigo criar a device snd0?

R: Simples, porque não existe a device `snd`. Esse nome é usado para identificar o conjunto de devices que compõem os drivers de som do FreeBSD, como as devices `mixer`, `sequencer` , e `dsp`.

Para criar tais devices, basta executar:

```
# cd -/dev
# sh MAKEDEV snd0
```

P: Como posso reler o `/etc/rc.conf` e reiniciar o `/etc/rc` sem rebootar o sistema?

R: Vá para o modo monousuário e volte para o modo multiusuário.

É simples; no console, faça:

```
# shutdown now
(Note: without --r or --h)

# return
# exit
```

P: O que é uma sandbox?

R: "Sandbox" é um jargão usado em discussões pertinentes à segurança de sistemas. Pode significar duas coisas:

- Um processo enquadrado em um conjunto de paredes virtuais que são criadas para prevenir que algum usuário, ao explorar alguma inconformidade do processo, possa também explorar e obter privilégios no sistema operacional como um todo.

  O processo deve conseguir "rodar" dentro dessas paredes, ou seja, nada que o processo possa fazer ao executar seu código, pode ser capaz de violar tais paredes. Dessa forma não é necessária uma auditoria detalhada do código e das ações do processo para que se possa realizar algumas afirmações pertinentes à segurança de tal sistema.

  Tais paredes podem ser a identificação de um usuário (userid), por exemplo. Essa é a definição de sandbox usada nas páginas de manuais do named e de security.

  Observe o serviço `ntalk`, como exemplo (veja o /etc/inetd.conf). Esse serviço costumava ser executado com userid do `root`. Hoje em dia o processo roda com o userid do `tty`. O usuário `tty`, portanto, é uma sandbox criada para dificultar qualquer atividade de um usuário malicioso que por ventura consiga acesso ao sistema por meio do ntalk. Com essa sandbox, uma violação de segurança bem sucedida via `ntalk` dificultaria qualquer ação tomada além das possíveis com o userid do `tty`.

- Um processo criado dentro de um ambiente de simulação. Essa é uma situação mais complexa. Basicamente implica que qualquer pessoa má intencionada que consiga explorar tal processo, acreditará que pode obter acesso à todo o ambiente, nas na verdade, estará apenas acessando um sistema de simulação, não alterando nenhum dado real.

  A forma mais comum de conseguir criar um ambiente simulado como esse, é criando um subdiretório à partir de onde o processo consiga acessar (uma cópia de) qualquer arquivo do sistema que por ventura ele precise, e executar esse processo simulando um diretório raiz (ou seja, para o processo, o `/` será o subdiretório determinado, e não o verdadeiro `/` do sistema).

  Outra situação comum é montar um sistema de arquivos base com apenas permissão de leitura, e depois criar um outro sistema de arquivos em uma camada superior, com acesso de escrita/leitura, dando ao processo a impressão de poder ler/escrever em todo o sistema de arquivos. Apenas o processo em questão percebe esse ambiente, enquanto os outros não são necessariamente ludibriados.

  A intenção é que tais sandbox sejam tão transparentes que qualquer usuário (ou hacker) não consiga perceber que está dentro de uma.

Os sistemas Unix costumam implementar esses dois principais tipos de sandbox, um em nível de processo e o outro, muito comum, em nível de userid.

Cada processo Unix é completamente separado dos outros, por meio de algum tipo de parede de segurança. Um processo nunca modifica o espaço de endereçamento de outro, diferente do ambiente Windows onde cada processo pode facilmente sobrescrever endereços de outros processos, fazendo o sistema travar.

Cada processo Unix é de propriedade de um userid em particular. Caso o userid não seja do `root`, ele serve de parede de segurança em relação aos processos pertencentes a outros usuários. Os userid também são usados para proteger dados armazenados em disco.

P: O que é `securelevel` (nível de segurança do sistema)?

R: `securelevel` (nível de segurança do sistema) é um mecanismo de segurança implementado no *kernel* do FreeBSD. Basicamente, quando o `securelevel` é positivo, o *kernel* restringe algumas tarefas do sistema; nem mesmo o superusuário (por exemplo, o `root`) tem permissão de realizar tais tarefas. Na data que este `FAQ` foi escrito, o mecanismo de `securelevel` do FreeBSD era capaz de, entre outras coisas, limitar as habilidades de:

- retirar algumas flags de arquivos, como a `schg` (flag de imutabilidade do sistema),

- escrever na memória do *kernel* por meio do `/dev/mem` e `/dev/kmem`,

- carregar módulos do *kernel*, e
- alterar regras de Firewall do ipfirewall(4).

Para verificar o estado do `securelevel` (nível de segurança do sistema) em um sistema em funcionando, simplesmente execute o seguinte comando:

```
# sysctl kern.securelevel
```

A saída apresentará o nome da variável do sysctl(8) (nesse caso, `kern.securelevel`) e um número. Esse último será o valor atual do nível de segurança do *kernel* do FreeBSD. Caso esse valor seja positivo (maior que `0`), ao menos algumas das características dos níveis de segurança estarão habilitadas.

Os níveis de segurança não podem ser diminuídos em um sistema que está funcionando se isso fosse possível o `securelevel` (nível de segurança do sistema) perderia sua funcionalidade. Caso seja necessário executar alguma tarefa que necessite que o nível de segurança seja não-positivo (por exemplo, um `installworld` ou alterar a data do sistema) será preciso alterar as definições de `securelevel` (nível de segurança do sistema) no `/etc/rc.conf` (mais precisamente, as variáveis `kern_securelevel` e `kern_securelevel_enable`) e reiniciar o sistema.

Para obter mais informações quanto aos níveis de segurança e sobre as funções específicas de cada nível, por gentileza, consulte a página de manual do init(8).

Atenção

O `securelevel` (nível de segurança do sistema) não é uma bala de prata; ele tem várias deficiências óbvias. A mais frequênte é provocar uma falsa sensação de segurança.

Um dos maiores problemas, e portanto que deve ser bem observada pelo administrador do sistema, é que, para que o `securelevel` (nível de segurança do sistema) se torne efetivo, todos os arquivos usados pelo processo de inicialização até que os níveis de segurança se tornem positivos, devem estar seguros. Se um usuário que deseja atacar o sistema, conseguir que seu código seja executado antes que o nível de segurança seja definido (o que ocorre pouco depois do processo de inicialização, visto que algumas funções que o sistema precisa realizar, não podem ser iniciadas com um nível elevado de segurança), a proteção do `securelevel` (nível de segurança do sistema) será invalidada. Por outro lado, a tarefa de assegurar que todos os

arquivos necessários pelo processo de inicialização estejam em conformidade, não é tecnicamente impossível, mas, O processo de manutenção de um ambiente em tais condições se tornaria um pesadelo, visto que seria necessário baixar o sistema, no mínimo para modo monousuário sempre que fosse necessário modificar os arquivos de configuração do mesmo.

Esse e outros pontos são freqüentemente discutidos nas listas do FreeBSD, em especial na freebsd-security. Por gentileza, queira fazer uma busca no histórico da lista, clicando aqui, para uma discussão extensa sobre o assunto. Algumas pessoas estão esperançosas de que o securelevel logo será afastado, em favor de um mecanismo de segurança mais refinado, mas as coisas ainda estão confusas a este respeito.

Considere-se advertido.

P: Tentei atualizar meu sistema para o último -STABLE, mas ele se tornou -RC ou -PRERELEASE! O que está havendo?

R: A resposta mais curta: É só um nome, RC é um acrônimo para "Release Candidate". Significa que uma nova versão está eminente. No FreeBSD, -PRERELEASE é tipicamente um sinonimo de código congelado antes de uma nova versão. (Em algumas versões, o título -BETA foi usado sob as mesmas circunstâncias em que o -PRERELEASE seria).

A resposta longa: O FreeBSD normalmente deriva suas versões de duas fontes de origem. As versões principais, ponto-zero, como o 3.0-RELEASE e o 4.0-RELEASE que são marcadas inicialmente como o topo da cadeia de desenvolvimento, normalmente chamados de -CURRENT. As versões menores (como 3.1-RELEASE ou 4.2-RELEASE), são criados a partir do *snapshot* mais recente da ramificação ativa marcada como -STABLE. A partir do 4.3-RELEASE, cada versão conta também com sua própria ramificação, que pode ser acessada por usuários que queiram apenas um nível extremamente conservador de desenvolvimento (tipicamente, apenas consultores de segurança).

Quando uma versão está para ser criada, a ramificação de onde ela se derivará deve passar por um certo processo. Parte desse processo é o congelamento do código. Quando o processo de congelamento do código se inicia, o nome desta ramificação é alterado para indicar que ela está para se tornar uma versão. Por exemplo, se a ramificação usada chamava-se 4.5-STABLE, ela passa a se chamar 4.6-PRERELEASE para indicar que o código está congelado, e indicar que testes extras, pré versão, estão acontecendo. Durante esse período alterações pertinentes a correções de problemas são realizadas. Quando o novo código está pronto para ser lançado, ele

passa a ser chamado de -RC (nesse exemplo, 4.6-RC), indicando que provavelmente a nova versão será criada a partir do código atual. Nesse estágio, apenas os problemas mais sérios são corrigidos. Depois que a versão é finalmente lançado (4.6-RELEASE nesse exemplo) e a nova ramificação com o nome dessa versão foi criada, ela passa a se chamar -STABLE; 4.6-STABLE no nosso exemplo.

Para obter mais informações sobre a numeração das versões e sobre as várias ramificações CVS, por gentileza, refira-se ao artigo sobre a Engenharia de Releases.

P: Tentei instalar um novo *kernel*, mas a rotina de chflags falhou. O que posso fazer?

R: A resposta curta: provavelmente você está com o `securelevel` (nível de segurança do sistema) acima do 0. Reinicie o sistema em modo mono usuário e instale o *kernel*.

A resposta mais completa: O FreeBSD não permite que as flags do sistema sejam alteradas caso o `securelevel` (nível de segurança do sistema) seja maior que 0. O nível atual do `securelevel` (nível de segurança do sistema) pode ser verificado com o comando:

```
# sysctl kern.securelevel
```

O `securelevel` (nível de segurança do sistema) não pode ser diminuído; é necessário iniciar o sistema em modo mono usuário, ou alterar o nível de segurança em `/etc/rc.conf` , depois reiniciar. Veja a página de manual do init(8) para obter informações mais detalhadas sobre o `securelevel` (nível de segurança do sistema), e veja também o `/etc/defaults/rc.conf` e a página de manual do rc.conf(5) para obter mais informações quanto ao rc.conf.

P: Não consigo alterar mais de um segundo na hora no meu sistema. O que posso fazer?

R: A resposta curta: provavelmente o sistema está com `securelevel` (nível de segurança do sistema) acima do 1. Reinicie o sistema em modo mono usuário e altere a data.

A resposta mais completa: O FreeBSD não permite que a hora do sistema seja alterada por mais de um segundo quando o `securelevel` (nível de segurança do sistema) do *kernel* é maior que 1. O nível atual do `securelevel` (nível de segurança do sistema) pode ser verificado com o comando:

```
# sysctl kern.securelevel
```

O `securelevel` (nível de segurança do sistema) não pode ser diminuído; é necessário iniciar o sistema em modo mono usuário, ou alterar o nível de segurança em `/etc/rc.conf` , depois reiniciar. Veja a página de manual do init(8) para obter informações mais detalhadas sobre o `securelevel` (nível de segurança do sistema), e veja também o `/etc/defaults/rc.conf` e a página de manual do rc.conf(5) para obter mais informações quanto ao rc.conf.

P: Por que motivo o `rpc.statd` está usando 256 megabytes de memória?

R: Não, mão existe nenhuma falha no uso da memória, e ele nã é usando 256MB de RAM. Ele simplesmente gosta de (ele sempre faz isso) mapear uma quantia obscena de memória em seu endereçamento, simplesmente por conveniência. Não existe nada terrivelmente errado com esse comportamento, de um ponto de vista técnico; a única questão é que assim o top(1) e o ps(1) ficam completamente perdidos.

O rpc.statd(8) mapeia seu arquivo de status (localizado sob o `/var`) no seu endereçamento para economiza preocupações sobre esse remapeamento em um segundo momento, quando o arquivo precisa crescer. O mapeamento é feito a um valor enorme. Analisando o código fonte, podemos evidenciar que o tamanho do argumento do mmap(2) é `0x10000000` , ou exatos 256MB em sistemas de arquitetura IA32.

P: Por que eu não posso retirar a flag `schg` dos arquivos?

R: O sistema está sendo executado em um nível de segurança elevado (maior que `0`). Diminua o nível de segurança e tente novamente. Para obter mais informações, por gentileza, refira-se à seção sobre `securelevel` (nível de segurança do sistema) do `FAQ`, e à página de manual do init(8)

P: Por que a autenticação do SSH via `.shosts` não funciona por padrão nas versões recentes do FreeBSD?

R: O motivo é simples. A autenticação via `.shosts` não funciona mais por padrão porque o ssh(1) não está instalado com suid de root por padrão. Razões óbvias de segurança. Para "corrigir" isto, pode-se fazer o seguinte:

- Para uma alteração permanente, defina `ENABLE_SUID_SSH` como `true` no arquivo `/etc/make.conf` e recompile o ssh (ou execute um make world).
- Uma correção temporária pode ser mudar os modos de permissão do `/usr/bin/ssh` para `4555` simplesmente executando o comando `chmod 4555 /usr/bin/ssh` logado como `root`. Depois, defina `ENABLE_SUID_SSH= true` no `/etc/make.conf` para que as alterações tenham efeito todas as vezes que um make world for feito.

P: O que é o `vnlru`?

R: O `vnlru` limpa e libera os vnodes quando o sistema atinge o limite do `kern.maxvnodes` . Essa thread do *kernel* se mantém inativa a maior parte do tempo, e só se inicia caso exista uma grande quantidade de memória RAM, e o sistema esteja acessando dezenas de milhares de arquivos pequenos.

Capítulo 11. O sistema X, sistema de interface gráfica e os Consoles Virtuais

P: Quero rodar a interface gráfica X, como procedo?

R: A maneira mais fácil é simplesmente especificar o desejo de usar o X durante o processo de instalação do FreeBSD.

Depois disso, leia e siga as instruções documentadas na ferramenta `xf86config` , que auxilia o usuário a configurar o XFree86 para os diversos monitores, placas de vídeo, mouse e etc, suportados pelo X, sistema de interface gráfica.

Também pode ser interessante dar uma olhada no servidor Xaccel. Confira a seção do `FAQ` pertinente à Xi Graphics ou Metro Link para obter mais detalhes.

P: *Tentei* rodar o X, mas o erro KDENABIO failed (Operation not permitted) sempre aparece, quando eu digito o comando `startx`. O que posso fazer?

R: Seu sistema está rodando com um `securelevel` (nível de segurança do sistema) elevado, não está? É impossível iniciar o X com um secureleve elevado. Para saber exatamente os motivos dessa inviabilidade, por gentileza, de uma olhada na página de manual do init(8).

Então, a pergunta pode ser sobre o que você deve fazer nesse caso; basicamente, existem duas escolhas: diminua seu `securelevel` (nível de segurança do sistema), colocando-o de volta para zero (normalente via `/etc/rc.conf`), ou então inicie o xdm(1) durante o processo de inicialização do sistema (antes que o `securelevel` (nível de segurança do sistema) seja elevado).

Veja a pergunta P:, para obter mais informações sobre como iniciar o xdm(1) durante o boot.

P: Por que meu mouse não funciona com o X?

R: Caso esteja usando o syscons (o driver padrão do console), o FreeBSD pode ser configurado para suportar um cursor de mouse em cada tela virtual. Com o intúito de evitar conflitos com o X, o syscons suporta um dispositivo virtual, chamado `/dev/sysmouse` . Todos os eventos relacionados ao mouse, que o sistema recebe, são antes enviados para o device sysmouse, por meio do moused. Se a intenção é usar o mouse em um ou mais consoles virtuais, e também usar o X, leia P: e configure o moused.

Depois, edite o `/etc/XF86Config` e garanta que existam as seguintes linhas no arquivo:

```
Section         Pointer
Protocol        -"SysMouse"
Device          -"/dev/sysmouse"
.....
```

O exemplo acima refere-se ao XFree86 3.3.2 e posteriores. Para versões anteriores, a cláusula *Protocol* deve ser substituída por *MouseSystems*.

Alguns preferem usar a device `/dev/mouse` sob o X. Para que isso funcione, faça um link de `/dev/mouse` para `/dev/sysmouse` (veja a página de manual do sysmouse(4)).

```
# cd -/dev
# rm --f mouse
# ln --s sysmouse mouse
```

P: Meu mouse possui aquela bolinha (esfera) simpática de scroll. Posso usa-lo no X?

R: Pode, mas é necessário customizar os programas do X. Veja a página do Colas Nahaboo sobre o assunto (http://www.inria.fr/koala/colas/mouse-wheel-scroll/.

Caso queira usar o programa imwheel, simplesmente siga os seguintes passos:

1. Traduza os eventos da esfera de scroll:

   O programa imwheel funciona assim: ele traduz os botões 4 e 5 do mouse em eventos do teclado do computador. Dessa forma é necessário assegurar que o driver do mouse esteja traduzindo os eventos da esfera de scroll para os eventos dos botões 4 e 5, ou seja assimilar suas funções. Existem duas formas de fazer isso, a primeira é usando o moused(8) para fazer essas assimilações, e a segunda, é usar o próprio X para traduzir os eventos.

   a. Usando o moused(8) para traduzir os eventos da bolinha de scroll.

      Para que o moused(8) faça as assimilações de eventos, basta adicionar as opções `-z 4` nas opções de linhas de comando, usadas para iniciar o moused(8). Por exemplo, se normalmente você inicia o moused(8) via `moused -p /dev/psm0` basta substituir o comando por `moused -p /dev/psm0 -z 4`. Se o moused(8) é executado automaticamente durante o processo de inicialização do FreeBSD, por meio das entradas definidas no `/etc/rc.conf`, basta adicionar `-z 4` na variável `moused_flags` do `/etc/rc.conf`.

      Você precisa agora dizer para o X que você tem o botão 5 no mouse. Para fazer isto, simplesmente adicione a linha `Buttons 5` para a seção "Pointer" do `/etc/XF86Config`. Por exemplo, você pode seguir a seção "Pointer" em `/etc/XF86Config`.

Exemplo 11.1. Seção “Pointer” no XF86Config para o mouse com bolinha de scroll, da série 3.3.x do XFree86, usando a tradução se;rie 3.3.x do XFree86, usando a tradução por meio do moused

```
Section -"Pointer"
Protocol        -"SysMouse"
Device          -"/dev/sysmouse"
Buttons         5
EndSection
```

Exemplo 11.2. Seção “InputDevice” do XF86Config para usar a tradução do X Server na série 4.X do XFree86.

```
Section -"InputDevice"
Identifier      -"Mouse1"
Driver          -"mouse"
Option          -"Protocol" -"auto"
Option          -"Device" -"/dev/sysmouse"
Option          -"Buttons" -"5"
EndSection
```

Exemplo 11.3. Exemplo de “.emacs” para usar paginamento em mouse com bolinha de scroll.

```
;; wheel mouse
(global-set-key [mouse-4] -'scroll-down)
(global-set-key [mouse-5] -'scroll-up)
```

b. Usando o X Server para traduzir os eventos da esfera de scroll.

Se você não usa o moused(8) ou simplesmente não quer que ele faça a tradução de eventos, é possível que o servidor X faça o trabalho, no lugar do moused(8). Essa ação requer algumas alterações no seu arquivo `/etc/XF86Config`. Primeiro, é necessário definir o protocolo apropriado para o mouse. A maioria dos mouses com esferas de scroll usam o protocolo "IntelliMouse". De qualquer forma, o XFree86 não suporta outros protocolos como o "MouseManPlusPS/2" dos MouseMan+ Logitechfor. Uma vez definido o protocolo, é necessário criar uma entrada apropriada na seção "Pointer".

Depois, é preciso definir que o servidor X deve remapear os eventos 4 e 5 do mouse. A opção `ZAxisMapping` é usada para essa finalidade.

Por exemplo, caso não estejas usando o moused(8) e exista um IntelliMouse ligado na PS/2 do seu computador, use o seguinte, no `/etc/XF86Config`.

Exemplo 11.4. Seção "Pointer" do **XF86Config** com um mouse com scroll na série 3.3.x do XFree86.

```
Section -"Pointer"
Protocol        -"IntelliMouse"
Device          -"/dev/psm0"
ZAxisMapping    4 5
EndSection
```

Exemplo 11.5. Seção "InputDevice" do **XF86Config** com um mouse com scroll na série 4.x do XFree86.

```
Section -"InputDevice"
Identifier      -"Mouse1"
Driver          -"mouse"
Option          -"Protocol" -"auto"
Option          -"Device" -"/dev/psm0"
Option          -"ZAxisMapping" -"4 5"
EndSection
```

Exemplo 11.6. Arquivo “.emacs” para usar paginamento em mouse com bolinha de scroll.

```
;; wheel mouse
(global-set-key [mouse-4] -'scroll-down)
(global-set-key [mouse-5] -'scroll-up)
```

2. Instale o imwheel

   Depois, instale o imwheel à partir da coleção de `ports` do FreeBSD; ele pode ser encontrado sob a categoria `x11`. A finalidade desse programa é assimilar os eventos dos botões 4 e 5 do mouse, com os eventos de alguma tecla do teclado. Por exemplo, o programa deve enviar o evento da tecla Page Up quando a esfera for deslocada para frente. O imwheel usa um arquivo de configurações para assimilar esses eventos à uma tecla, de forma que possam ser configuradas ações diferentes (teclas diferentes) para aplicações diferentes. O arquivo de configuração padrão do imwheel é instalado em `/usr/X11R6/etc/imwheelrc` . Ele pode ser copiado para `~/.imwheelrc` e editado, caso se deseja customizar o arquivo de configuração. O formato esperado para o arquivo é documentado na página de manual do imwheel(1).

3. Configure o Emacs para trabalhar em conjunto com o Imwheel (*optional*)

   Se você usa o emacs ou o Xemacs, será necessário adicionar uma breve seção ao arquivo `~/.emacs` . No emacs, adicione o seguinte:

Exemplo 11.7. Configuração do Emacs para Imwheel

```
;;; For imwheel
(setq imwheel-scroll-interval 3)
(defun imwheel-scroll-down-some-lines ()
  (interactive)
  (scroll-down imwheel-scroll-interval))
(defun imwheel-scroll-up-some-lines ()
  (interactive)
  (scroll-up imwheel-scroll-interval))
(global-set-key [?\M-\C-\)] -'imwheel-scroll-up-some-
lines)
(global-set-key [?\M-\C-\(] -'imwheel-scroll-down-some-
lines)
```

```
;;; end imwheel section
```

Pro Xemacs, adicione o seguinte, no seu arquivo `~/.emacs` :

Exemplo 11.8. Configuração do Xemacs para Imwheel

```
;;; For imwheel
(setq imwheel-scroll-interval 3)
(defun imwheel-scroll-down-some-lines ()
  (interactive)
  (scroll-down imwheel-scroll-interval))
(defun imwheel-scroll-up-some-lines ()
  (interactive)
  (scroll-up imwheel-scroll-interval))
(define-key global-map [(control meta \))] -'imwheel-
scroll-up-some-lines)
(define-key global-map [(control meta \()] -'imwheel-
scroll-down-some-lines)
;;; end imwheel section
```

4. Execute o Imwheel

   Basta digitar `imwheel` em algum terminal X (xterm) para inicia-lo, uma vez que tudo esteja pronto. Imediatamente o programa vai estar efetivo e vai se tornar um processo em segundo plano. Caso queira sempre iniciar o imwheel, basta adicionar o comando no seu arquivo `.xinitrc` ou no `.xsession` . É possível que o imwheel mostre algumas mensagens de advertência sobre arquivos PID; elas podem ser seguramente ignoradas, visto que são mensagens que se aplicam à versão para Linux.

P: Por quê os menus e caixas de diálogo do X, sistema de interface gráfica não funcionam direito?

R: Tente desativar a tecla Num Lock.

Se por padrão seu Num Lock é ativo na hora do processo de inicialização, adicione a seguinte linha a seção `Keyboard` do seu arquivo `XF86Config` .

```
# Deixar o servidor fazer o trabalho do NumLock.  Deve ser ↺
usado apenas em versoes anteriores a R6
    ServerNumLock
```

P: O que é um console virtual, e como eu crio mais consoles?

R: Consoles virtuais simplesmente permitem que se tenha várias sessões simultâneas em uma mesma máquina, sem a necessidade de fazer nada complicado como configurar uma rede ou usar um servidor X.

Quando o sistema é iniciado, a primeira ação é apresentar um prompt de login na tela do usuário, tão logo todas as mensagens do processo de inicialização sejam apresentadas. Nesse momento é possível entrar com seu nome de usuário e senha para começar trabalhar (ou brincar!) no primeiro console virtual.

Em algum momento, é provável que se deseje iniciar uma outra sessão, talvez para ler a documentação de alguma aplicação que está sendo usada, ou para ler e-mail enquanto a transferência FTP se conclúi, enfim, qualquer ação (a)típica de um sistema multitarefa. Nesse caso, basta pressionar Alt+F2 (segure a tecla Alt e depois aperte a tecla F2), e outro prompt de login estará esperando você no segundo "console virtual"! Quando quizer alternar de volta à sessão original, digite Alt+F1.

A instalação padrão do FreeBSD oferece três consoles virtuais já habilitados (8 a partir do 3.3-RELEASE), e as teclas Alt+F1, Alt+F2, e Alt+F3 irá alternar entre esses consoles.

Para habilitar mais consoles, edite o `/etc/ttys` (veja a página de manual do ttys(5)) e adicione as entradas da `ttyv4` à `ttyvc` depois do comentário sobre "Virtual terminals":

```
# Edite as entradas existentes para ttyv3 e mude de -"off" ↺
para -"on"
ttyv3   -"/usr/libexec/getty Pc"         cons25  on secure
ttyv4   -"/usr/libexec/getty Pc"         cons25  on secure
ttyv5   -"/usr/libexec/getty Pc"         cons25  on secure
ttyv6   -"/usr/libexec/getty Pc"         cons25  on secure
ttyv7   -"/usr/libexec/getty Pc"         cons25  on secure
ttyv8   -"/usr/libexec/getty Pc"         cons25  on secure
ttyv9   -"/usr/libexec/getty Pc"         cons25  on secure
ttyva   -"/usr/libexec/getty Pc"         cons25  on secure
ttyvb   -"/usr/libexec/getty Pc"         cons25  on secure
```

Use quantos consoles desejar. Quanto mais, maior o uso de recursos; essa é uma consideração relevante quando se tem 8MB de RAM ou menos. Também pode ser interessante mudar o terminal de `secure` para `insecure`.

Importante

Caso se deseje usar um servidor X, é necessário garantir que exista ao menos um terminal virtual fora de uso (ou desligado). Com isso, entenda que, se sua inteção for usar

consoles virtuais nas suas doze teclas de funções, nada feito; apenas onze poderão ser usadas caso deseje-se usar o X na mesma máquina.

A maneira mais simples de desabilitar um console, é desligando-o. Por exemplo, caso existam 12 terminais definidos, como mencionado na situação acima, e se queira usar o servidor X, o mais interessante é mudar as configurações do terminal 12 de:

```
ttyvb   -"/usr/libexec/getty Pc"        cons25  on  secure
```

para:

```
ttyvb   -"/usr/libexec/getty Pc"        cons25  off  secure
```

Caso seu teclado tenha apenas dez teclas de funções, basta encerrar as definições com:

```
ttyv9   -"/usr/libexec/getty Pc"         cons25  off secure
ttyva   -"/usr/libexec/getty Pc"         cons25  off secure
ttyvb   -"/usr/libexec/getty Pc"         cons25  off secure
```

(Claro que as linhas poderiam simplesmente ser apagadas.)

Uma vez editado o `/etc/ttys` , o passo seguinte é garantir que existam devices o bastante pros terminais virtuais. A forma mais fácil de fazer isso é:

```
# cd -/dev
# sh MAKEDEV vty12
```

Em seguida, a maneira mais fácil (e mais limpa) de ativar cada um dos consoles virtuais é reiniciar o sistema. Mas se reiniciar o FreeBSD não é a intenção, basta desligar o servidor X, sistema de interface gráfica e executar (logado como `root`):

```
# kill --HUP 1
```

É obrigatório tirar por completo o X, sistema de interface gráfica do ar antes de dar esse comando, caso o X esteja sendo usado. Se isso não for feito, o sistema vai parecer que travou.

P: Como posso acessar os consoles virtuais quando eu estiver no X?

R: Use Ctrl+Alt+F***n*** para alternar de volta para algum console virtual. Por exemplo, Ctrl+Alt+F1 retornaria ao primeiro console virtual.

Uma vez de volta ao console textual, pode-se usar Alt+F***n*** normalmente, para alternar entre os consoles virtuais.

Pra voltar para sessão X basta alternar para o console virtual onde o X está sendo executado. Caso o X tenha sido iniciado por linha de comando (por exemplo, com o comando `startx`) a sessão terá sido assimilada ao próximo console virtual fora de uso, e não ao console onde o comando foi digitado. Caso existam oito terminais virtuais ativos, o X estará sendo executado no nono. Nesse caso as teclas Alt+F9 retornarão ao sistema gráfico.

P: Como eu inicio o XDM no processo de inicialização?

R: Existem duas formas clássicas de iniciar o xdm. A primeira consiste em inciá-lo a partir do `/etc/ttys` (veja a página de manual do ttys(5)) usando o exemplo disponível no arquivo; a segunda forma é simplesmente executar o xdm a partir do `rc.local` (veja a página de manual do rc(8)) ou então por um script `X.sh` em `/usr/local/etc/rc.d` . As duas maneiras são igualmente válidas, mas algumas podem ser mais eficientes em algumas situações, onde a outra forma não seria ideal. Nos dois casos, o resultado será o mesmo: o X iniciará o mostrando uma tela de login: gráfica.

O método de inicialização via ttys oferece a vantagem de definir explicitamente em qual vtyX o servidor gráfico vai ser carregado, passando a responsabilidade da reinicialização do X para o init, no momento do logout. O método via rc.local oferece facilidades caso seja necessário encerrar o processo xdm, no caso, por exemplo, de ocorrerem problemas ao carregar o servidor gráfico.

Ao usar o rc.local para carregar o `xdm`, ele não deve ser acompanhado de nenhum argumento (deve ser iniciado como um daemon e deve ser iniciado DEPOIS que o getty já estiver em execussão, senão é provável que ocorram conflitos entre ambos, podendo travar o console. A melhor forma de assegurar o correto funcionamento desse método é fazer com que o script espere 10 segundos (por exemplo, com um sleep 10;) antes de iniciar o xdm.

Se a inteção é iniciar o `xdm` a partir do `/etc/ttys` , ainda existe a probabilidade de conflitos entre o `xdm` e o getty(8). Uma forma interessante de evitar esse tipo de desconforto, é definir, no arquivo `/usr/X11R6/lib/X11/xdm/Xservers` , o número do `vt` onde o X deve ser iniciado, da seguinte forma:

```
:0 local -/usr/X11R6/bin/X vt4
```

O exemplo acima indica que o servidor X será ativado no `/dev/ttyv3` . Note que existe um offset de um vt, já que o X começa a contar os terminais (vty) a partir do um, enquando o *kernel* do FreeBSD os conta a partir do zero.

P: Por que eu enfrento um Couldn't open console ao executar o xconsole?

R: Se o X for iniciado com um `startx`, as permissões do `/dev/console` não serão redefinidas, resultando em situações onde um `xterm -C` ou mesmo o `xconsole` não funcionarão corretamente.

O motivo disso é a forma como as permissões são definidas por padrão. Em sistemas multiusuário, normalmente não se espera que qualquer pessoa possa escrever no console do sistema. Para os usuários que estão se logando diretamente na máquina, em algum VTY, existe o arquivo fbtab(5) que resolve esse tipo de problema.

Se for apropriado, garanta que exista uma linha assim

```
/dev/ttyv0 0600 -/dev/console
```

No arquivo `/etc/fbtab` (veja a página de manual do fbtab(5)). Essa linha garantirá que qualquer usuário que se logar no `/dev/ttyv0` será também proprietário do console.

P: Antes eu conseguia usar o XFree86 com um usuário sem privilégios. Porque agora o servidor diz que eu tenho que ser `root`?

R: Todo servidor gráfico precisa ser executado como `root` para que o sistema permita acesso direto aos equipamentos de vídeo. Acontece que nas versões mais antigas, o XFree86 (versões <= 3.3.6) instalava o servidor de forma que ele era automaticamente executado como `root` (setuid de `root`). Óbviamente esse comportamente implica em riscos de segurança em qualquer caso onde o programa em questão seja complexo e grande; esse é o caso dos servidores X. As versões mais atuais do XFree86 não instalam os servidores gráficos com todo esse poder, exatamente por esse motivo.

É claro que rodar o X como usuário `root` não é uma idéia muito aceitável, especialmente em relação à segurança. Existem duas formas de usar o X como usuário comum. A primeira é usar o `xdm` ou qualquer outro gerenciador de display (como o `kdm`); a segunda é usar o `Xwrapper`.

O `xdm` é um daemon que controla logins gráficos. Normalmente ele é iniciado no processo de inicialização e é responsável pela autenticação dos usuários, e por inciar suas sessões; essencialmente é a união gráfica do getty(8) como o login(1). Para mais informações sobre o `xdm`, por gentileza, refira-se à documentação do XFree86 e à questão do `FAQ` sobre xdm.

O `Xwrapper` é um intermediador do servidor gráfico; é um programa bem pequeno que possibilita a inicialização manual do servidor gráfico por qualquer usuário, garantindo razoável segurança à operação. O programa ainda faz algumas verificações na linha de comando definida pelo usuário, para garantir a sanidade das intenções do mesmo. Se todas as intenções forem aprovadas, ele executa o X. Se por qualquer razão, a idéia de usar um gerenciador de displays não te agrada, o `Xwrapper` é feito para você. Caso a coleção de `Ports` esteja instalada, o programa pode ser encontrado em `/usr/ports/x11/wrapper`.

P: Por que meu mouse PS/2 não se comporta corretamente no X?

R: O seu mouse e a device que o controla devem ter desincronizado.

Nas versões 2.2.5 e anteriores, a simples alternância entre o X e o terminal, e voltar para o X, força a resincronização do mouse. Se o problema se tornar frequênte, adicione a seguinte opção ao arquivo de configuração do seu *kernel*, e o recompile:

```
options PSM_CHECKSYNC
```

Veja a seção sobre a compilação do *kernel*, caso você não tenha experiência com isso.

Com essa opção as chances de ter problemas com a sincronia do mouse são bem pequenas. Contudo, se ainda assim o problema persistir, clique em qualquer botão durante o movimento do mouse. É o bastante para resincroniza-lo.

Infelizmente essa opção pode não funcionar em alguns sistemas, dependendo de qual driver controle o seu mouse PS/2; especialmente se a device de controle for do tipo ALPS GlidePoint.

Na versão 2.2.6 e posteriores, a verificação de sincronia se tornou razoávelmente melhor, e é padrão nos mouses PS/2. Deve funcionar corretamente com GlidePoint, inclusive (como o código de verificação de sincronia ter se tornado padrão, a opção PSM_CHECKSYNC não existe mais). Contudo, em situações muito raras, o driver de controle do mouse pode, errôneamente reportar problemas de sincronização, mostrando a seguinte mensagem do *kernel*:

```
psmintr: out of sync (xxxx -!= yyyy)
```

Pensando que seu mouse não está funcionando corretamente.

Se for o caso, desligue o código de verificação de sincronia do mouse PS/2, definindo a flag 0x100 na device de controle do mesmo. Entre no modo *UserConfig* definindo a opção `-c` na tela do processo de inicialização:

```
boot: -c
```

Depois, na linha de comando do *UserConfig*, digite:

```
UserConfig> flags psm0 0x100
UserConfig> quit
```

P: Por que meu mouse PS/2 da MouseSystems não funciona?

R: Existem notícias que alguns modelos de mouse PS/2 da MouseSystems funcionam corretamente apenas em modo de alta resolução. Do contrário, o cursor do mouse costuma pular para diagonal superior esquerda da tela com certa frequência.

Infelizmente não existe solução à esse problema, nas versões 2.0.X e 2.1.X. Contudo, das versões 2.2 à 2.2.5, basta aplicar o seguinte patch, no `/sys/i386/isa/psm.c` e recompilar o *kernel*. Veja a seção sobre compilação do *kernel* caso não tenha experiência com o assunto.

```
@@ --766,6 +766,8 @@
     if (verbose >= 2)
         log(LOG_DEBUG, -"psm%d: SET_DEFAULTS return code:%04x
\n",
             unit, i);
+    set_mouse_resolution(sc->kbdc, PSMD_RES_HIGH);
+
 #if 0
     set_mouse_scaling(sc->kbdc);        -/* 1:1 scaling */
     set_mouse_mode(sc->kbdc);               -/* stream mode */
```

Na versão 2.2.6 e versões posteriores, basta especificar a flag 0x04 para device de controle do mouse PS/2, colocando-o em modo de alta resolução. Entre no modo *UserConfig* com a opçãp `-c` na tela do processo de inicialização:

```
boot: -c
```

Depois, na linha de comando do *UserConfig* digite:

```
UserConfig> flags psm0 0x04
UserConfig> quit
```

Veja a pergunta anterior, sobre outra causa possível de problemas com o mouse.

P: Ao compilar uma aplicação X, o `imake` não consegue encontrar o `Imake.tmpl` . Onde ele está?

R: O `Imake.tmpl` é parte do pacote Imake, uma aplicação padrão para construção de aplicações gráficas. O `Imake.tmpl` , assim como vários outros arquivos de cabeçalhos necessários para compilar aplicações gráficas, é parte da distribuição do X. Eles podem ser instalados pelo sysinstall ou manualmente a partir dos arquivos da distribuição.

P: Estou construindo uma aplicação gráfica que depende do XFree86 3.3.X, mas eu estou com o XFree86 4.X instalado. O que fazer?

R: Pra definir que a construção do `Port` deve ser linkada às bibliotecas do XFree86 4.X, adicione o seguinte, no seu `/etc/make.conf` , (se o arquivo não existir, crie-o):

```
XFREE86_VERSION= 4
```

P: Como posso inverter as funções dos botões do mouse?

R: Execute o comando `xmodmap -e "pointer = 3 2 1"` à partir do `.xinitrc` ou do `.xsession` .

P: Como instalar uma Splash Screen e onde posso encontra-las?

R: A partir da lançamento do FreeBSD 3.1, uma nova característica foi adicionada ao sistema, permitindo que alguns arquivos de imagens sejam usados como “Splash Screens” durante as mensagens do processo de inicialização. Tais imagens devem

ser arquivos do tipo bitmap com 256 cores (`*.BMP`) ou então ZSoft PCX (`*.PCX`). Devem ainda ter resolução de 320x200 pixels (ou menos), para funcionarem corretamente em adaptadores de vídeo VGA tradicionais. Caso o *kernel* tenha sido compilado com suporte à VESA, então podem ser usados bitmaps maiores, até 1024.768 px. O suporte à VESA pode ser diretamente compilado no *kernel*, com a opção VESA no arquivo de configuração, ou carregado como módulo, com o kld, durante o processo de inicialização do sistema.

Para definir a "Splash Screens", basta modificar alguns arquivos de inicialização que controlam o processo de inicialização do FreeBSD. Tais arquivos foram alterados na versão 3.2 do FreeBSD, existindo portanto duas formas de carregar uma "Splash Screens":

- No FreeBSD 3.1

  O primeiro passo é escolher o seu bitmap, e sua versão. Até o FreeBSD 3.1, apenas os bitmaps do tipo Windows eram suportados. Assim que escolher (ou criar) sua "Splash Screens", copie-a para `/boot/splash.bmp` . Depois, basta editar (ou criar, caso não exista) o arquivo `/boot/loader.rc` e adicionar as seguintes linhas:

  ```
  load kernel
  load --t splash_image_data -/boot/splash.bmp
  load splash_bmp
  autoboot
  ```

- No FreeBSD 3.2 e posteriores

  Além de adicionar suporte a "Splash Screens" de formato PCX, o FreeBSD 3.2 passou a oferecer uma maneira mais interessante de configurar o processo de inicialização. Caso prefira, o método descrito acima, para o FreeBSD 3.1 também funciona. Nesse caso, se a imagem for do tipo PCX basta substituir a entrada `splash_bmp` por `splash_pcx` . Caso queira usar a nova configuração do processo de inicialização, basta criar um arquivo `/boot/loader.rc` com o seguinte conteúdo:

  ```
  include -/boot/loader.4th
  start
  ```

  e depois, um `/boot/loader.conf` com o seguinte:

  ```
  splash_bmp_load="YES"
  bitmap_load="YES"
  ```

  Essa configuração assume que o `/boot/splash.bmp` deve ser usado como sua "Splash Screens". Caso prefira usar um arquivo PCX, copie para o `/boot/splash.pcx` , e crie um `/boot/loader.rc` , da forma como foi indicado anteriormente; depois crie um `/boot/loader.conf` com o seguinte:

```
splash_pcx_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.pcx"
```

Agora você só precisa de uma imagem, para servir de “Splash Screens”. Pra isso, dê uma navegada na galeria disponível em http://www.baldwin.cx/splash/.

P: Posso usar as teclas do Windows® que meu teclado possui, sob o X?

R: Pode. Basta usar o xmodmap(1) para redefinir a função das teclas.

Assumindo que todos os teclados “Windows®” sejam padrão, os códigos de mapeamento pras 3 teclas são:

- 115 - Windows®, entre a tecla Ctr e a Alt do lado esquerdo.
- 116 - Windows®, à direita a tecla AltGr.
- 117 - Menu, do lado esquerdo da tecla Ctrl esquerda

Por exemplo, para fazer com que a tecla Windows® esquerda imprima uma vírgula, faça o seguinte:

```
# xmodmap --e -"keycode 115 = comma"
```

É provável que seu gerenciador de janelas tenha que ser reiniciado, para visualizar o resultado.

Pra forçar o carregamento automático do mapeamento das teclas Windows®, coloque os comandos do `xmodmap` no arquivo `~/.xinitrc` ou de preferência, crie um arquivo `~/.xmodmaprc` e inclua as opções do `xmodmap` uma por linha, nesse arquivo. Depois adicione:

```
xmodmap $HOME/.xmodmaprc
```

No seu `~/.xinitrc` .

Por exemplo, pode-se mapear as 3 teclas em questão para fazer o papel das teclas F13, F14, e F15, respectivamente. Dessa forma, seria fácil mapear as aplicações de forma que as teclas tivessem ações no seu sistema, como veremos agora.

Adicione o seguinte conteúdo, no arquivo `~/.xmodmaprc` .

```
keycode 115 = F13
keycode 116 = F14
keycode 117 = F15
```

Se o gerenciador de janelas em questão for o `fvwm2`, por exemplo, pode-se mapear as teclas de forma que o F13 minimize (ou maximize) a janela que o cursor está

apontando, a tecla F14 de forma que ela traga a janela marcada pelo cursor para frente (ou volte para trás, caso já esteja à frente), e o F15 pode alternar o menu da área detrabalho principal, o que é bem útil quando a tela não é visível.

As seguintes definições no `~/.fvwmrc` implementam a configuração acima descrita:

```
Key F13 FTIWS    A Iconify
Key F14        FTIWS    A        RaiseLower
Key F15        A        A        Menu Workplace Nop
```

P: Como posso obter a aceleração de equipamentos 3D para o OpenGL?

R: A disponibilidade da aceleração 3D depende da versão do XFree86 e da placa de vídeo. Caso a placa seja NVIDIA, verifique a página da Iniciativa de Driver NVIDIA para o FreeBSD, que discute a aceleração 3D em chips NVIDIA com XFree86-4. Pra outras placas em conjunto com o XFree86-4, incluindo a Matrox G200/G400, a ATI Rage 128/Radeon, as 3dfx Voodoo 3, 4, 5, e Banshee, refira-se à página sobre Renderização Direta do XFree86-4 no FreeBSD. Usuários do XFree86 na versão 3.3 podem usar o `port` do Utah-GLX que pode ser encontrado em graphics/utah-glx para conseguir alguma (limitada) aceleração 3D para o OpenGL em placas Matrox Gx00, ATI Rage Pro, SiS 6326, i810, Savage, e algumas NVIDIA antigas.

Capítulo 12. Redes

P: Onde obter informações a respeito do processo de processo de inicialização sem disco rígido (*diskless booting*)?

R: O processo de processo de inicialização sem disco implica na possibilidade de uma máquina FreeBSD ser inicializada através da rede, lendo os arquivos necessários à partir de um servidor, ao invés de um disco rígido. Para maiores detalhes, por favor, consulte o ítem *diskless booting* no Manual do FreeBSD.

P: Um sistema FreeBSD pode ser utilizado como roteador dedicado para uma rede?

R: Pode. Por gentileza, refira-se à documentação do Manual do FreeBSD sobre configurações avançadas de rede, mais especificamente, a seção sobre roteamento e gateways.

P: Posso conectar minha máquina com Win95 à Internet, através do meu FreeBSD?

R: Normalmente, as pessoas que propõem esse tipo de questão possuem dois computadores em casa, um com FreeBSD e outro com Win95. A idéia é utilizar a maquina com FreeBSD para se conectar à Internet, e então oferecer acesso Internet a máquina Win95 através do FreeBSD. Essa é apenas uma extensão especial da questão anterior.

... e a resposta é sim! No FreeBSD 3.x, o ppp(8) em modo usuário oferece a opção `-nat`. Se o ppp(8) for executado com essa opção, basta definir a variável `gateway_enable` para *YES* no arquivo `/etc/rc.conf` , e configurar corretamente a máquina Windows. Isso é o bastante.

Para obter mais informações, por gentileza, refira-se a página de manual do ppp(8)

Se o ppp(8) estiver sendo usado em modo *kernel* (*kernel-mode*) ou a conexão com a Internet for via Ethernet, a opção mais viável será utilizar o natd(8). Por favor, consulte a seção natd dessa documentação.

P: O FreeBSD suporta SLIP e PPP?

R: Claro. Veja as páginas de manual do slattach(8), sliplogin(8), ppp(8), e pppd(8). O ppp(8) e o pppd(8) oferecem suporte à conexões entrantes e de saída (conexões incoming/outgoing), enquanto o slattach(8) à conexões de saída (outgoing).

Para obter mais informações sobre a correta utilização desses recursos, por gentileza, refira-se ao Capítulo sobre PPP e SLIP do Manual do FreeBSD.

Se o seu acesso à Internet for apenas por meio de uma conta Shell, pode ser interessante dar uma olhada no `port` da aplicação net/slirp. Esse `port` oferece acesso (limitado) à serviços como FTP e HTTP direto da máquina local.

P: O FreeBSD suporta NAT ou Masquerading?

R: Se no seu caso, existe uma subrede (uma ou mais máquinas locais interconectadas em rede), mas o seu Provedor de Internet disponibiliza apenas um IP (ou se o endereço IP em questão é dinâmico), com certeza é interessante dar uma olhada no natd(8). O natd(8) possibilita que uma subrede inteira acesse a Internet através de um único endereço IP.

O ppp(8) oferece suporte interno à essa mesma funcionalidade, através da opção `-nat` do programa. A biblioteca libalias(3) é usada tanto pelo ppp(8) quanto pelo natd(8).

P: Como posso conectar duas estações FreeBSD por linha paralela, usando o PLIP?

R: Por gentileza, refira-se à seção sobre PLIP do Manual do FreeBSD.

P: Por que eu não posso criar um dispositivo /dev/ed0?

R: Porque não é preciso! Na estrutura de redes de Berkeley, as interfaces de rede são acessadas somente (e diretamente) pelo código do *kernel*. Por favor verifique o arquivo `/etc/rc.network` e as páginas do manual para os diversos programas de rede ali mencionados, para maiores informações. Se isto deixá-lo completamente confuso, consulte um livro que descreva a administração de rede em um outro sistema operacional baseado no modelo BSD. Com poucas exceções significativas, a administração de rede em sistemas FreeBSD é basicamente a mesma da utilizada em sistemas como o SunOS 4.0 ou o Ultrix.

P: Como eu configuro aliases (apelidos) de Ethernet?

R: Se a intenção é definir um apelido IP para uma subrede previamente configurada, basta adicionar a máscara `0xffffffff` junto à sintaxe usual para definição de alias no ifconfig(8):

```
# ifconfig ed0 alias 192.0.2.2 netmask 0xffffffff
```

Do contrário, basta definir o endereço de rede e a netmask em questão, da forma tradicional:

```
# ifconfig ed0 alias 172.16.141.5 netmask 0xffffff00
```

P: Como eu configuro minha placa 3Com 3C503 para utilizar outra interface de conexão ?

R: Se você deseja utilizar uma outra interface de conexão, deverá especificar alguns parâmetros adicionais na linha de comando do ifconfig(8). A porta padrão é a `link0`. Para usar a porta AUI ao invés da porta BNC utilize a flag `link2`. Tais flags devem ser definidas através das variáveis ifconfig_* no arquivo `/etc/rc.conf` . (consulte o rc.conf(5)).

P: Por que eu tenho problemas com NFS no FreeBSD?

R: Certas interfaces de rede para PC são melhores do que outras (para adotarmos um eufemismo) e as vezes podem causar problemas em aplicações que utilizam a rede de modo intensivo, como o NFS.

Consulte o item NFS do Manual do FreeBSD para obter mais informações sobre o assunto.

P: Por que não é possível montar sistemas de arquivos NFS de máquinas Linux?

R: Algumas versões do código NFS do Linux aceitam requisições de montagem provenientes apenas de portas privilegiadas, experimente o comando:

```
# mount --o --P linuxbox:/blah -/mnt
```

P: Por que não é possível montar sistemas de arquivos NFS de máquinas Sun?

R: Estações de trabalho Sun rodando SunOS 4.X aceitam requisições de montagem provenientes apenas de portas privilegiadas, experimente o comando:

```
# mount --o --P sunbox:/blah -/mnt
```

P: Por que o `mountd` informa repetidamente que can't change attributes e bad exports list utilizando o servidor de NFS do FreeBSD?

R: O problema mais freqüente é o não entendimento correto do formato do arquivo `/etc/exports` . Por gentileza, leia com atenção a página de manual do exports(5) e a documentação sobre NFS no Manual do FreeBSD, especialmente a seção sobre a configuração do NFS.

P: Por que existem problemas na comunicação (via protocolo PPP) com máquinas NeXTStep?

R: Experimente desabilitar a variável TCP extensions no arquivo `/etc/rc.conf` (consulte rc.conf(5)) alterando a variável abaixo para NO:

```
tcp_extensions=NO
```

Máquinas Annex da Xylogic também apresentam um problema similar neste aspecto, e você deve adotar a mesma solução para conectar-se a estes sistemas.

P: Como habilitar o suporte a IP multicast?

R: Desde o FreeBSD 2.0 que operações Multicast são completamente suportadas por padrão. Se a itenção é fazer o sistema FreeBSD atuar como um roteador multicast, será necessário que o *kernel* do sistema seja recompilado com a opção `MROUTING` e que o mrouted(8) seja executado. O FreeBSD, à partir da versão 2.2, pode iniciar

o mrouted(8) durante o processo de inicialização se a variável `mrouted_enable` estiver configurada com o parâmetro `"YES"` no arquivo `/etc/rc.conf` .

As ferramentas MBONE estão disponíveis em sua própria categoria na coleção de `ports`, mbone. Se você está procurando as ferramentas de conferência `vic` e `vat`, procure neste diretório!

P: Quais interfaces de rede são baseadas no chipset DEC PCI?

R: Esta é uma lista compilada por Glen Foster <gfoster@driver.nsta.org >, com algumas adições recentes:

Tabela 12.1. Interfaces de rede baseadas no chipset DEC PCI

Vendedor	Modelo
ASUS	PCI-L101-TB
Accton	ENI1203
Cogent	EM960PCI
Compex	ENET32-PCI
D-Link	DE-530
Dayna	DP1203, DP2100
DEC	DE435, DE450
Danpex	EN-9400P3
JCIS	Condor JC1260
Linksys	EtherPCI
Mylex	LNP101
SMC	EtherPower 10/100 (Model 9332)
SMC	EtherPower (Model 8432)
TopWare	TE-3500P
Znyx (2.2.x)	ZX312, ZX314, ZX342, ZX345, ZX346, ZX348
Znyx (3.x)	ZX345Q, ZX346Q, ZX348Q, ZX412Q, ZX414, ZX442, ZX444, ZX474, ZX478, ZX212, ZX214 (10mbps/hd)

P: Por que preciso utilizar um FQDN (Nomes de domínio completamente qualificados) pras estações da minha rede?

R: Provavelmente você vai estar trabalhando com estações em domínios diferentes. Por exemplo, se você esta em foo.bar.edu e deseja alcançar uma estação chamada `mumble` no domínio `foo.bar.edu` , deverá referir-se à essa esse host através do seu nome de domínio qualificado, `mumble.foo.bar.edu` , ao invés de apenas `mumble.`

Normalmente era possível alcançar a estação apenas por seu nome. Essa função era realizada pelos resolvedores BIND do ISC. Contudo, as versões atuais do BIND (veja o named(8)) que acompanham o FreeBSD não oferecem mais abreviações padrão para domínios que não sejam FQDN, com a única exceção do domínio que sua própria estação faz parte. Dessa forma, o host `mumble`, se não for localizado como `mumble.foo.bar.edu` , será localizado através de uma busca direta à partir da raiz dos servidores de resolução.

Este comportamento é diferente do verificado anteriormente onde a pesquisa continuaria através de `mumble.bar.edu` e `mumble.edu` . Consulte a RFC 1535 para descobrir porque isso é considerado uma prática ruím, e até mesmo uma brecha de segurança.

Uma alternativa é adicionar a linha abaixo:

```
search foo.bar.edu bar.edu
```

ao invés da linha previamente existente

```
domain foo.bar.edu
```

em seu arquivo `/etc/resolv.conf` (consulte resolv.conf(5)). Contudo verifique se a ordem de pesquisa não vai além da fronteira entre a administração pública e a local, conforme definido na RFC 1535.

P: Por que obtenho o erro, Permission denied, para todas as operações de rede?

R: Se o *kernel* do seu FreeBSD foi compilado com a opção `IPFIREWALL` , você deve compreender que a política padrão, à partir da versão 2.1.7 (atualmente alterada durante o desenvolvimento da versão 2.1-STABLE) é negar todos os pacotes que não forem explicitamente permitidos.

A seu firewall foi erroneamente configurado, de forma não intencional, a operacionalidade do sistema pode ser restaurada, simplesmente digitando o seguinte (conectado como `root`):

```
# ipfw add 65534 allow all from any to any
```

A variável `firewall_type="open"` também pode ser definida, no arquivo `/etc/rc.conf`.

Para maiores informações sobre a configuração de firewall, por gentileza, consulte a seção correspondente no Manual do FreeBSD.

P: Qual o acréscimo de sobrecarga ocasionado pelo IPFW?

R: Por gentileza, refira-se ao capítulo sobre Firewalls do Manual do FreeBSD mais especificamente, a seção sobre Overhead & Otimização do IPFW.

P: Minha regra de fwd do IPFW, que deveria redirecionar um serviço para outra estação, não está funcionando. Por que?

R: Provavelmente porque a verdadeira intenção é traduzir os pacotes que chegam na sua estação, e rescrevê-los para renviar para a outra máquina, e não simplesmente redirecionar o pacote. Normalmente o ideal é fazer NAT (tradução de endereços de rede). Uma regra de reenvio de pacotes, faz exatamente o que ela deve fazer: reenviar pacotes. As regras não alteram (rescrevem) o conteúdo ou cabeçalhos dos dados presentes no pacote. Por exemplo, digamos que a questão seja a seguinte regra:

```
01000 fwd 10.0.0.1 from any to foo 21
```

Quando um pacote destinado à estação *foo* chegar no FreeBSD que filtra essa regra, ele será encaminhado para a máquina cujo endereço IP é *10.0.0.1*, mas o endereço de destino original do pacote será mantido, ou seja, os pacotes chegando em *10.0.0.1* ainda terão a estação *foo* como destino final, marcado em seu cabeçalho TCP. O endereço de destino não é alterado (reescrito) para a máquina *10.0.0.1*, o que propicia um comportamento de verificação de checksum do cabeçalho IP. O comportamento normal é que a máquina *10.0.0.1* descarte o pacote, já que o endereço de destino do mesmo não é o endereço da estação em questão. Esse comportamento costuma confundir alguns usuários menos experientes, não correspondendo a ação com suas expectativas. Essa é uma característica do IPFW, e não um problema.

Consulte o FAQ sobre Redirecionamento de Serviços, a página de manual do natd(8), ou uma das diversas ferramentas de redirecionamento disponíveis na Coleção de Ports, para verificar a forma correta de obter o comportamento desejado.

P: Como redirecionar requisições de serviço de uma máquina para outra?

R: Serviços como FTP (e outros) podem ser redirecionados com o pacote socket, disponível na árvore de Coleção do Ports, sob a categoria "sysutils". Simplesmente substitua a linha de comando do serviço a ser redirecionado para executar a ferramenta socket, como no exemplo abaixo:

```
ftp stream tcp nowait nobody -/usr/local/bin/socket ↺
socket ftp.example.com  ftp
```

Onde ftp.foo.com e ftp são a máquina (host) e a porta de conexão que será utilizada para o redirecionamento, respectivamente.

P: Onde obtenho uma ferramenta de gerenciamento de banda?

R: Existem três ferramentas de gerenciamento de banda disponíveis para o FreeBSD. O dummynet(4), integrada ao FreeBSD (ou mais especificamente ao ipfw(4));, o ALTQ, disponível gratuitamente, e o Bandwidth Manager da Emerging Technologies, um produto comercial.

P: O que causa o erro /dev/bpf0: device not configured?

R: Você está tentando usar um programa que precisa do Berkeley Packet Filter (veja a página de manual do bpf(4) para obter maiores informações), mas ele não está compilado no *kernel*. Adicione a seguinte linha no arquivo de configuração do seu *kernel* e recompile-o:

```
pseudo-device bpf       # Berkeley Packet Filter
```

Depois que reiniciar o sistema com o novo *kernel*, basta criar a interface do dispositivo, o que pode ser feito ao executar o seguinte comando, no diretório `/dev`:

```
# sh MAKEDEV bpf0
```

Por gentileza, refira-se à seção sobre Criação de interface de Dispositivos do Manual do FreeBSD, para obter mais informações sobre o assunto.

P: Como montar o disco de uma estação Windows na minha rede, de forma semelhante ao smbmount em sistemas Linux?

R: Use o conjunto de ferramentas SMBFS. Se trata de um conjunto de modificações no *kernel*, e uma série de aplicações específicas. O programa e outras informações podem ser obtidos na Coleção de `Ports` do FreeBSD, em net/smbfs, ou no sistema base do FreeBSD a partir da versão 4.5-RELEASE.

P: O que são as mensagens sobre "icmp-response bandwidth limit 300/200 pps" em meus registros de logs?

R: É o resultado de seu *kernel* informando-o que alguma atividade esta provocando o envio de um número de respostas ICMP ou TCP reset (RST) superior ao número que o *kernel* julga adequado. Respostas ICMP são, geralmente, comportamento ocasionado pela tentativa de conexão em portas UDP não utilizadas. As respostas TCP reset são o resultado gerado pelas tentativas de conexão em portas TCP não poníveis. Entre outras causas, algumas das atividades que podem ocasionar esse tipo de mensage, são:

- Ataques de negação de serviço (DoS) por força bruta (em oposição a ataques baseados em um único pacote que visa explorar uma vulnerabilidade específica).
- Varreduras de porta (*port scans*) que visam rastrear um elevado número de portas (em oposição a ataques que tentam varrer apenas um pequeno número de portas conhecidas).

O primeiro número (valor) na mensagem indica quantos pacotes foram enviados pelo *kernel* antes do limite passar a vigorar e o segundo valor indica o limite estabelecido no *kernel*. Você pode controlar este limite através da variável `net.inet.icmp.icmplim` do sysctl com instruções como esta abaixo, onde estabelecemos um limite de 300 pacotes por segundo:

```
# sysctl --w net.inet.icmp.icmplim=300
```

Se a intenção é não registrar essas mensagens nos arquivos de registros, mas ainda assim manter a capacidade do *kernel* limitar as respostas, a variável `net.inet.icmp.icmplim_output` do sysctl pode ser usada para desabilitar o registro:

```
# sysctl --w net.inet.icmp.icmplim_output=0
```

Finalmente se a inteção é desabilitar esse comportamento por completo, basta definir a variável `net.inet.icmp.icmplim` do sysctl (conforme o exemplo acima) como 0. Desabilitar o recurso de limite de resposta é desencorajado pelas razões acima expostas.

P: Do que se trata estas mensagens de erro arp: unknown hardware address format?

R: Significa que alguma interface de rede no mesmo barramento Ethernet que você, está usando um endereço de MAC cujo formato não é reconhecido pelo FreeBSD. Provavelmente isso deve estar sendo causado por algum outro usuário, fazendo experiências com placas Ethernet em algum lugar na sua mesma rede. Em redes com Cable Modens esse comportamento é ainda mais comum; não é prejudicial e não atrapalha a performance do seu FreeBSD.

P: Acabei de instalar o CVSup, mas ao tentar usá-lo, aparecem erros. O que está havendo?

R: Primeiro, verifique se as mensagens de erro em questão são como essas:

```
/usr/libexec/ld-elf.so.1: Shared object -"libXaw.so.6" not found
```

Esse tipo de erro se deve à instalação do `port` net/cvsup em estações sem o XFree86. Se a inteção é usar a interface gráfica oferecida pelo CVSup, então instale o XFree86 imediatamente. Do contrário, se a intenção é usar o CVSup apenas por linha de comando, basta desinstalar a aplicação anterior e instalar o `port` net/cvsup-without-gui. A seção sobre CVSup do Manual do FreeBSD cobre essas questões de forma mais detalhada.

Capítulo 13. Segurança

P: O BIND (`named`) está escutando na porta 53 e em outras portas elevadas. O que está havendo?

R: O FreeBSD, a partir da versão 3.0, utiliza portas não privilegiadas e elevadas, aleatoriamente, para responder à requisições de DNS. Se a intenção é usar a porta 53 para responder a estas requisições, para adequar o comportamento do BIND à um *firewall* ou apenas para sentir-se melhor, experimente acrescentar a instrução abaixo no arquivo `/etc/namedb/named.conf` :

```
options {
            query-source address * port 53;
            -};
```

O * deve ser substituído por um endereço IP único, caso se deseje restringir ainda mais este comportamento.

De qualquer forma, parabéns. É uma pratica saudável verificar registros não usuais no conteúdo de saída do sockstat(1)

P: O Sendmail está ouvindo na porta 587 além da tradicional porta 25! O que está havendo?

R: As versões mais novas do Sendmail tem suporte à uma característica que se chama *Mail Submission*, que ouve na porta 587. Esse serviço não é completamente suportado ainda, mas sua popularidade vem crescendo.

P: O que é a conta do usuário `toor` que tem UID 0? Meu sistema foi comprometido?

R: Não se preocupe. O usuário `toor` é uma conta "alternativa" com poderes de super usuário (toor é root escrito ao contrário). Normalmente esse usuário era criado quando o interpretador de comandos bash(1) era instalado, mas agora o usuário existe por padrão no sistema. A intenção é que o usuário seja usado com um interpretador de comandos fora do padrão, de forma que o ambiente de linha de comando do usuário `root` não tenha que ser alterada. Essa é uma situação importante quando nos referimos à interpretadores de comandos que não fazem parte da base do sistema operacional (por exemplo, que foram instaladas do `Ports` ou como pacotes, já que normalmente, elas são instaladas sob o `/usr/local/bin` que por padrão está em um sistema de arquivos diferente da raiz do sistema. Em uma situação onde o interpretador de comandos do usuário estiver sob `/usr/local/bin` ou sob `/usr` (ou onde quer que seja) e esse sistema de arquivos não puder ser montado por alguma razão, o usuário `root` estará impossibilitado de se logar no sistema para corrigir o problema (contudo, ao entrar em modo mono tarefa, o sistema pede o caminho completo para algum interpretador de comandos.

Alguns administradores costumam usar o `toor` para tarefas do dia-a-dia, com um interpretador de comandos (*shell*) não comum, e deixando o `root` com seu

interpretador de comandos (*shell*) padrão para realizar tarefas de emergência ou de modo mono usuário. Por padrão, o usuário `toor` não pode ser usado, já que ele não tem uma senha definida. Para habilitar a conta, logue-se como `root` no sistema e defina uma senha para o `toor`.

P: Por que o `suidperl` não está funcionando corretamente?

R: Por motivos de segurança, a instalação padrão do `suidperl` não tem o bit suid definido. Os administradores de sistemas podem reaver o comportamento esperado com o seguinte comando:

```
# chmod u+s -/usr/bin/suidperl
```

Se a intenção é que o `suidperl` seja compilado com suid durante as atualizações do sistema, edite o `/etc/make.conf` e adicione a linha `ENABLE_SUIDPERL=true` no arquivo, antes de começar um `make buildworld`.

Capítulo 14. PPP

P: Não consigo fazer meu ppp(8) funcionar. O que estou fazendo de errado?

R: Deve-se primeiro ler a man page do ppp(8) e seção PPP do Manual do FreeBSD. Habilite os logs com o comando

```
set log Phase Chat Connect Carrier lcp ipcp ccp command
```

Este comando pode ser digitado no prompt do ppp(8) ou pode ser colocado no `/etc/ppp/ppp.conf` (A seção `default` no início do arquivo é o melhor lugar para colocar isso). Tenha certeza de que seu`/etc/syslog.conf` (veja syslog.conf(5)) tenha as linhas

```
!ppp *.*/var/log/ppp.log
```

e que o arquivo `/var/log/ppp.log` exista. Agora pode-se ver o que está acontecendo analisando seu arquivo de log. Não se preocupe se isso não faz sentido mas se precisar de ajuda, esta informação fará sentido a eles.

Se a sua versão do ppp(8) não suporta o comando set log ,deve-se fazer o download da versão mais recente. O FreeBSD 2.1.5 (e posteriores) suporta a compilação do código mais recente.

P: Por que o ppp(8) fica pendurado quando eu o inicializo?

R: Isso normalmente acontece porque seu hostname não esta sendo resolvido. A melhor maneira de corrigir isso é certificar-se de que o `/etc/hosts` está sendo consultado pelo resolvedor; editando primeiro o `/etc/host.conf` e colocando `hosts` na primeira linha. Para isso, simplesmente adicione no `/etc/hosts/` uma entrada para sua máquina local. Se não tem nenhuma rede local, mude a linha do `localhost`:

```
127.0.0.1       foo.example.com foo localhost
```

Senão, adicione uma outra entrada para seu host. Consulte as man pages relevantes para maiores detalhes.

Ao terminar, deve ser possível dar um `ping -c1 `nomedohost`` com sucesso.

P: Porque o ppp(8) não disca em modo`-auto`

R: Primeiro verifique se há uma rota padrão. Ao executar um `netstat -rn` (veja netstat(1)), devem aparecer duas entradas como essas:

```
Destination        Gateway            Flags     Refs     ↺
Use     Netif Expire
default            10.0.0.2           UGSc        0       ↺
0      tun0
```

```
10.0.0.2           10.0.0.1           UH          0        ↺
0      tun0
```

Assume-se o uso do endereço recomendado pelo Manual do FreeBSD, páginas de manual ou no arquivo ppp.conf.sample. Se a rota padrão não foi definida, é possível que a versão do ppp(8) seja antiga, uma vez que não entendem a palavra `HISADDR` no arquivo ppp.conf. Se a versão do seu FreeBSD for anterior a 2.2.5, mude a linha

```
add 0 0 HISADDR
```

para

```
add 0 0 10.0.0.2
```

Outra razão para a rota padrão estar ausente, pode ser porquê você equivocadamente definiu uma rota default em seu arquivo `/etc/rc.conf` (veja rc.conf(5)) (este arquivo era chamado `/etc/sysconfig` nas versões anteriores a 2.2.2) e omitiu a linha

```
delete ALL
```

no `ppp.conf`. Se este for o problema, volte para seção de configurações Finais do Manual do FreeBSD.

P: O que No route to host significa?

R: Este erro é causado pela falta das linhas

```
MYADDR:
  delete ALL
  add 0 0 HISADDR
```

no seu arquivo `/etc/ppp/ppp.linkup`. Isso somente é necessário se você tem um IP dinâmico ou não sabe o endereço do seu gateway. Se você esta usando o modo interativo, pode-se digitar o seguinte, depois de ter entrado no `modo packet` (O modo packet é indicado pelo PPP maiísculo no prompt):

```
delete ALL
add 0 0 HISADDR
```

Consulte a seção PPP e Endereços IPs dinâmicos do Manual do FreeBSD para maiores detalhes

P: Porque minha conexão cai depois de (em média) 3 minutos?

R: O default timeout do PPP é de 3 minutos. Isso pode ser ajustado com a linha

```
set timeout NNN
```

Onde NNN é o número em segundos de inatividade antes da conexão ser fechada. Se *NNN* é zero a conexão nunca será fechada devido a um timeout. É possível colocar

esse comando no `ppp.conf` ou digitá-lo no modo interativo. Também é possível que isso seja ajustado enquanto sua conexão esta ativa conectando pelo socket do servidor ppp usando telnet(1) ou pppctl(8). Consulte a man page do ppp(8) para maiores detalhes.

P: Por que minha conexão cai quando está sobrecarregada?

R: Se você tem o relatório da qualidade da ligação (opção LPR) configurado é possível que muitos pacotes LQR entre sua máquina e a origem estejam sendo perdidos. O ppp deduz que sua linha deve ser ruim e desconecta. Antes da versão 2.2.5 do FreeBSD, o LQR era habilitado por default. Agora ele é desabilitado por default. O LQR pode ser desabilitado com a linha

```
disable lqr
```

P: Por que minha conexão cai após um período aleatório de tempo?

R: Às vezes, em uma linha telefônica com ruídos, ou quando a linha tem espera de ligações, seu modem pode desligar, pensando (incorretamente) que perdeu o sinal de linha.

Existem ajustes em grande parte dos modems determinando o quão tolerante ele deve ser a respeito de perdas de sinal de linha. No USR Sportster, por exemplo, isso é medido pelo registrador S10 em décimos de segundo. Para fazer com que seu modem não caia, poderia-se adicionar a seguinte string dial up para envio/espera:

```
set dial -"...... ATS10=10OK......"
```

Consulte o manual do seu modem para detalhes.

P: Por que minha conexão fica pendurada depois de um período aleatório de tempo?

R: Muitas pessoas experimentam conexões presas sem aparentemente nenhuma explicação.

Se você usa um modem externo, você simplesmente pode tentar um ping(8) para ver se a luz TD fica piscando quando você transmite dados. Se piscar (e a luz do RD não), o problema é com a extremidade remota. Se o TD não piscar, o problema é local. Com um modem interno você precisará usar o comando `set server` no seu arquivo ppp.conf. Quando ocorrer de cair, conecte com o ppp(8) usando o pppctl(8). Se a conexão de sua rede voltar de repente (o ppp voltou devido a atividade no diagnóstico do socket) ou se você não consegue conectar (assumindo que o comando `set socket` foi iniciado com sucesso) o problema é local. Se você puder conectar e ainda as coisas estiverem "penduradas" habilite os logs do async local com o comando `set log local async` e use o ping(8) de outra janela ou terminal para forçar a atividade da conexão. Os logs async irão mostrar os dados que estão sendo transmitidos e recebidos durante a ligação. Se os dados estão indo e não estão voltando, o problema é remoto.

Tendo estabelecido que o problema é local ou remoto, você tem agora duas possibilidades:

- Se o problema é remoto, leia a secção P:.
- Se o problema for remoto, leia a secção P:.

P: A extremidade remota não está respondendo. O que eu posso fazer?

R: Há muito pouco que pode ser feito em relação a isso. A maioria dos provedores irão recusar ajuda-lo se você não usar o Windows. Você pode habilitar a `lqr` no seu arquivo `ppp.conf` , permitindo ao ppp(8) detectar a falha remota e desligar-se, mas essa detecção é relativamente lenta e consequentemente inútil. Pode-se querer evitar dizer ao seu provedor que você está rodando o user-ppp....

Primeiro tente desabilitar toda compressão local adicionando o seguinte em sua configuração:

```
disable pred1 deflate deflate24 protocomp acfcomp shortseq vj
deny pred1 deflate deflate24 protocomp acfcomp shortseq vj
```

Então reconecte para assegurar de que isso não fez diferença. Se as coisas melhoraram ou se o problema foi resolvido completamente, determine quais ajustes fizeram a diferença através de tentativas e erros. Isto fornecerá uma boa cartada quando você contactar seu provedor (embora possa parecer que você não esteja rodando um produto Microsoft).

Antes de contactar seu provedor, habilite o log async localmente e aguarde até que sua conexão caia novamente. Isto pode usar um bocado de espaço em disco. A última leitura de dados da porta pode ser de seu interesse. São geramente dados em ascii, e podem mesmo descrever o problema (“Memory fault, core dumped”?).

Se seu provedor for prestativo, ele deve ser capaz de habilitar o log da extremidade da conexão dele, então quando a proxima queda de link ocorrer, eles podem ser capazes de dizer porque o seu lado esta tendo problemas. Sinta-se livre para enviar os detalhes para Brian Somers ou peça para seu provedor contactá-lo diretamente.

P: O ppp(8) está travado. O que eu posso fazer?

R: A melhor coisa a fazer aqui é recompilar o ppp(8) adicionando `CFLAGS+=-g` e `STRIP=` no final do Makefile, depois faça `make clean && make && make install` . Quando o ppp(8) ficar travado, procure o id do processo com um `ps ajxww | fgrep ppp` e execute `gdb ppp PID`. No prompt do gdb você pode então usar o `bt` para obter um rastreamento da pilha.

Envie os resultados para: <brian@Awfulhak.org >.

P: Por que não acontece nada depois da mensagem “Login OK!”?

R: Antes da versão 2.2.5 do FreeBSD, uma vez que a conexão foi estabelecida, o ppp(8) espera que o modem remoto inicie o protocolo do controle de linha (LCP), Muitos provedores não iniciarão a negociação e esperarção que o cliente a faça. Para forçar o ppp(8) para iniciar o LCP, use a seguinte linha:

```
set openmode active
```

Nota

Nota: isto geralmente não prejudica em nada se a negociação for iniciada por ambos os lados, assim, a opção openmode é agora ativada por padrão. Entretanto, na próxima seção será explicado quando isso realmente *proporciona* algum problema.

P: Eu vejo os mesmos erros sobre `magic is the same .` O que significa?

R: Ocasionalmente, depois da conexão, você pode ver mensagens no log dizendo “"magic is the same"”. Às vezes essas mensagens são sem importância e as vezes um lado ou o outro sai. A maioria das implementações do ppp não consegue continuar com esse problema, e mesmo se a conexão estiver para ser estabelecida, você verá repetidas requisições de configuração e reconhecimentos de configuração no arquivo de log até que o ppp(8) eventualmente desista e feche a conexão.

Isto normalmente acontece em máquinas com discos lentos que estão dando spawning getty na porta e executam o ppp(8) através de um login script ou programa depois do login. Eu também ouvi relatórios disso estar acontecendo consistentemente ao usar slirp. A razão é que no tempo entre saída do getty(8) e a inicialização do ppp(8), o ppp(8) do lado cliente começa a emitir pacotes do protocolo do controle da linha LCP). Porque o ECHO ainda esta ligado na porta do servidor, o cliente então vê esses pacotes serem “refletidos” de volta.

Uma parte da negociação LCP deve estabelecer um número mágico para cada lado da ligação de modo que as “reflexões” possam ser detectadas. O protocolo diz que quando um ponto tenta negociar o mesmo número mágico, um NAK deve ser emitido e um novo número mágico deve ser escolhido. Durante o período que a porta do server tem o ECHO ligado, o ppp(8) cliente manda pacotes LCP, vê o mesmo número mágico nos pacotes refletidos e manda NAKs a ele. Ve-se também o NAK que foi refletido (que significa que o ppp(8) deve mudar o seu número mágico). Isto produz um número potencialmente enorme de mudanças de números mágicos que estão sendo empilhados no buffer do tty do servidor. Tão logo o ppp(8) se inicia no servidor, ele é inundado com mudanças de números mágicos imediatamente ao negociar o LCP, e desiste. Enquanto isso, o cliente não vê por muito tempo as reflexões e fica feliz apenas por algum tempo, até receber a desconexão do servidor.

Isto pode ser evitado permitindo que a negociação seja iniciada pelo servidor (peer), adicionando a seguinte linha no seu arquivo ppp.conf:

```
set openmode passive
```

Isso diz ao ppp(8) para esperar que o servidor inicie as negociações LCP. Alguns servidores, entretanto, podem nunca iniciar as negociações. Se este for o caso, você deve fazer algo como:

```
set openmode active 3
```

Isto diz ao ppp(8) para ser passivo durante 3 segundos, e então começar a enviar requisições LCP. Se o servidor (peer) começar a enviar requisições durante esse período, o ppp(8) irá responder imediatamente, ao invés de aguardar pelo período completo de 3 segundos.

P: As negociações LCP continuam até que a conexão seja fechada. O que está errado?

R: Há atualmente uma característica faltando no ppp onde ele não associa respostas LCP , CCP & IPCP com suas requisições originais. Como consequência, se uma implementação ppp é 6 segundos mais lenta do que o outro lado, esse lado emitirá duas requisões adicionais de configuração LCP. Isto é fatal.

Considere duas implementações, A e B. A emite requisições LCP imediatamente após a conexão e B leva 7 segundos para iniciar. Quando B inicia, A emitiu 3 LCP REQs. Nós estamos supondo que a linha ECHO esteja desabilitada, senão nós veríamos problemas com número mágico como descritos na seção anterior. B emite um REQ, entao um ACK para o primeiro dos REQs de A. Isto resulta em A entrando no estado OPENED e enviando um ACK (o primeiro) de volta a B. Enquanto isso, B envia de volta mais dois ACKs em resposta aos dois REQs adicionais enviados por A antes de B ter iniciado. B então recebe o primeiro ACK de A e também entra no estado de OPENED. A recebe o segundo ACK de B e retorna ao estado de REQ-SENT, enviando um outro (seguinte) REQ conforme a RFC ordena. Então, recebe o terceiro ACK e entra no estado OPENED. Enquanto isso, B recebe REQ (posterior) de A, tendo como resultado uma reversão para o estado ACK-SENT e enviando um outro (segundo) REQ e (depois) ACK conforme a RFC. Conseguindo o REQ, A entra em REQ-SENT e envia outro REQ. Imediatamente recebe o ACK seguinte e entra em OPENED.

Isto continuará até que um dos lados descubra que eles não estão indo a lugar algum e desista.

A melhor maneira de evitar isso é configurar um lado para ser `passivo` - isso faz com que um lado espere pelo outro para iniciar uma negociação. Isto pode ser feito com o comando

```
set openmode passive
```

Deve-se ter cuidado com esta opção. Você também deve usar o comando

```
set stopped N
```

para limitar a quantidade de tempo que o ppp(8) esperará pelo outro lado iniciar a negociação. Alternativamente o comando

```
set openmode active N
```

(onde *N* é o número de segundos para esperar antes de iniciar as negociações) pode ser usado. Consulte as páginas de manual para detalhes.

P: Por que o ppp(8) trava logo após a conexão?

R: Antes da versão 2.2.5 do FreeBSD, era possível que sua ligação fosse desabilitada logo após a conexão, devido a incapacidade de negociação da compressão Predictor1 do ppp(8). Isto aconteceria somente se ambos os lados tentassem negociar protocolos diferentes do controle da compressão (CCP). Este problema atualmente esta corrigido, mas se você ainda roda uma versão antiga do ppp(8), o problema pode ser resolvido com a linha:

```
disable pred1
```

P: Por que o ppp(8) trava quando executo comandos shell pelo ppp(8)?

R: Quando executa-se o `shell` com o comando `!` , o ppp(8) executa um shell (ou se você passar quaisquer argumentos, o ppp(8) executará aqueles argumentos). Ppp aguardará o comando terminar antes de continuar. Se você tentar usar a sua ligaço ppp enquanto é executado o comando, a ligação aparacerá congelada. Isto porque o ppp estará esperando pelo finalização da comando.

Se você desejar executar comandos assim, use o comando `!bg`. Ele executará o comando em background e o ppp(8) poderá continuar a servir a conexão normalmente.

P: Como uma conexão por ppp(8) usando um nunca termina?

R: Não há nenhuma maneira para que o ppp(8) determine automaticamente que uma conexão direta caiu. Isto é devido às linhas que são usadas na série de cable null modems. Ao usar esse tipo de conexão o LQR deve sempre ser habilitado com a linha:

```
enable lqr
```

A LQR é aceita por padrão, se negociada com o outro lado.

P: Por que o ppp(8) disca sem nenhuma razão no modo -auto?

R: Se o ppp(8) esta discando inesperadamente, você deve determinar a causa, e setar os filtros de discagem (dfilters) para evitar esse comportamento.

Para determinar a causa, use a seguinte linha:

```
set log +tcp/ip
```

Isto registrará todo o tráfego através da conexão. A próxima vez que a linha acima for ativada, você verá a razão logada.

Você pode agora desabilitar a discagem sob estas circunstâncias. Geralmente, este tipo de problema ocorre devido aos lookups do DNS. Para evitar que os lookups do DNS estabeleçam uma conexão (isto *não* impedirá que o ppp(8) passe os pacotes através de uma conexão já estabelecida) use o seguinte:

```
set dfilter 1 deny udp src eq 53
set dfilter 2 deny udp dst eq 53
set dfilter 3 permit 0/0 0/0
```

Nem sempre isso é apropriado, porque irá interromper suas capacidades de discagem por demanda - a maioria dos programas precisarão de lookup do DNS antes de fazer alguma outra coisa relacionada a rede.

No caso do DNS, você deve tentar determinar o que esta realmente tentando resolver um hostname. Na maioria das vezes o sendmail(8) é o culpado. Você deve certificar-se que o sendmail não deve fazer nenhum DNS lookup em seu arquivo de configuração. Veja na seção Configuração de Mail para detalhes de como criar seu próprio arquivo de configuração e o que deve conter nele. Você pode também adicionar a seguinte linha ao seu arquivo `.mc`:

```
define(`confDELIVERY_MODE', `d')dnl
```

Isto fará o sendmail enfileirar tudo, até que a fila esteja funcionando (normalmente o sendmail é invocado com `-bd -q30m`, dizendo a ele para rodar a fila (queue) a cada 30minutos) ou até que o `sendmail -q` seja feito (talvez do arquivo ppp.linkup)

P: O que estes erros de CCP significam?

R: Eu estou vendo os seguintes erros em meus logs:

```
CCP: CcpSendConfigReq
CCP: Received Terminate Ack (1) state = Req-Sent (6)
```

Isto é porque o ppp(8) esta tentando negociar a compressão Predictor1, e o outro ponto não quer negociar nenhuma compressão. As mensagens são inofensivas, mas se você desejar remove-las, poderá desabilitar a compressão Predictor1 também localmente:

```
disable pred1
```

P: Por que o ppp(8) trava durante a transferência de arquivo com erros de IO(Input/Output)?

R: No FreeBSD 2.2.2 e anteriores, havia um bug no driver tun, que impedia pacotes de entrada de um tamanho maior do que o MTU da interface. O O recebimento de pacotes maiores que o MTU da interface resulta em erro de IO que é logado através do syslogd.

A especificação do ppp diz que um MTU de 1500 deve ser *sempre* aceito como mínimo. Apesar de todas as negociações de LCP, é possível diminuir o MTU para menos de 1500, independentemente disso, seu provedor irá transmitir pacotes de 1500 para você, e isso travará sua conexão.

O problema pode ser contornado por nunca ajustar um MTU menor que 1500 sobre o FreeBSD 2.2.2 ou anteriores.

P: Por que o ppp(8) não loga minha velocidade de conexão?

R: A fim de logar todas as linhas de "conversação" de seu modem você deve habilitar o seguinte:

```
set log +connect
```

Isto irá fazer o ppp(8) logar tudo, até a ultima requisição "expect".

Se você desejar ver sua velocidade de conexão e estiver usando PAP ou CHAP (e consequentemente não tenha qualquer coisa para o "chat" depois do CONNECT no script de discagem - no `set login` script), você deve certificar-se que instruiu o ppp(8) a ``a esperar" a linha inteira CONNECT, algo como:

```
set dial -"ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 4 \
  \"\" ATZ OK-ATZ-OK ATDT\\T TIMEOUT 60 CONNECT \\c \\n"
```

Aqui nosso CONNECT, não envia nada, ele espera então um line-feed, forçando o ppp(8) a ler toda a resposta do CONNECT.

P: Por que o ppp(8) ignora o caracter \ no meu script de conversação?

R: O Ppp analiza cada linha no seu arquivo de configuração, então isso pode ser interpretado como string tal como `set phone "123 456 789"` corretamente e compreende de fato que o número é um *único* argumento. A fim de especificar um caracter", você deve escapar desse comportamento, usando um a barra invertida \.

Quando o script de conversação analizar cada argumento, ele reinterpretará o argumento a fim de encontrar quaisquer sequências especiais como \P ou \T (veja a man page). Como resultado desta dupla análise, você deve conseguir usar o número correto de espaços.

Se você deseja enviar um caracter \ para dizer a seu modem, você precisará de algo assim:

```
set dial -"\"\" ATZ OK-ATZ-OK AT\\\\X OK"
```

resultando a seguinte sequência:

```
ATZ
OK
AT\X
OK
```

ou

```
set phone 1234567
set dial -"\"\" ATZ OK ATDT\\T"
```

resultando na seguindo sequência:

```
ATZ
OK
ATDT1234567
```

P: Por que o ppp(8) dá um seg-fault, mas eu não vejo nenhum arquivo `ppp.core` ?

R: O Ppp (ou qualquer programa desta natureza) nunca dever dar core dump. Porque o ppp(8) roda com um e-userid (effective user id) `0`, o sistema operacional não irá escrever a imagem do core do ppp(8) em disco antes de terminá-lo. Mas se acontecer de o ppp(8) terminar devido a uma falha de segmentação ou algum outro sinal que normalmente causaria um core dumped, *e* você tem certeza de que esta usando uma versão mais recente do ppp (veja o começo desta seção), você deve fazer o seguinte:

```
% tar xfz ppp-*.src.tar.gz
% cd ppp*/ppp
% echo STRIP= >>Makefile
% echo CFLAGS+=-g >>Makefile
% make clean all
% su
# make install
# chmod 555 -/usr/sbin/ppp
```

Você agora tem uma versão debugável do ppp(8) instalada. Você precisará ser `root` para rodar o ppp porque todos os seus privilégios foram revogados. Quando você iniciar o ppp(8), tome nota com cuidado de qual era o seu diretório corrente naquele instante.

Agora, quando o ppp(8) receber uma violação de segmento (seg-fault), você terá um arquivo chamado ppp.core. Você deve então fazer o seguinte:

```
% su
# gdb -/usr/sbin/ppp ppp.core
(gdb) bt
.......
(gdb) f 0
......
(gdb) i args
```

```
......
(gdb) l
.......
```

Toda essa informação deve ser dada com a sua pergunta, sendo possível agora diagnosticar o problema.

Se você é familiarizado com o gdb, você pode desejar encontrar alguns outros bits e partes como as que causaram o dump e também os endereços e valores das variáveis revelantes.

P: Por que o processo que força a discagem em auto mode nunca conecta?

R: Este era um problema conhecido na configuração ppp(8) para negociar um IP local dinâmico com o outro ponto no auto mode. Isto foi corrigido na versão mais recente. Procure na man page do ppp(8) por `iface`.

O problema era que quando este programa inicial chamava o connect(2), o número IP da interface tun estava atribuído ao endpoint do soquete. O *kernel* cria o primeiro pacote de saída e escreve-o no dispositivo tun. O ppp(8) então lê o pacote e estabelece a conexão. Se em consequência da atribuição dinâmica do IP do ppp(8) o endereço da interface for mudado, o endpoint do soquete original será inválido. Todos os pacotes subsequentes emitidos ao outro ponto serão geralmente descartados. Mesmo se não forem descartados, nenhuma das respostas irá voltar pela rota da máquina de origem, isto porque o número IP já não pertence a essa máquina.

Há diversas maneiras teóricas para abordargem desse problema. Seria mais agradável se o ponto reatribuísse, se possível o mesmo número IP `:-)` A versão atual do ppp(8) faz isso, mas a maioria das outras implementações não.

O método mais fácil do nosso lado, seria nunca mudar o número IP da interface tun, mas ao invés disso, mudar todos os pacotes de saída de modo que a origem do número IP é mudada da interface IP para o IP negociado dinâmicamente. Isto é essencialmente o que a opção `iface-alias` na versão mais recente do ppp faz (com a ajuda da libalias(3) e da opção `-nat` do ppp(8)) - esta mantendo endereços anteriores da interface e fazendo NAT do último endereço negociado.

Uma outra alternativa (e provavelmente a mais confiável) seria implementar uma chamada de sistema que mudasse todos os soquetes ligados de um IP para outro. O ppp(8) usaria essa chamada para modificar os soquetes de todos os programas em execução quando um novo endereço IP é negociado. O mesmo sistema de chamadas poderia ser usado por clientes dhcp quando são forçados a religar seus soquetes.

Ainda, outra possibilidade é permitir a interface para ser ativada sem um número IP. Os pacotes de saída seriam dados um número IP 255.255.255.255 até que a primeira SIOCAIFADDR ioctl esteja pronta. Isto resultaria na completa ligação com o soquete. Seria até o ppp(8) mudar o número IP de origem, mas somente se foi setado para 255.255.255.255, e somente o número IP e o IP checksum deveriam ser mudados. Isto

porém, é um pequeno hack do *kernel* que deve estar enviando maus pacotes para uma interface configurada, na suposição de que algum outro mecanismo é capaz de corrigir as coisas de forma restrospectiva.

P: Por que a maioria dos jogos não funciona com a opção -nat?

R: A razão para os jogos e outros programas não funcionarem quando a libalias esta em uso é porque a máquina de fora irá tentar abrir uma conexão ou enviar pacotes UDP (não solicitados) para a máquina de dentro. O software NAT não sabe que deve enviar esses pacotes para a máquina interna.

Para que as coisas funcionem, certifique-se de que a única coisa que esta rodando é o software que você esta tendo problemas, a seguir rode o tcpdump na interface tun do gateway ou habilite ppp(8) tcp/ip logging (`set log+tcp/ip`) na gateway.

Quando você iniciar o software, você deve ver pacotes passando através da máquina gateway. Quando alguma coisa volta vindo de fora, será descartado (este é o problema). Tome nota do número da porta desses pacotes e a seguir feche o software. Faça isso algumas vezes para ver se os números da porta são consistentes. Se eles forem, a seguinte linha no `/etc/ppp/ppp.conf` fará o software funcional:

```
nat port proto máquinainterna :
porta porta
```

Onde *`proto`* é ou `tcp` ou `udp`, *`máquinainterna`* é a máquina de onde você quer que os pacotes sejam enviados e *`porta`* é número da porta de destino dos pacotes.

Você não poderá usar o software em outras máquinas sem mudar o comando acima, e rodar o software em duas máquinas internas ao mesmo tempo é fora de questão - Apesar de tudo, o lado de fora esta vendo toda sua rede interna como sendo somente uma máquina.

Se os números da porta não são consistentes, há ainda mais 3 opções.

1. Enviar o suporte na libalias. Exemplos de 'casos especiais' podem ser encontrados em `/usr/src/lib/libalias/alias_*.c` (`alias_ftp.c` eh um bom tipo de protocolo). Isto geralmente envolve ler determinados pacotes reconhecidos na saída, identificando a instrução que chama a máquina externa para iniciar a conexão de volta para a máquina interna em uma porta (aleatória) específica e setar a “rota” na tabela de aliases de modo que os pacotes subsequentes saibam para onde ir.

   Esta solução é a mais difícil, mas é a melhor e irá fazer o software trabalhar com múltipla máquinas.

2. Use um proxy. A aplicação poderá suportar sock5 por exemplo, ou (como no caso do “cvsup”) pode ter uma opção “passive” que evita sempre requisições feitas pelo outro ponto de volta para a máquina local.

3. Redirecione tudo para a máquina interna usando `nat addr`. Pode-se dizer que essa seja a apelação.

P: Alguém fez uma lista de número de portas úteis?

R: Não ainda, mas a intensão é produzir tal lista (se algum interesse for mostrado). Em cada exemplo, *`internal`* deve ser substítuido pelo IP da máquina que esta jogando o jogo.

- Asheron's Call

  ```
  nat port udp internal:65000 65000
  ```

  Mude manualmente o número da porta dentro do jogo para 65000. Se você começar com um determinado número de máquinas que você deseja jogar atribua uma porta para cada (por ex 65001, 65002, etc) e adicione uma `porta nat` para cada uma.

- Half Life

  ```
  nat port udp internal:27005 27015
  ```

- PCAnywhere 8.0

  ```
  nat port udp internal:5632 5632
  ```

  ```
  nat port tcp internal:5631 5631
  ```

- Quake

  ```
  nat port udp internal:6112 6112
  ```

  Alternativamente, você pode querer ir em www.batle.net para dar uma olhada no suporte de proxy do quake.

- Quake 2

  ```
  nat port udp internal:27901 27901
  ```

  ```
  nat port udp internal:60021 60021
  ```

  ```
  nat port udp internal:60040 60040
  ```

- Red Alert

  ```
  nat port udp internal:8675 8675
  ```

  ```
  nat port udp internal:5009 5009
  ```

P: O que são erros FCS?

R: FCS significa Frame Check Sequence. Cada pacote do ppp tem um checksum anexado para assegurar-se de que os dados que estão sendo recebidos sejam os dados que estão sendo emitidos. Se o FCS de um pacote de entrada estiver incorreto, o pacote sera perdido e a contagem do HDLC FCS é aumentada. Os valores de erro HDLC podem ser mostrados usando o comando `show hdlc`.

Se a sua ligação é ruim ou se o driver serial esta perdendo pacotes), você irá ver ocasionalmente erros FCS. Isto geralmente não é motivo para se preocupar, embora diminua substancialmente os protocolos de compressão. Se você tem um modem externo, certifique-se que seu cabo esteja protegido corretamente de interferências - Isso pode erradicar o problema.

Se sua ligação congelar assim que você conectar e vier um grande número de erros FCS, pode ser porque seu link não esta com o bit 8 limpo. Certifique-se que seu modem não esteja usando o controle de fluxo do software (XON/XOFF). Se o seu datalink deve usar software de controle de fluxo, use o comando `set accmap 0x000a0000` para dizer ao ppp para ignorar os caracteres `^Q` e `^S`.

Uma outra razão para estar vendo muitos erros FCS pode ser que a extremidade remota parou de comunicar com o PPP. Você pode querer habilitar registros `async` neste ponto para determinar se os dados entrantes são realmente um alerta de início de sessão do prompt da shell. Se você tiver um prompt shell na extremidade remota, é possível terminar o ppp(8) sem deixar cair a linha usando o comando `close lcp` (o comando `term` irá reconectar você shell da máquina remota). Se nada em seus logs indicar o porque de sua ligação ter sido terminada, você pode perguntar ao administrador remoto (do seu provedor?) porque a sessão foi terminada.

P: Por que a conexão do MacOS e Windows98 travam ao rodar o PPPoE no gateway?

R: Agradecimentos a Michael Wozniak <mwozniak@netcom.ca > por descobrir o problema, e a Dan Flemming <danflemming@mac.com > pela solução do Mac:

Isto é devido ao que é chamado de roteador “Buraco Negro” (*Black Hole*). MacOS e Windows98 (e talvez outros SO's da Microsoft) envia m pacotes TCP com um tamanho de segmento requisitado muito grande para ser contido em um frame do PPPoE (MTU por default na ethernet é de 1500) *e* tenha o “não fragmento” do bit ajustado (default do TCP) e o roteador Telco não esta enviando ICMP “deve ser fragmentado” de volta ao sitio www que você esta tentando carregar. (Alternativamente o roteador está enviando pacotes ICMP corretamente, mas o firewall no sitio www esta deixando perdê-los). Quando o servidor www esta enviando seus frames que não cabem no pipe do PPPoE o roteador Telco deixa-os perder e sua página não é carregada (algumas páginas gráficas carregam porque são menores que um MSS). Esta parece ser a configuração default da maioria dos Telco PPPoE (somente eles sabem como programar o roteador).

Um maneira de fixar isso é usando o regedit em sua seu Windows 95/98 e adicionar a seguinte entrada de registro:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class
\NetTrans\0000\MaxMTU
```

Deve ser uma string com um valor "1436", porque há relatos de que alguns roteadores ADSL são incapazes de tratar os pacotes maiores que esse. Esta chave de registro foi mudada para `Tcpip\Parameters\Interfaces\ID-para-o-adptador \MTU` no windows 2000 e tornou-se um DWORD.

Consulte os documentos da Microsoft Q158474 - Windows TCPIP Entradas de registroi e Q120642 - TCPIP & NBT Parametros de configuração para Windows NT para maiores informacoes sobre alterações de MTU no Windows para funcionar com um roteador NAT.

Uma outra possibilidade do regedit sob o Windows 2000 é setar `Tcpip\Parameters \Interfaces\ID-para-o-adaptador\EnablePMTUBHDetect` DWORD para 1 como mencionado no documento original da Microsoft 120642 comentado acima.

Infelizmente o MacOS não oferece uma mudança TCP/IP nas configurações da interface. Entretanto, há um software comercial disponível, o OTAdvancedTuner (OT para OpenTransport, a pilha TCP/IP do MacOS) feito pela Sustainable Softworks, ele permite aos usuários customizar as configurações TCP/IP. Os usuários de NAT do MacOS devem selecionar o `ip_interface_MTU` no menu drop-down, colocar `1450` em vez de `1500`, clique na caixa próximo ao `Save as Auto Configure` , e clique em `Make Active` .

Versões mais recentes do ppp(8) (2.3 ou mais recente) tem o comando `enable tcpmssfixup` que irá automaticamente ajustar um valor apropriado ao MSS. Esta facilidade é habilitada por default. Se você for apaixonado pela versão mais antiga do ppp(8) você pode querer dar uma olhada no `port` do tcpmssd.

P: Nada disso me ajudou - Eu estou desesperado! O que eu posso fazer?

R: Se tudo falhar, envie o máximo de informacões que você puder, incluindo seus arquivos de configuração, a forma como está iniciando o ppp(8), os trechos relevantes de seu arquivo de log e a saída do comando `netstat -rn` (antes e depois de conectado) para a lista lista de discussão para perguntas gerais sobre o FreeBSD ou para o grupo de notícias comp.unix.bsd.freebsd.misc. Alguém deve ajudar a solucionar seu problema.

Capítulo 15. Comunicações Seriais

Essa seção cobre as perguntas mais comuns sobre comunicação serial com o FreeBSD. PPP e SLIP são abordados na seção Capítulo 12, *Redes*.

P: Como eu posso dizer se o FreeBSD encontrou minhas portas seriais?

R: Assim que o kernel do FreeBSD é carregado, ele irá varrer as portas seriais do seu sistema procurando dispositivos nas portas configuradas no kernel. Pode-se observar atentamente as mensagens que o sistema exibe, ou então executar o seguinte comando:

```
% dmesg -| grep sio
```

assim que o sistema estiver em funcionamento e execução.

Aqui estão alguns exemplos dos resultados do comando executado acima:

```
sio0 at 0x3f8-0x3ff irq 4 on isa
sio0: type 16550A
sio1 at 0x2f8-0x2ff irq 3 on isa
sio1: type 16550A
```

Eles mostram duas portas seriais. A primeira está na irq 4, está usando o endereço de porta `0x3f8`, e tem um chip modelo UART 16550A. O segundo utiliza o mesmo tipo de chip mas está na irq 3 e seu endereço de porta é `0x2f8`. Modems internos são tratados como se fossem portas seriais -- exceto que sempre tem um modem "conectado" à porta.

O *kernel* `GENERIC` tem suporte para duas portas seriais utilizando os mesmos irqs e configurações de endereços de portas do exemplo acima. Se tais configurações não estão certas para seu sistema, ou se você adicionou placas de modem ou tem mais portas seriais para o qual o *kernel* foi configurado, apenas recompile seu *kernel*. Veja a a seção de compilação do *kernel* para obter mais detalhes.

P: Como eu posso dizer se o FreeBSD encontrou minha placa de fax modem?

R: Refira-se à resposta da pergunta anterior.

P: Eu acabei de atualizar para a versão 2.0.5 e as minhas `ttyOX` desapareceram! Como eu resolvo esse problema?

R: Não se preocupe, eles foram incluídos com os dispositivos `ttydX`. No entanto, você deve mudar todos os arquivos da configuração antiga que você tiver.

P: Como eu acesso as portas seriais no FreeBSD?

R: A terceira porta serial, `sio2` (veja sio(4), conhecida como COM3 no DOS), está na `/dev/cuaa2` para os dispositivos dial-out, e na `/dev/ttyd2` para os dispositivos dial-in. Qual é a diferença entre essas duas classes de dispositivos?

Você utiliza `ttydX` para dial-ins. Quando o `/dev/ttydX` se abre no modo de bloqueio, um processo irá aguardar que o dispositivo `cuaaX` correspondente torne-se inativo, e aguarda a detecção do *carrier detect* da linha para ativar-se. Quando a `cuaaX` se abre, ela deixa claro que a porta não esta ainda em uso pelo dispositivo `ttydX`. Se a porta estiver disponível, ela é "roubada" do dispositivo `ttydX`. Além disso, o dispositivo `cuaaX` não se importa com o carrier detect. Com esse esquema de auto-resposta do modem, você pode ter usuários remotos conectando e você pode ainda discar para fora com o mesmo modem, que o sistema irá cuidar de todos os conflitos.

P: Como eu habilito suporte para uma placa serial de múltiplas portas?

R: Novamente, a seção de configuração do *kernel* provê informações sobre a configuração de seu *kernel*. Para uma placa serial de múltiplas portas, coloque uma linha sio(4) para cada porta serial da placa, no arquivo de configuração do *kernel*. Mas coloque o irq e as espeficicações do vetor apenas em uma das entradas. Todas as portas da placa devem compartilhar uma irq. Para consistência, utilize a última porta serial para especificar a irq. Além disso, especifique a opção `COM_MULTIPORT` .

O exemplo seguinte é para uma placa serial AST 4-portas na irq 7:

```
options -"COM_MULTIPORT"
device sio4 at isa? port 0x2a0 tty flags 0x781
device sio5 at isa? port 0x2a8 tty flags 0x781
device sio6 at isa? port 0x2b0 tty flags 0x781
device sio7 at isa? port 0x2b8 tty flags 0x781 irq 7 vector ↺
siointr
```

As flags indicam que a porta master tem um "minor number" `7` (`0x700`), diagnósticos habilitados durante o escaneamento (`0x080`), e todas as portas compartilham uma irq (`0x001`).

P: O FreeBSD pode trabalhar com placas seriais de múltiplas portas compartilhando irqs?

R: Ainda não. Você deverá utilizar uma irq diferente para cada placa.

P: Posso definir os parâmetros seriais padrão para uma porta?

R: O `ttydX` (`cuaaX`) é um dispositivo regular que você vai querer abrir para suas aplicações. Quando um processo abre um dispositivo, ele tem um conjunto padrão de configurações de terminais de E/S. Você pode ver essas configurações com o comando

```
# stty --a --f -/dev/ttyd1
```

Ao alterar as configurações para esse dispositivo, elas se manterão em efeito até que o dispositivo seja fechado. Quando ele for reaberto, vai para o estado padrão. Para fazer mudanças nos ajustes padrão, pode-se abrir e ajustar as configurações do "estado inicial" do dispositivo. Por exemplo, para ligar o modo CLOCAL, 8 bits, e o controle de fluxo XON/XOFF padrão para a ttyd5, faça:

```
# stty --f -/dev/ttyid5 clocal cs8 ixon ixoff
```

Um bom lugar para fazer isso é no `/etc/rc.serial` . Agora, uma aplicação terá estas configurações por padrão quando abrir o `ttyd5`. No entanto, pode-se ainda modificar estas configurações a seu gosto.

Você pode prevenir certas configurações de serem modificadas por uma aplicação fazendo ajustes no dispositivo de "lock state". Por exemplo, para travar a velocidade do `ttyd5` em 57600 bps, faça:

```
# stty --f -/dev/ttyld5 57600
```

Agora, uma aplicação, ao abrir o `ttyd5`, se tentar modificar a velocidade da porta, ficará travada a 57600 bps.

Naturalmente você deve garantir que os dispositivos de estado inicial e o estado de trava (`lock`) tenham permissão de escrita apenas para o `root`. O script MAKEDEV(8) *NÃO* faz isso quando ele cria as entradas de dispositivos.

P: Como eu habilito logins dial-up no meu modem?

R: Então você quer tornar-se um provedor de serviços internet, não é? Primeiro, você precisa de um ou mais modems que auto-respondam às chamadas. Seu modem precisa confirmar o "carrier detect" quando ele for detectado e não fazê-lo todo o tempo. Ele precisará desligar o telefone e resetar a si mesmo quando a linha DTR (Data Terminal Ready) alternar de ligado para desligado. Ele provavelmente deve utilizar o controle de fluxo `RTS/CTS` ou nenhum controle local de fluxo. Finalmente, ele deve utilizar uma velocidade constante entre o computador e si mesmo, mas (para ser simpático com seus usuários) ele deve negociar uma velocidade entre si mesmo e o modem remoto.

Para muitos modems compatíveis com o conjunto de comandos do Hayes este comando criará estas configurações e as armazenará na memória não volátil:

```
AT &C1 &D3 &K3 &Q6 S0=1 &W
```

Veja a seção enviando comandos AT abaixo para mais informações sobre como fazer estas configurações sem o auxílio de um programa de terminal MS-DOS.

Depois, faça uma entrada em `/etc/ttys` (veja ttys(5)) para o modem. Este arquivo lista todas as portas nas quais o sistema irá aguardar pelos logins. Adicione uma parecida com essa:

```
ttyd1 -"/usr/libexec/getty std.57600" dialup on insecure
```

Esta linha indica que a segunda porta serial (`/dev/ttyd1`) tem um modem conectado e rodando a 57600 bps e sem paridade (`std.57600`, que vem do arquivo `/etc/gettytab`, veja gettytab(5)). O tipo de terminal para esta porta é `dialup`. A porta esta `ligada` e é `insegura` - quer dizer que o login do usuário `root` não é permitido. Para portas dialin como esta, utiliza-se a entrada `ttydX`

É uma prática comum utilizar `dialup` como o tipo do terminal. Muitos usuários configuram um *prompt* para seus arquivos `.profile` ou `.login` para o tipo de terminal existente se o tipo iniciante é dialup. O exemplo mostra a porta como insegura. Para tornar-se `root` nesta porta, você tem que logar-se como um usuário regular, e então su(1) para tornar-se `root`. Se você usar `seguro`, então o `root` vai poder efetuar o login diretamente.

Após efetuar as operações no `/etc/ttys`, você precisa enviar um sinal de *hangup* ou HUP para o processo init(8):

```
# kill --HUP 1
```

Esse comando forçará a releitura do arquivo `/etc/ttys`. O processo init iniciará os processos getty em todas as portas configuradas em `on` (ligadas). Você pode descobrir se seus logins estão disponíveis para sua porta digitando:

```
% ps --ax -| grep -'[t]tyd1'
```

Você deve ver algo como:

```
747 -??  I      0:00.04 -/usr/libexec/getty std.57600 ttyd1
```

P: Como eu conecto um terminal burro ao meu FreeBSD?

R: Se você esta usando outro computador como um terminal de seu sistema FreeBSD, consiga um cabo “null modem” para ser usado entre as duas portas seriais. Se você esta utilizando um terminal próprio, veja as instruções que o acompanham.

Então, modifique o `/etc/ttys` (veja ttys(5)), como acima. Por exemplo, se você esta ligando um terminal WYSE 50 à quinta porta serial, utilize uma entrada como esta::

```
ttyd4 -"/usr/libexec/getty std.38400" wyse50 on secure
```

Esse exemplo mostra que a porta em `/dev/ttyd4` tem um terminal wyse50 conectado a 38400 bps (bits por segundo) sem nenhuma paridade (`std.38400` de `/etc/gettytab`, veja gettytab(5))) e o login do `root` é permitido (`seguro`).

P: Por que eu não posso executar o `tip` ou o `cu`?

R: Em seu sistema, os programas tip(1) e cu(1) são provavelmente executáveis somente pro `uucp` e para o grupo `dialer`. Você pode utilizar o grupo `dialer` para controlar

quem acessa o seu modem ou sistemas remotos. Basta adicionar você mesmo ao grupo dialer.

Alternativamente, você pode permitir a todos no seu sistema executarem o tip(1) e o cu(1) digitando:

```
# chmod 4511 -/usr/bin/cu
# chmod 4511 -/usr/bin/tip
```

P: Meu modem modelo Hayes não é suportado - o que eu posso fazer?

R: De fato a manpage para o tip(1) esta desatualizada. Há um discador generico do Hayes já incorporado. Apenas insira `at=hayes` em seu arquivo `/etc/remote` (veja remote(5)).

O drive do Hayes não é inteligente o bastante para reconhecer algumas das avançadas características dos modems mais novos - mensagens como `BUSY`, `NO DIALTONE`, ou `CONNECT 115200` estarão apenas confundindo-o. Você deve desabilitar estas mensagens quando utilizar o tip(1) (com o comando `ATX0&W`).

Além disso, o timeout para discagem com o tip(1) é de 60 segundos. Seu modem deve utilizar um valor menor, senão o tip pensará que existe um problema de comunicação. Tente `ATS7=45&W`.

De fato, como o tip(1) não foi compilado para suportar HAYES, essa funcionalidade não é completamente suportada. A solução é editar o arquivo `tipconf.h` no diretório `/usr/src/usr.bin/tip/tip`. Obviamente você precisa da distribuição fonte para fazer isso.

Edite a linha `#define HAYES 0` alterando-a para `#define HAYES 1`. Depois digite `make` e `make install`. Tudo funcionará bem depois disso.

P: Como posso entrar nestes comandos AT?

R: Faça o que é chamado de uma entrada "direta" no seu `/etc/remote` (veja remote(5)). Por exemplo, se o seu modem está definido na primeira porta serial, `/dev/cuaa0`, coloque a seguinte linha:

```
cuaa0:dv=/dev/cuaa0:br#19200:pa=none
```

Utilize a taxa de velocidade mais alta que seu modem suportar na capacidade br. Então digite `tip cuaa0` (veja tip(1)) e você estará conectado ao seu modem.

Se não existir nenhum `/dev/cuaa0` no sistema, faça isso:

```
# cd -/dev
# sh MAKEDEV cuaa0
```

Ou utilize cu como `root` com o seguinte comando:

```
# cu --lline --sspeed
```

com a *line* sendo a porta serial (por exemplo, `/dev/cuaa0`) e *speed* sendo a velocidade (por exemplo, `57600`). Quando terminar com os comandos AT, digite `~.` para sair.

P: Porque o sinal <@> para a capacidade pn não funciona?

R: O sinal <@> no número de telefone diz ao tip para procurar em `/etc/phones` por um número de telefone. Mas o sinal <@> é também um caracter especial em arquivos como o `/etc/remote`. Escape dele com um \ (barra invertida):

```
pn=\@
```

P: Como eu posso discar um número de telefone pela linha de comando?

R: Coloque o que é chamado de uma entrada "genérica" no arquivo `/etc/remote` (veja remote(5)). Por exemplo:

```
tip115200|Disque para qualquer número em 115200 bps:\
        -:dv=/dev/cuaa0:br#115200:at=hayes:pa=none:du:
tip57600|Disque para qualquer número em 57600 bps:\
        -:dv=/dev/cuaa0:br#57600:at=hayes:pa=none:du:
```

Depois você pode fazer algo como `tip -115200 5551234`. Se preferir o cu(1) ao invés do tip(1), utilize uma entrada genérica:

```
cu115200|Use o cu para discar qualquer número em 115200bps:\
        -:dv=/dev/cuaa1:br#57600:at=hayes:pa=none:du:
```

e digite `cu 5551234 -s 115200`.

P: Tenho que definir a taxa de bits por segundo sempre que fazer isso?

R: Coloque uma entrada para `tip1200` ou para `cu1200`, mas vá em frente e utilize quaisquer taxas de bps (bits por segundo) que sejam apropriadas para a capacidade br. O tip(1) diz que um bom padrão é 1200 bps porquê ele procura uma entrada `tip1200`. De qualquer forma, você não precisa utilizar 1200 bps.

P: Como posso facilmente acessar vários hosts através de um `terminal server`?

R: Ao invés de esperar até a conexão, digitando `CONNECT host` sempre, use a opção `cm` do tip. Por exemplo, estas entradas em `/etc/remote` (veja remote(5)):

```
pain|pain.deep13.com|Forrester's machine:\
        -:cm=CONNECT pain\n:tc=deep13:
muffin|muffin.deep13.com|Frank's machine:\
        -:cm=CONNECT muffin\n:tc=deep13:
deep13:Gizmonics Institute terminal server:\
        -:dv=/dev/cuaa2:br#38400:at=hayes:du:pa=none:pn=5551234:
```

permitirá a você digitar `tip pain` ou `tip muffin` para se conectar aos hosts `pain` ou `muffin`; e `tip deep13` para se conectar ao `terminal server`.

P: O "tip" pode tentar discar mais de uma linha por vez?

R: Normalmente esse é um problema tradicional onde uma universidade possue várias linhas de modems e vários milhares de estudantes tentando usá-las...

Faça uma entrada para sua universidade em `/etc/remote` (veja remote(5)) e utilize a `<\@>` para a característica `pn`:

```
big-university:\
        -:pn=\@:tc=dialout
dialout:\
        -:dv=/dev/cuaa3:br#9600:at=courier:du:pa=none:
```

Então, liste os números de telefones para a universidade em `/etc/phones` (veja phones(5)):

```
big-university 5551111
big-university 5551112
big-university 5551113
big-university 5551114
```

O tip(1) tentará usar cada um na ordem listada, e depois desistirá. Se você quer manter-se tentando, execute o tip(1) em um loop while.

P: Por que eu tenho que pressionar duas vezes o CTRL+P para enviar CTRL+P uma única vez?

R: CTRL+P é o padrão para "force character", utilizado para dizer ao tip(1) que o próximo caracter é um dado literal. Você pode definir o force character para qualquer outro caracter com o escape `~s`, que quer dizer "defina uma variável".

Digite `~sforce=single-char` seguido de uma *newline*. *`single-charcaracter`* é qualquer caracter único. Se você deixar *`single-char`*, então o "force character" será o caracter nulo, que você pode ao digitar CTRL+2 ou CTRL+SPACE. Um valor muito bom para o *`single-char`* é SHIFT+CTRL+6, o qual eu vi sendo usado em alguns servidores de terminais.

Você pode ter o "force character" que você quiser especificando o seguinte em seu arquivo `$HOME/.tiprc` o seguinte:

```
force=single-char
```

P: Por que tudo o que eu digito inesperadamente está em CAIXA ALTA?

R: Você deve ter pressionado CTRL+A, o "raise character do tip(1)", especialmente projetado para pessoas com teclas Caps Lock que não funcionam. Use o `~s` como

acima, e defina a variável "raisechar" para algo razoável. De fato, você pode definir isso para o *force character* também, se você nunca espera utilizar ambas as características.

Aqui está um exemplo de arquivo `.tiprc` perfeito para os usuários de Emacs que precisam digitar muitos CTRL+2 e CTRL+A:

```
force=^^
raisechar=^^
```

O ^^ é obtido com SHIFT+CTRL+6.

P: Como eu posso transferir arquivos com o `tip`?

R: Se você está conversando com outro sistema Unix, você pode enviar e receber arquivos com `~p` (put) e com `~t` (take). Estes comandos executam o cat(1) e o echo(1) no sistema remoto para aceitar e enviar arquivos. Sua sintaxe é:

```
~p <local-file> [<remote-file>]
~t <remote-file> [<local-file>]
```

Não há nenhuma checagem de erro, então você provavelmente deve usar um outro protocolo, como o zmodem

P: Como eu posso executar o zmodem com o tip?

R: Primeiro, instale um dos programas zmodem da coleção de `ports` (tais como lrzsz ou o rzsz).

Para receber arquivos, inicie o programa de envio no destino remoto. Então, pressione ENTER e digite `~C rz` (ou `~C lrz` caso tenha instalado o lrzsz) para iniciar o recebimento local

Para enviar arquivos, inicie o programa do lado remoto. Depois, aperte ENTER e digite `~C sz arquivos` (ou `~C lsz arquivos`) para envia-los ao sistema remoto.

P: O FreeBSD pode vir a não encontrar minhas portas seriais, mesmo quando as configurações estão corretas?

R: Sim, se sua placa-mãe for Acer UARTS. Elas não escaneiam corretamente o barramento serial, não permitindo que o FreeBSD encontre as Serial Input/Output (sio) da placa. O patch disponível em www.lemis.com pode corrigir esse problema.

Capítulo 16. Perguntas Variadas

P: O FreeBSD usa bem mais espaço de swap do que o Linux. Por quê?

R: Só parece que o FreeBSD usa mais swap do que o Linux. Na verdade não usa. A principal diferença entre o FreeBSD e o Linux nesse quesito é que o FreeBSD vai sempre remanejar - de forma pró-ativa - toda memória que estiver completamente inativa e subutilizada, para o swap, dessa forma garantindo sempre mais memória principal disponível para utilização. O Linux tende a remanejar páginas de memória para o swap apenas como última alternativa. A utilização mais acentuada do swap é balanceada pela utilização mais eficiente da memória principal.

Note que, pelo fato do FreeBSD ser próativo nesse quesito, ele não decide arbitrariamente fazer swap das páginas quando o sistema está de fato inativo. Portanto você não corre o risco de encontrar todo seu sistema despaginado pela manhã, depois de uma noite inteira de inatividade.

P: Por que o `top` me mostra pouquíssima memória livre, mesmo quando eu não tenho muitos programas rodando?

R: A resposta simples é que memória principal livre é memória desperdiçada. Toda memória que não estiver ativamente alocada pelos seus programas são utilizadas pelo Kernel do FreeBSD como cache de disco. Os valores que o top(1) mostra como `Inact`, `Cache`, e `Buf` são dados referentes ao cache de disco, em estágios distintos de utilização. Esses dados cacheados garantem que o sistema não tenha que fazer acesso em um disco local (muito mais lento que a memória) para utilizar os dados que foram acessados recentemente, garantindo assim melhora significativa na performance geral. Na maioria dos casos, se o top(1) mostrar que existe pouca memória disponível, isso é uma boa indicação, a não ser que seja uma quantidade *extremamente* baixa.

P: Por que usar (alias, o que são) os formatos executáveis a.out e ELF?

R: Para entender porque o FreeBSD usa o formato `ELF`, você deve primeiro saber um pouco sobre os 3 formatos de executáveis Unix "dominantes" atualmente:

Nota

Até a versão 3.x o FreeBSD usava o formato a.out.

- a.out(5)

O mais antigo e "classico"formato de objetos Unix. Ele usa um cabeçalho curto e compacto, com um "magic number" no início que é frequentemente utilizado para identificar seu formato (mais detalhes veja a.out(5)). Ele contém três segmentos a serem carregados: .text, .data, e .bss acrescidos de uma tabela de símbolos e uma tabela de caractereres adicionais.

- COFF

  O formato de objetos SVR3. Seu cabeçalho se consiste agora em uma tabela de seções, dessa forma garantindo que você tenha outras seções além de .text, .data, e .bss.

- ELF

  O sucessor do COFF, atribuído de Múltiplas seções e valores de 32-bit ou 64-bit. Um de seus principais inconvenientes: O formato ELF foi originalmente desenvolvido presumindo-se que existiria apenas um único ABI por arquitetura. A presunção é incorreta, e nem mesmo em relação ao mundo comercial do SYSV (onde encontramos ao menos três ABIs distintas: SVR4, Solaris, SCO) isso acontece.

  O FreeBSD tenta se virar com esse problema com um utilitário que *identifica* um executável ELF relacionando-o ao ABI com o qual ele é compatível. Veja a página de manual do brandelf(1) para maiores informações.

O FreeBSD vem de tradição "clássica" e por isso sempre usou o formado a.out(5) que é uma tecnologia que foi experimentada e aprovada por várias gerações de sistemas BSD. Apesar de, há algum tempo também ser possível para o FreeBSD trabalhar nativamente com binários ELF (e também *kernels*), o FreeBSD inicialmente resistiu à "pressão" em assumir o ELF como formato padrão. Por quê? Bem, quando o campo do Linux resolveu fazer sua dolorosa transição para o formato ELF, não sobrou muito para ser aproveitado dos formatos `a.out` especialmente por causa das limitações de tabelas que podiam ser utilizadas em seus cabeçalhos, e isso tornou o desenvolvimento de bibliotecas compartilhadas extremamente árduo para fabricantes e desenvolvedores em geral. Depois disso, as ferramentas ELF começaram à oferecer soluções para o compartilhamento de bibliotecas, soluções que fossem extremamente satisfatórias, e a migração, apesar dos custos necessários que a envolvia, foi aceita, e a transição para ELF passou a ser o "caminho à ser seguido".

No caso do FreeBSD, o nosso mecanismo de bibliotecas compartilhadas tem uma base mais próxima do estilo do SunOS, da Sun, e é extremamente fácil de ser utilizado. Contudo, à partir da série 3.0, o FreeBSD oficialmente adotou o formato de binários ELF como padrão. Apesar do formato `a.out` sempre ter servido muito bem às nossas necessidades, o pessoal da GNU, autores de algumas das ferramentas de compilação que nós usamos, simplesmente deixaram de suportar

o formato `a.out`. Tal fato nos forçou à manter versões distintas do compilador e do linkador, e nos permitiriam usufruir dos esforços que nós achássemos interessantes nos desenvolvimentos GNU. Finalmente, a demanda pelo ISO-C++, notáveis compiladores e descompiladores, também contribuiram para uma adoção nativa dos binários ELF nas versões futuras do FreeBSD.

P: Certo, mas por que existem tantos formatos diferentes?

R: De volta às origens, em um passado obscuro, existiam apenas hardwares mais simples. Esse hardware simples, suportava sistemas simples e pequenos. A a.out era completamente adequada para o serviço de representar o formato binário nesses sistemas (os PDP-11). Conforme as pessoas iam portando o Unix desse sistema mais simples, eles mantinham o formato a.out porque era bom o bastante para portar para arquiteturas como o Motorola 68k, VAXen, etc.

Então, algum engenheiro de hardware brilhante, decidiu que se ele pudesse forçar o software à dar conta de algumas coisinhas, alguns truquezinhos, ele poderia então passar por cima de algumas restrições de design, e permitir que a base de sua CPU tivesse um desempenho melhor. Para poder trabalhar como esse novo tipo de hardware (que hoje é conhecido como RISC), a `a.out` não se encaixava muito bem em suas funções, e então muitos formatos foram desenvolvidos afim de obter melhor performance desse hardware, que a simples `a.out` não podia comportar. Coisas como COFF, ECOFF e outras ainda mais obscuras foram inventadas, e todas suas limitações foram exploradas, até que se resultasse o formato ELF.

Em adição, o tamanho dos programas passou a crescer, e os discos (assim como a memória física) ainda eram relativamente pequenos, então nasceu o conceito de compartilhamento de bibliotecas. O sistema de Memória Virtual (VM) também se tornou mais sofisticado. Cada um desses avanços eram feitos utilizando-se o formato `a.out`, e o seu uso crescia mais e mais com cada nova característica. Depois, as pessoas começaram a querer que as coisas fossem dinâmicamente carregadas em tempo de execussão, ou então queriam poder descartar algum trecho de seus programas depois que seu código de inicialização tivesse sido executado, de modo à economizar memória principal ou mesmo Swap. As linguagens de programação se tornaram mais sofisticadas, então as pessoas queriam códigos com chamadas automáticas antes do programa principal (main). Começou-se então a hackear a `a.out` de forma que ela pudesse suprir essas necessidades. E de fato por algum tempo ela as supriu. Depois a `a.out` passou a não suportar mais determinados problemas sem resultar em uma sobrecarga ou complexidade exagerada de código. Por outro lado, o formato ELF resolvia a maioria desses problemas, mas seria doloroso demais simplesmente abandonar um formato e sistema que, basicamente funcionavam bem. Então o formato ELF teve que esperar até que fosse ainda mais doloroso continuar com o formato `a.out` do que migrar para ELF.

Contudo, com o passar do tempo, as ferramentas de desenvolvimento às quais o FreeBSD derivava suas próprias ferramentas de desenvolvimento (especialmente o assembler e o carregador - loader) se envolveram em duas árvores paralelas. A árvore do FreeBSD adicionou inúmeras bibliotecas compartilhadas, e arrumou

inúmeros bugs. E a rapaziada do GNU, que originalmente escreviam algumas dessas ferramentas, passaram a rescreve-las e adicionaram suporte para compilação derivada, adoções de formatos diferentes, etc. Depois pensou-se em desenvolver um formato derivado, visando o FreeBSD, mas não obteve-se sorte o bastante, especialmente porque os fontes antigos do "ld" do FreeBSD não davam conta da tarefa. A corrente de novas ferramentas GNU (as chamadas binutils) agora suportam compilação derivada, formato ELF, bibliotecas compartilhadas, extensões de C++, etc, etc. Em adição ainda, muitos fabricantes passaram à lançar binários ELF, e então, por que continuar nos chateando com o formato `a.out`? A `a.out` é um cavalo velho e muito cansado, que já provou ser extremamente útil no passado, mas agora está na hora de tira-lo do pasto, como recompensa por seus longos e fiéis anos de serviço.

O formato ELF é mais expressivo do que o `a.out`, e vai permitir muito mais extensibilidade à base do sistema. As ferramentas ELF são mantidas de forma mais confiável, e oferecem suporte à compilação derivada, o que é importante para muita gente. O formato ELF é um pouco mais lento do que o formato `a.out`, mas é quase impossível comparar ambos, existem inúmeros detalhes que os fazem diferentes, desde o mapeamento de páginas de memória, até a forma como eles tratam o código de inicialização de um binário (init code). Nenhuma dessas questões é importante, mas existem diferenças. Com o tempo, o suporte para o formato `a.out` vai ser retirado do *kernel* GENERIC, e eventualmente será retirado em definitivo do *kernel*, uma vez que a necessidade de rodar programas `a.out` tenham se tornado passado.

P: Por que o chmod não modifica as permissões dos links simbólicos?

R: Links simbólicos não tem permissões, e por padrão, o chmod(1) não vai seguir os links afim de mudar as permissões do arquivo original. Portanto, se você tem um arquivo qualquer, e um link simbólico para esse arquivo, o seguinte comando vai lhe servir.

```
% chmod g-w <link simbólico>
```

Contudo, as permissões para o arquivo original não serão alteradas. Mas se você usar a opção `-H` ou `-L` em conjunto com `-R`, você vai poder alterá-la. Veja as páginas de manuais do chmod(1) e do symlink(7) para mais informações.

Atenção

A opção `-R` resulta em um chmod(1) RECURSIVO. Tome muito cuidado quando for definir diretórios ou links simbólicos com chmod(1). Se você quer alterar as permissões dentro do diretório referenciado pelo symlink, então basta usar o chmod(1) sem qualquer outra opção, mas com uma barra `/`. Por exemplo, se `A` for um link simbólico para o arquivo original `B`, então para alterar sua permissão basta um simples:

```
% chmod 555 A/
```

Com essa barra, o chmod(1) vai seguir o link simbólico para mudar as permissões do arquivo original.

P: Por que os nomes de login (ou username) são restritos à 8 caracteres no FreeBSD 2.2.X e anteriores?

R: Você pode pensar que seria bem confortável simplesmente mudar o `UT_NAMESIZE` e depois recompilar todo o sistema operacional, ai tudo iria funcionar maravilhosamente bem. Infelizmente não é assim que as coisas funcionam, existem estruturas de aplicações e utilitários (incluindo ferramentas do sistema) que foram codificadas utilizando números pequenos (nem sempre `8` ou `9`, mas alguns valores mais arbitrários como `15` e `20`) em estruturas e buffers. Você não vai ter problemas apenas com arquivos de logs, que serão inutilizados (devido ao tamanho variável dos dados gravados, quando apenas um tamanho constante era esperado), mas vai também ter problemas com clientes NIS de máquinas Sun, e potencialmente provocar outros problemas ao interagir com outros sistemas Unix.

No FreeBSD 3.0 e posteriores, o tamanho máximo do nome de usuário foi elevado para 16 caracteres, e todas as ferramentas e trechos do código principal do sistema que poderiam apresentar problemas em relação à isso, foram encontradas e corrigidas. O fato dessa alteração mudar tantos fatores importantes no sistema é que, nenhuma mudança tinha sido feita até a versão 3.0.

Se você confia completamente em suas habilitades para procurar e corrigir esses prováveis problemas sozinho, então basta aumentar o tamanho do nome de usuário no arquivo `/usr/include/utmp.h` e mudar a UT_NAMESIZE para o valor desejado. Você também vai ter que atualizar o MAXLOGNAME no `/usr/include/sys/param.h` para ficar de acordo com a mudança no UT_NAMESIZE. Finalmente, se você vai recompilar os fontes, não se esqueça que o /usr/include é atualizado sempre. Mude então os arquivos apropriados em /usr/src(...) para garantir que você vai estar alterando sempre a fonte do problema, e não apenas a instância instalada, no sistema.

P: Posso rodar binários do DOS sob FreeBSD?

R: Sim, à partir da versão 3.0, você pode utilizar o emulador doscmd da BSDI. A emulação DOS desse aplicativo foi totalmente redefinida depois da sua integração do FreeBSD. Entre na lista de discussão sobre a emulação de outros sistemas operacionais no FreeBSD se você tem interesse em se juntar ao grupo que se esforça nessa jornada.

Em sistemas anteriores ao 3.0, existe um utilitário não muito interessante, chamado pcemu no `Ports`. O pcemu emula um 8088 e algumas função de BIOS que são o

bastante para rodar aplicações DOS que sejam textuais. Ele requer o X, sistema de interface grafica (XFree86).

P: O que eu preciso para traduzir um documento do FreeBSD para a minha língua nativa?

R: Veja o FAQ de Tradução no *FreeBSD Documentation Project Primer*.

P: Por que meu e-mail para qualquer endereço em FreeBSD.org sempre falha?

R: O sistema de correio eletrônico do site FreeBSD.org implementa algumas das restrições do Postfix, verificando nas mensagens que estão chegando, se elas estão sendo entregues por algum servidor mal configurado, ou se representa algum tipo de SPAM em potencial. As suas mensagens podem estar voltando por algum dos seguintes motivos:

- A mensagem está sendo enviada de um domínio ou bloco de endereços IP reconhecidamente utilizados para SPAM.

  Os servidores de correio eletrônico do projeto FreeBSD rejeitam mensagens de qualquer fonte de SPAM conhecida. Se você utiliza os serviços de uma empresa que costuma fazer SPAM ou permitir que seus clientes o façam, por gentileza, mude o seu provedor de serviços, para um que não permite tal prática.

- O corpo da mensagem contém apenas HTML.

  Mensagens de correio eletrônico devem ser enviadas apenas como texto puro. Mensagens de e-mail não são web sites. Configure o seu cliente de correio eletrônico de modo que ele apenas envie mensagens de texto puro.

- Os servidores da FreeBSD.org não conseguem resolver o seu endereço IP para o nome da estação que está entregando a mensagem eletrônica.

  Por padrão, ter registros de DNS reverso é um dos requisitos para que nossos servidores recebam sua mensagem. Configure o DNS reverso para o IP do seu servidor de e-mail. Lembre-se que, alguns serviços residênciais (como ADSL, dialup, cable, etc) não permitem que você mesmo configure o seu reverso. Nesse caso, envie sua mensagem pelo servidor de e-mail do seu provedor de serviços, ou peça ao provedor que ajuste o reverso do seu IP.

- O nome da estação enviada no cabeçalho inicial EHLO/HELO, parte do protocolo de envio SMTP não pode ser resolvido em um endereço IP correspondente.

  O servidor que está tentando entregar a mensagem deve ter o registro de nomes configurado corretamente, de forma que o nome da estação possa ser resolvido em um endereço IP. Caso sua estação não tenha um registro DNS configurado, utilize o servidor de correio eletrônico do seu provedor de serviços.

- Sua mensagem teve uma identificação que terminava com o conjunto de caracteres “localhost”.

  Alguns clientes de correio eletrônico geram ID - identificações - das mensagens de forma imprópria. Nesse caso, o seu servidor de correio deve redefinir o ID da mensagem, ou você deve reconfigura-lo de modo que ele gere tal identificação de forma aceitável.

P: Onde eu consigo uma conta gratuíta em um FreeBSD?

R: O Projeto FreeBSD não permite acesso público a nenhum dos seus servidores, contudo algumas empresas oferecem acesso irrestrito à sistemas Unix. Os preços variam, e alguns serviços limitados podem ser disponibilizados.

A Arbornet, Inc, também conhecida como M-Net, provê acesso à sistemas Unix desde 1983. Inicialmente rodando sob um System III em arquitetura Altos, o site mudou seu sistema para BSD/OS em 1991. Em junho de 2000 o site mudou novamente seu sistema para FreeBSD. A M-Net pode ser acessada via telnet e SSH, e proporciona acesso básico a uma gama completa de softwares do FreeBSD. Contudo, o acesso à rede é limitado aos membros e patronos da instituição, que fazem doações à empresa, uma vez que a mesma é uma organização sem fins lucrativos. A M-Net também oferece um Boletim periódico e Chat interativo.

A Grex também oferece um acesso parecido com o da M-Net, inclusive com os mesmos serviços, contudo a máquina é uma Sun 4M e seu sistema Unix é o SunOS. Vale pela curiosidade, e para comparação entre os sistemas.

P: O que é `sup`, e como eu uso isso?

R: SUP significa Protocolo de Atualização de Programa (Software Update Protocol), e foi desenvolvido pela CMU para manter suas árvores de desenvolvimento sempre sincronizadas. Nós utilizamos esse protocolo para manter alguns sites remotos em sincronia com os nossos servidores centrais de desenvolvimento.

SUP náo é amigável com a banda de tramissão (consome muita banda) , e por isso foi aposentado. Atualmente recomendados que você faça uso do CVSup para manter seus fontes atualizados.

P: Qual o nome daquele capetinha vermelho simpático?

R: Ele não tem um nome, é simplesmente chamado de “the BSD daemon”. Se você insiste em dar um nome à ele, por gentileza, chame-o de “beastie” ;-) Note que “beastie” se pronuncia “BSD”.

Você pode saber mais sobre o BSD daemon na sua home page.

P: Posso usar a imagem do BSD daemon?

R: Talvez. O BSD daemon é de direitos autorais de Marshall Kirk McKusick. Você deve pedir a permissão do McKusick para usar a imagem do BSD Daemon, e pedir para saber os termos de utilização da figura pública do nosso querido capetinha ;-)

Resumindo, você pode fazer uso da imagem dele, dependendo da maneira, para uso pessoal, por exemplo. Se os créditos apropriados forem dados, tudo bem. Para fazer uso comercial da imagem, ai sim você deve falar com o McKusick, e dar uma olhada na home page do BSD Daemon para mais detalhes..

P: Vocês tem algumas imagens do BSD daemon que eu poderia usar?

R: Você vai encontrar algumas figuras em eps e Xfig sob o diretório `/usr/share/examples/BSD_daemon/` .

P: O que significa MFC?

R: MFC é um acrônimo para "obtido a partir do ramo -CURRENT" (*Merged From -CURRENT*). É usado nos logs do CVS para identificar uma mudança que seja originada e migrada da série de desenvolvimento (-CURRENT) para série estável (-STABLE).

P: O que significa BSD?

R: O significado da sigla BSD é algo, em uma língua secreta que apenas os membros podem saber. Literalmente não seria possível traduzir BSD para uma língua que você pudesse entender, mas poderíamos tentar explicar seu significado como algo bem próximo de "Equipe de Fórmula-1", "Penguins são aperitivos saborosos", e também "Nós temos mais senso de humor do que o Linux". :-)

A versão séria é que BSD é um acrônimo para "Berkeley Software Distribution", que é o nome que o Grupo de Pesquisa de Ciência da Computação da Universidade de Berkeley - Berkeley CSRG (Computer Systems Research Group) - escolheu para sua própria distribuição do Unix.

P: O que significa POLA?

R: É o Princípio de Menor Alteração. Significa que durante o processo de desenvolvimento do FreeBSD, toda e qualquer modificação que seja visível para o usuário, deve ser menos surpreendente possível, mantendo assim uma compatibilidade prévia com a forma de utilização do sistema. Por exemplo, não se pode alterar arbitrárriamente as variáveis dos scripts de configuração do sistema, em `/etc/defaults/rc.conf` , pois esse tipo de ação violaria a POLA. O desenvolvimento do FreeBSD apenas considera POLA quando as alterações são visíveis pelo usuário.

P: O que é um repo-copy?

R: Um repo-copy (que é uma forma breve de chamar um "repository copy") é simplesmente a cópia direta de arquivos em um repositório CVS.

Sem um repo-copy, uma alteração por parte de algum mantenedor, se tornaria uma cópia comum, originada via `cvs`, seguida de um `rm` para deletar o arquivo original que tivesse sido modificado.

Esse processo contudo, resulta em uma não constatação histórica do arquivo antigo, nos novos registros de log. O Projeto FreeBSD considera extremamente importante a manutenção desse histórico, e por isso as cópias de repositório são frequentemente utilizadas. Nesse processo, um dos repositórios centrais vai copiar os arquivos diretamente para outro repositório, e não simplesmente fazer uma sincronia com o programa cvs(1).

P: Porque eu devo me preocupar com a cor do quartinho de bicicletas (bikeshed)?

R: A resposta mais curta, é que você não deve. A resposta longa é que, só porque você é capaz de fazer quarto para guardar sua própria bicicleta, você não pode fazer as outras pessoas pararem de construir seus próprios quartinhos também, simplesmente porque você não gosta da cor que as pessoas os pintam. Essa metáfora indica que você não tem que argumentar nem reclamar sobre cada coisinha, só porque tem conhecimento o bastante para critica-la. Algumas pessoas dizem que a quantidade de barulho provocada por uma alteração é inversamente proporcional à complexidade da mudança.

A resposta ainda mais completa, e maior, é que, depois de muita discussão sobre quando o sleep(1) deveria trabalhar com argumentos de segundos fracionados, Poul-Henning Kamp enviou uma mensagem, longa, chamada de "Um quarto de bicicleta (qualquer cor serve) em um gramado mais verde...". As partes devidas da mensagem estão citadas abaixo.

> "O que é isso sobre a cor do quartinho da bicicleta?" Alguns de vocês me perguntaram.
>
> É uma longa história, ou melhor, é uma antiga história, mas pode ser meio curta na verdade. Um cara chamado C. Northcote Parkinson escreveu um livro em meados de 1960 chamado de "A Lei de Parkinson", que continha vários insights sobre as dinâmicas da administração.
>
> [trechos irrelevantes foram cortados]
>
> Vamos deixar de falar sobre o livro em sí, mas vamos tratar apenas um exemplo. O exemplo específico, do caso do quartinho da bicileta, tem outro componente de vital importância, que é uma usina nuclear. Isso ilustra a época que o livro foi escrito.
>
> Parkinson mostra como você pode fazer para chegar em um corpo de diretores de uma empresa e convencê-los a construir uma usina atômica multi-milhionária ou até mesmo bilhionária,

> mas diz que se você abordar os diretores da mesma forma, tentando aprovar a construção de um quartinho de bicicleta, você vai cair em uma discussão profunda, e sem fim.
>
> Parkinson explica que isso acontece porque uma usina atômica é tão vasta, tão cara e tão complicada que as pessoas simplesmente preferem não discutir, e mesmo que tentem fazê-lo, eles assumem que alguém já observou todos os detalhes possíveis antes que tal proposta chegasse à tal ponto. Richard P. Feunmann dá alguns exemplos interessantes, sobre o que acontece em Los Alamos em seus livros.
>
> Por outro lado, um barracão de bicicleta pode ser construído por qualquer um, em um fim de semana qualquer, e ainda sobraria tempo ao seu desenvolvedor para assistir o jogo na TV. Então, não importa o quão bem preparado você esteja, alguém vai sempre querer aparecer diante de uma situação dessas, e querer discutir as coisas mais pequenas possíveis. Na Dinamarca isso se chama "Deixar sua marquinha". Envolve prestígio e orgulho pessoal, envolve a possibilidade de apontar para algum lugar (qualquer lugar que seja) e apontar dizendo "*Aquilo! Eu fiz aquilo*" (o que quer que aquilo seja). Isso é comum em políticos, mas aparece em qualquer pessoa a quem se dê a chance. Simplesmente pense em pegadas, no cimento fresco."
>
> —Poul-Henning Kamp na freebsd-hackers, em 2 de Outubro de 1999

Na verdade, um "quartinho de bicicletas" ou "barracão de bicicletas" é uma tradução literal para a expressão "bikeshed"; comumente utilizada na língua inglesa. Um "bikeshed", no significado definido pelo dicionário norte americano é um pequeno quarto ou barração não raramente encontrado no fundo de uma casa, que é utilizado para guardar bicicletas e outras coisas pequenas. Normalmente os norte americanos constroem esses quartinhos eles mesmos, de madeira, no fundo de suas casas ou próximos à garagem de automóveis. A expressão é normalmente utilizada pelos desenvolvedores do FreeBSD quando se começa uma discussão sobre algum assunto que não é tão importante para o bom funcionamento de alguma outra coisa, como por exemplo, qual a importância da cor de um quartinho de bicicletas, quando o mesmo já está construído e servindo bem ao seu propósito?

Capítulo 17. As gracinhas do FreeBSD

P: Quão fresco é o FreeBSD?

R: P. Alguém já fez algum tipo de teste de temperatura ao rodar o FreeBSD? Eu sei que o Linux costuma ser mais fresco que o DOS, mas nunca ouvi falar nada a respeito do FreeBSD. Parece que ele é um sistema muito *caliente*!

R. Não, mas nós já fizemos vários testes de sabor com usuários vendados que, além de tudo, haviam tomado 250 microgramas de LSD-25. 35% dos voluntários disseram que o FreeBSD tinha um sabor parecido com laranja, enquanto o Linux tinha sabor de névoa púrpura. Nenhum dos dois grupos comentou nada significante sobre a variação de temperatura. Eventualmente, tivemos que jogar o resultado desses testes fora, já que descobrimos que vários desses voluntários estavam vagando fora do quarto, prejudicando os resultados. Acreditamos que hoje, a maioria desses voluntários trabalhe na Apple. Eles devem estar criando novas interfaces gráficas do tipo "arranha e cheira". É um trabalho divertido e clássico, do qual nós fazemos parte!

Falando sério, o FreeBSD e o Linux usam as instruções HLT (halt) quando o sistema está inativo. Isto diminui consideravelmente o consumo de energia e, conseqüentemente, o aquecimento que ele proporciona. Além disso, se seu sistema possui APM (sistema avançado de gerenciamento de energia) configurado, o FreeBSD coloca a CPU em modo de consumo menor de energia.

P: Quem está arranhando meus pentes de memória??

R: P. Existe alguma "bruxaria" que o FreeBSD faz ao compilar o *kernel* que, por ventura, estaria fazendo meus pentes de memória fazer barulhos estranhos, como se estivessem sendo arranhados? Durante a compilação do sistema (e um pouquinho depois, assim que a unidade de disquete é reconhecida, e após a inicialização também), um barulho estranho de arranhos começa a emanar de algum lugar que parece ser os pentes de memória.

R. Claro! Com muita frequência, você vai ouvir falar dos "daemons" na documentação do BSD. O que a maioria das pessoas não sabem é que essa é uma referência genuína às entidades não-corporais que estão possuindo o seu computador. O barulho que parece o som de alguma coisa sendo arranhada, na verdade são sussuros em tons extremamente agudos que os "daemônios" emanam, ao decidir entre si as melhores maneiras de tratar as várias tarefas referentes à administração do seu sistema.

Se o barulho te dominar, um bom `fdisk /mbr` no DOS pode fazer você se livrar dos sons, mas não se surpreenda se a reação dos *daemons* forem adversas, ao tentar

evitar que você faça isso. Na verdade, há qualquer momento, é possível que você ouça a voz satânica do Bill Gates pelo auto-falante interno do seu PC. Se isso acontecer, CORRA! Corra sem parar e não olhe para trás por qualquer que seja o motivo! Depois de se liberarem das influências contrastantes dos *daemons* BSD, os demônios gêmeos do DOS e do Windows costumam ter sucesso ao repossuir total controle do seu computador, e depois disso, o tempo garantirá que eles consigam a dominação total da sua alma. Agora que você já conhece a verdade, esperamos que sua escolha seja conviver com os barulhinos agudos. Ou não?

P: Quantos FreeBSD *hackers* são necessários para trocar uma lâmpada?

R: Mil cento e sessenta e nove:

Vinte e três para reclamarem no -CURRENT que estão sem luz;

Quatro para dizer que é um problema na configuração, e que essa pergunta deveria ser feita na freebsd-questions;

Três para enviar Relatório de Problemas sobre a lâmpada, dos quais ao menos um, não está completamente concluído, e consiste apenas de um breve “tá escuro”;

Um para adicionar uma lâmpada que nunca foi testada, que danifica todo o `buildworld` e depois de 5 minutos tem que ser retirada;

Oito para reclamarem para os autores dos Relatórios de Problemas por não ter incluído correções em seus relatórios;

Cinco para reclamar que o `buildworld` não está funcionando;

Trinta e um para responder que funciona para eles, e que os problemáticos devem ter feito CVSup na hora errada;

Um para enviar uma correção para a nova lâmpada na freebsd-hackers;

Um para reclamar que ele tinha correções para essa lâmpada há 3 anos, mas que quando elas foram enviadas para o -CURRENT, foram simplesmente ignoradas, e que sua experiência com o sistema de Relatório de Problemas não foram as melhores possíveis; além disso a nova lâmpada proposta não era reflexiva;

Trinta e sete para gritarem em alto e bom som que as lâmpadas não fazem parte da base do sistema, e que os desenvolvedores não tem o direito de sair fazendo esse tipo de coisa sem antes consultar a comunidade, e O QUE O -CORE ESTA FAZENDO SOBRE ISSO!?

Duzendos para reclamar da cor do quartinho de bicicletas;

Três para dizer que a correção enviada não está de acordo com os padrões que o código do *kernel* deve ter, conforme documentado na página de manual do style(9);

Dezessete para reclamar que a nova lâmpada proposta está licenciada sob a Licença Pública Geral GNU (GPL);

Quinhentos e oitenta e seis para entrarem de corpo e alma em uma discussão sobre as vantagens comparativas entre a licença Pública Geral GNU (GPL), a licença BSD, a licença do MIT, a NPL e a higiene pessoal dos fundadores da *Free Software Foundation*;

Sete para copiar vários trechos da discussão para a lista de discussão freebsd-chat e para a freebsd-advocacy;

Um para trocar a nova lâmpada sugerida, apesar de a nova brilha bem menos que a antiga;

Dois para retirarem a lâmpada furiosos, dizendo que o FreeBSD está melhor no escuro do que com uma lâmpada tão fraca;

Quarenta e seis para contestarem vorazmente sobre a retirada da lâmpada fraca e escreverem um relatório para o -core;

Onze para dar a idéia de criar uma lâmpada menorzinha, que poderia caber no Tamagotchi deles, se um dia nós decidirmos portar o FreeBSD para tal plataforma;

Setenta e três para reclamar da razão sinal versus ruído na freebsd-chat e na freebsd-hackers, e se retirarem das listas em protesto;

Treze para enviarem mensagens com o conteúdo "unsubscribe", "Como eu saio da lista?", ou "Por favor, me tirem da lista", seguidas do rodapé tradicional do servidor de discussão com as instruções para sair da lista;

Um, para adicionar uma nova lâmpada que funciona bem, enquanto todos os outros estão ocupados demais com a discussão para perceber que alguém já trocou a lâmpada por uma funcional;

Trinta e um para afirmar que a nova lâmpada brilha em média 0.364% mais, se comparada com as lâmpadas TenDRA (contudo, ela terá que ser refeita em formato de cubo) e que o FreeBSD deveria mudar para TenDRA ao invés do GCC;

Um para reclamar que a nova lâmpada não é honesta;

Nove (incluindo aqueles que enviaram os Relatórios de Problemas) para perguntar “o que significa MFC?”;

Cinquenta e sete para reclamar que ficaram no escuro por duas semanas até que a lâmpada fosse trocada.

Um adendo do Nik Clayton:

Eu estava rindo um bocado aqui.

Aí pensei, "Peraí, não deveria ter ao menos '1 para documentar a nova lâmpada' em algum lugar?

Daí eu fui iluminado :-)

P: Para onde vão os dados que são escritos no dispositivo `/dev/null` ?

R: Esses dados são enviados para um dissipador especial da CPU que os converte em calor, para que depois sejam ventilados pelo *cooler* do computador. É por isso que o esfriamento do processador é cada vez mais importante; quanto mais rápido os processadores se tornam, menos importância os usuários dão à seus dados, por isso cada vez mais lixo é enviado para o `/dev/null` , gerando um superaquecimento das CPUs. Se o `/dev/null` for apagado (dessa forma desabilitando o dissipador de dados da CPU), o sistema vai rodar a uma temperatura mais amena. Contudo, o computador vai manter tanto lixo inútil existente, que o sistema vai logo começar a falhar. Se você tiver uma conexão de rede bem rápida, dá para resfriar o computador lendo todos os dados criados na `/dev/random` e enviando-os para algum lugar da rede. Contudo existe o risco de superaquecer sua rede ou do Provedor de Serviço Internet ficar meio bravo com você, já que todo esse calor normalmente é recebido pelo equipamento do provedor. Mas não se preocupe, os provedores tem grandes ventiladores para esfriar suas máquinas, então se você não insistir nisso com muita frequência, vai ficar tudo bem.

Adendo de Paul Robinson:

Existem outros métodos. Como todo bom administrador de sistemas sabe, faz parte da prática comum enviar dados das mais variadas espécies para a tela. Isto mantêm todos os *pixies* (*pixie* significa fadinhas em inglês) de tela felizes. Os *pixies* de tela (normalmente escritos com erro de ortografia, como 'pixels') são divididos de acordo com o tipo de boné que eles usam (vermelho, verde ou azul) e costumam aparecer ou sumir (mostrando a cor de seus bonés) sempre que eles ganham alguma coisinha para comer. As placas de vídeo transformam os dados em comida de *pixies*, e manda essa comida para eles. Quanto mais cara for a placa de vídeo, melhor é a qualidade da comida. Dessa forma, mais felizes ficam os *pixies*. Os *pixies* também precisam ser constantemente estimulados – é para isso que existem as proteções de telas.

Então, para seguir a sugestão anterior, é interessante mandar todos os dados que saírem do `/dev/random` para a tela do console, para alimentar os *pixies*. Isso não causa nenhum aquecimento do computador, e em contrapartida, faz os *pixies* viverem mais felizes, e ainda pode ser que faça você se livrar rapidamente de todos os dados existentes no `/dev/random` , mesmo considerando que a tela fique um pouco confusa.

Como um ex-administrador de um provedor que teve algumas más experiências tentando manter a estabilidade da temperatura da sala dos servidores, eu recomendo sinceramente que as pessoas não tentem enviar todos os seus dados

para a rede. Existem umas pequenas fadinhas encantadas que fazem a alternância dos pacotes de redes, e que fazem o roteamento desses mesmos pacotes. Algumas vezes essas fadinhas ficam meio revoltadas com os usuários malvados que ficam mandando seus dados inúteis para a rede.

Capítulo 18. Tópicos Avançados

P: Como eu posso aprender mais sobre as características internas do FreeBSD?

R: Atualmente não há nenhum livro específico sobre as características internas do Sistema Operacional FreeBSD. Contudo, a maior parte do conhecimento genérico sobre UNIX pode ser aplicado diretamente a ele. Além disso existem livros específicos para sistemas BSD cuja leitura é recomendada.

Para uma lista, verifique a sessão de bibliografia sobre características internas dos sistemas operacionais no Manual do FreeBSD.

P: Como posso contribuir com o projeto FreeBSD?

R: Por gentileza, consulte o artigo Contribuindo com o Projeto FreeBSD para obter algumas dicas sobre o assunto. Toda ajuda é mais que bem vinda!

P: O que são SNAPs e RELEASEs?

R: Atualmente existem três séries ativas/semi-ativas no Repositório CVS do projeto FreeBSD (a RELENG_2 que é provavelmente alterada somente duas vezes ao ano, sendo esta a razão de termos somente três séries em desenvolvimento):

- `RELENG_2_2` ou *2.2-STABLE*
- `RELENG_3` ou *3.X-STABLE*
- `RELENG_4` ou *4-STABLE*
- `HEAD` ou *-CURRENT* ou *5.0-CURRENT*

`HEAD` não é um nome de uma tag de série, como os outros dois; é somente uma constante simbólica para "*o atual desenvolvimento corrente, mas não de série*" a qual nós simplesmente nos referimos como "-CURRENT".

Neste momento, a "-CURRENT" se refere ao desenvolvimento atual do FreeBSD 5.0. A série `4-STABLE` , RELENG_4 originou-se da "-CURRENT" em Março de 2000.

A série `2.2-STABLE` , RELENG_2_2, originou-se da "-CURRENT" em Novembro de 1996, e foi praticamente descontinuada.

P: Como faço a minha própria distribuição personalizada?

R: Por gentileza, consulte o artigo sobre a Engenharia de Releases..

P: Por que o `make world` sobrescreve os binários já instalados?

R: Porque essa é a idéia geral sobre como ele deve funcionar; como seu nome sugere, o `make world` reconstrói todo o sistema binário a partir do zero, garantindo que o usuário tenha um ambiente limpo e consistente ao final da operação (é por isso que o processo demora tanto).

Se a variável de ambiente `DESTDIR` estiver definida enquanto um `make world` ou `make installworld` estiver sendo executado, os binários recém criados serão distribuídos no diretório definido em `${DESTDIR}` , criando no mesmo uma réplica do conteúdo do / do sistema. Algumas alterações aleatórias nas bibliotecas compartilhadas podem ocasionar falhas na hora de reconstruir o sistema com o `make world.`

P: Por que quando meu sistema inicializa, ele diz "(bus speed defaulted)"?

R: Os controladores SCSI Adaptec 1542 permitem que o usuário defina a velocidade de acesso ao barramento por meio de software. Algumas versões mais antigas deste dispositivo tentavam determinar automaticamente a maior velocidade possível e tentavam ajustar sua velocidade à esse limite máximo. Descobriu-se contudo, que esse comportamento as vezes era prejudicial, e fazia com que algumas máquinas não funcionassem de forma adequada, por este motivo essa característica agora vem desabilitada por default, para ativá-la é necessário definir a opção TUNE_1542 no *kernel* do FreeBSD. Essa opção, em sistemas onde ela se aplica, provavelmente assegura que seus discos sejam acessados de forma mais rápida e eficiente; contudo, em sistemas onde o uso desse algoritmo é inviável, pode resultar em perda de dados.

P: Posso acompanhar a série -CURRENT mesmo tendo acesso limitado à Internet?

R: Sim, é possível acompanhar a série de desenvolvimento `sem precisar` baixar sempre todo o codigo fonte do sistema, basta utilizar o recurso de CTM.

P: Como o FreeBSD foi dividido em arquivos de 240k?

R: O comando split que acompanha as novas versões dos sistemas BSD têm uma opção `-b` que permite dividir os arquivos em limites arbitrários de bytes.

Eis um exemplo tirado do `/usr/src/Makefile .`

```
bin-tarball:
(cd ${DISTDIR}; \
tar cf -- -. \
gzip ---no-name --9 --c -| \
split --b 240640 -- \
${RELEASEDIR}/tarballs/bindist/bin_tgz.)
```

P: Eu escrevi uma extensão para o *kernel*; a quem eu envio?

R: Por gentileza, consulte o artigo "Contribuindo com o Projeto FreeBSD" para obter mais informações sobre como enviar código fonte ao projeto.

E obrigado pelo seu interesse! :)

P: Como as placas Plug N Play ISA são detectadas e inicializadas?

R: Por: Frank Durda IV <uhclem@nemesis.lonestar.org >

Simplificando, existem poucas portas de E/S que todas as placas PnP respondem quando o sistema indaga se algum dispositivo está usando-a. Então, quando a rotina de procura do PnP começa, ele pergunta se há alguma placa PnP presente, e todas as placas PnP respondem com seus respectivos números de modelo para uma leitura de E/S da mesma porta. A rotina de procura recebe então um sinal wired-OR representando um "sim" à pergunta em questão. Ao menos um bit positivo constitui essa resposta. Então o código de procura é capaz de fazer com que as placas com o modelo de identificação (atribuído pela Microsoft/Intel) inferior a X sejam colocados em modo "off-line". Ele então irá verificar se alguma placa respondeu a consulta. Se a resposta for `0` o sistema assume que não há placas com identificação acima de X. Depois a rotina de busca verifica se há alguma placa cujo ID é inferior a X. Se a resposta for positiva, a rotina de busca sabe que ainda existem placas identificadas com um valor menor que X. Aí a busca tenta identificar placas com ID superior à X (limite / 4) e ordena que entrem em modo off-line. Repete-se o ciclo de pesquisas e identificações nessa forma semi-binária até que um número necessário de interações seja concluído. Ao final do processo o sistema terá identificado todas as placas PnP presentes na máquina em questão, com o número de interações necessárias sempre menor que 2^64.

Os IDs (códigos de identificação) são dois campos de 32-bits (portanto, 2^64) acrescidos de 8 bits que é o checksum (verificação de consistência de dados). Os primeiros 32 bits identificam o fabricante da placa. Nenhum fabricante assume isso, mas podemos perceber que diferentes tipos de placas do mesmo fabricante costumam ter diferentes identificações de 32-bit. O motivo correto, não se sabe, mas percebe-se que 32 bits exclusivos para os fabricantes chega a ser um exagero.

Os últimos 32 bits é um número serial que torna a identificação dessa placa única. O fabricante não pode nunca produzir uma placa que tenha os 32 bits finais iguais, a não ser que os 32 bits iniciais sejam distintos. Dessa forma é possível existir várias placas do mesmo tipo e fabricante, e ainda assim todos os 64 bits dessas placas serem únicos.

Os grupos de 32 bits nunca podem ser todos zero. Isso permite ao wired-OR identificar bits não nulos durante a procura binária inicial.

Uma vez que o sistema tenha identificado todas as IDs presentes, ele vai reativar cada placa, uma por vez (pela mesma porta de E/S) e achar os recursos que cada uma necessita, quais opções de interrupções estão disponíveis, etc. Uma busca é feita em todas as placas para obter estas informações.

Tal informação é então combinada com as informações encontradas nos arquivos ECU, no sistema, ou então da MLB BIOS. O suporte da BIOS PnP e da ECU costuma ser sintética, portanto os periféricos não são exatamente PnP como é dito. Contudo, ao examinar as informações da da BIOS e da ECU, as rotinas de busca podem

identificar dispositivos ditos PnP e evitar que eles requeiram recursos também necessários por outros dispositivos, que por sua vez não podem realocar tais valores automaticamente.

Os dispositivos PnP são visitados mais uma vez e recebem seus endereços de E/S, DMA, IRQ e endereçamentos atribuídos na memória. Os dispositivos permaneceram naquela ordem até a próxima inicialização do sistema, apesar de que nada impede que eles sejam movidos quando se desejar.

Essa explicação é muito simplista, mas provavelmente você entendeu a idéia geral do comportamento PnP.

A Microsoft fez um exame sobre algumas das portas primárias de status de impressoras para fazer PnP, dentro da lógica que nenhuma placa poderia decodificar aqueles endereços para os ciclos opostos de E/S. Eu encontrei uma placa genuína de impressora IBM que enviou dados decodificados da porta de status durante o começo do período da proposta de revisão do PnP, mas a Microsoft "ficou brava". Então eles resolveram fazer um envio para a porta de status da impressora, de forma a justar o endereço usado (naquele instante + `0x800`) e uma terceira porta de E/S para a leitura que tecnicamente pode ser localizada em qualquer lugar entre `0x200` e `0x3ff`.

P: Vocês podem definir um número principal para um driver de dispositivo que eu escrevi?

R: Isso depende se você pretende tornar o driver disponível para o público. Se sim, então por favor nos mande uma cópia do código-fonte do driver, mais as devidas modificações para o `files.i386` , um exemplo do arquivo de configuração, e os devidos códigos do MAKEDEV(8) para criar qualquer arquivo especial que seu dispositivo precise. Se você não pode, ou está impedido por causa de restrições de licença, então o character major number 32 e o block major number 8 estão reservados especificadamente para este propósito; por favor, use-os. De qualquer maneira, nós gostaríamos de obter maiores informações sobre seu driver na lista de discussão de assuntos técnicos relacionados ao FreeBSD.

P: E sobre políticas alternativas de layout de diretórios?

R: Em resposta a questão da política de formatos alternativos para diretórios, o esquema que está atualmente em uso está imutável desde quando eu o escrevi em 1983. Eu escrevi aquela política para o FFS (fast filesystem) original, e nunca o revisei. Ele funciona bem em evitar que os os grupos de cilindros sejam completamente preenchidos. Como muitos de vocês notaram, ele funciona mediocremente para procura. A maioria dos sistemas de arquivos são criados à partir de arquivos que foram criados por uma primeira procura em profundidade (depth first search, também conhecido como ftw). Estes diretórios acabam sendo distribuídos pelo grupo de cilindros, criando assim um cenário horrível em relação a futuras primeiras buscas de profundidade. Se pudéssemos saber o número total de diretórios a serem criados, a solução seria criar (total / fs_ncg) por grupo de

cilindros antes de movê-los. Evidentemente, seria necessário criar um conjunto de métodos heurísticos para adivinhar esse número. Mesmo usando um pequeno número fixo, digamos 10, ele produziria um aumento na ordem de magnitude. Para diferenciar restaurações de operações normais (quando o algoritmo atual é provavelmente mais sensível), você poderia usar o agrupamento acima de 10 se eles fossem finalizados dentro de uma janela de dez segundos. De qualquer maneira, minha conclusão é que isso é uma área pronta para experimentações.

Kirk McKusick, Setembro de 1998

P: O que fazer com os dados que eu vejo quando tenho um kernel panic?

R: *[Esta seção foi extraída de um e-mail escrito por Bill Paul na freebsd-current por Dag-Erling C. Smørgrav, que arrumou alguns problemas de impressão e adicionou os comentários entre chaves]*

```
From: Bill Paul <wpaul@skynet.ctr.columbia.edu>
Subject: Re: the fs fun never stops
To: Ben Rosengart
Date: Sun, 20 Sep 1998 15:22:50 --0400 (EDT)
Cc: current@FreeBSD.org
```

Ben Rosengart posted the following panic message]

```
> Fatal trap 12: page fault while in kernel mode
> fault virtual address   = 0x40
> fault code              = supervisor read, page not present
> instruction pointer     = 0x8:0xf014a7e5
                                ^^^^^^^^^^
> stack pointer           = 0x10:0xf4ed6f24
> frame pointer           = 0x10:0xf4ed6f28
> code segment            = base 0x0, limit 0xfffff, type 0x1b
>                         = DPL 0, pres 1, def32 1, gran 1
> processor eflags        = interrupt enabled, resume, IOPL = 0
> current process         = 80 (mount)
> interrupt mask          =
> trap number             = 12
> panic: page fault
```

[Quando] você vê uma mensagem como essa, não é suficiente somente reproduzí-la e enviá-la em um e-mail. O valor do ponteiro de instrução (instruction pointer) que eu destaquei acima é muito importante; infelizmente, ele também depende de configuração. Em outras palavras, os valores variam de acordo com a exata imagem do *kernel* que você estiver usando. Se você estiver usando uma imagem GENERIC do *kernel* de um dos *snapshots*, então é possível que alguém acompanhe a função ofensiva, mas se você está rodando um *kernel* customizado então só *você* pode nos dizer aonde a falha ocorreu.

O que você deve fazer é isso:

1. Anote o valor do ponteiro de instrução. Observe que o `0x8:` parte do começo não é significante. Nesse caso é o `0xf0xxxxxx` que nós queremos.

2. Quando o sistema reinicializar, faça o seguinte:

   ```
   % nm --n -/kernel.that.caused.the.panic -| grep f0xxxxxx
   ```

   Onde `f0xxxxxx` é o valor do ponteiro de instrução. As chances são que você não terá um resultado exato visto que os símbolos na tabela de símbolos do *kernel* são para os pontos de entrada (entry points) de funções e o endereço do ponteiro de instrução estarão em algum lugar dentro de uma função, não no começo. Se você não receber um resultado exato, omita o último dígito do valor do ponteiro de instrução e tente novamente, ex:

   ```
   % nm --n -/kernel.that.caused.the.panic -| grep f0xxxxx
   ```

   Se isso não produz nenhum resultado, corte outro dígito. Repita até que você tenha algum tipo de retorno. O resultado será uma possível lista de funções que causaram o panic. Isso é menos do que um mecanismo exato para rastreamento de um ponto de falha, mas é melhor que nada.

Eu vejo pessoas constantemente mostrando mensagens de panic como essa, mas eu raramente vejo alguém comparar o ponteiro de instrução com uma função na tabela de símbolos do *kernel*.

A melhor maneira de rastrear a causa de um panic é guardar as mensagens de falha (crash dump), e então usar o gdb(1) para gerar um stack trace da falha.

Em qualquer caso, o método que eu normalmente uso é esse:

1. Definir um arquivo de configuração do *kernel*, opcionalmente adiconando a `options DDB` se você acha que precisa do debugger do *kernel* para algo. (Eu uso isso principalmente para ajustar breakpoints se eu suspeito que há uma condição de laço infinito (infinite loop ou algo do tipo).

2. Use `config -g KERNELCONFIG` configurar o diretório da construção.

3. `cd /sys/compile/ KERNELCONFIG; make`

4. Espere o *kernel* acabar de compilar.

5. `make install`

6. reboot

O processo do make(1) terá construído dois *kernels*. `kernel` e `kernel.debug`. O `kernel` foi instalado como `/kernel`, enquanto o `kernel.debug` pode ser usado como fonte símbolos de debug para o gdb(1).

Para ter certeza que você irá capturar o crash dump, você precisa editar o `/etc/rc.conf` e ajustar o `dumpdev` para apontar para sua partição swap. Isso fará com que os scripts rc(8) usem o comando dumpon(8) para habilitar os crash dumps. Você pode também executar o dumpon(8) manualmente. Depois de um panic, o crash dump pode ser recuperado usando o savecore(8); se variavel `dumpdev` estiver definida no `/etc/rc.conf` , os scripts rc(8) irão executar o savecore(8) automaticamente e colocar o crash dump em `/var/crash` .

Nota

Os crash dumps do FreeBSD são geralmente do mesmo tamanho da memória RAM física da sua máquina. Isto é, se você tem 64MB de RAM, você terá um crash dump de 64MB. Então você deve ter certeza que há espaço suficiente em `/var/crash` para alocar o dump. Alternativamente, você executa o savecore(8) manualmente e pode fazê-lo recuperar o crash dump para onde você tenha mais espaço. É possível limitar o tamanho do crash dump utilizando a opção `options MAXMEM=(foo)` para ajustar a quantia de memória que o *kernel* irá usar para algo um pouco mais sensível. Por exemplo, se você tem 128MB de RAM, você pode limitar o uso de memória do *kernel* para 16MB para que o tamanho do seu crash dump tenha somente 16MB ao invés de 128MB.

Uma vez que você recuperou o crash dump, você pode ter um stack trace com o gdb(1) como segue:

```
% gdb --k -/sys/compile/KERNELCONFIG/kernel.debug -/var/crash/
vmcore.0
(gdb) where
```

Note que há várias telas com informações valiosas; seria ideal o uso do script(1) para capturar todas elas. Usando a imagem (unstripped) do *kernel* com todos os símbolos de debug deve mostrar a linha exata do código-fonte do *kernel* onde o panic ocorreu. Geralmente é mais interessante ler o stack trace de baixo para cima a fim de rastrear a exata seqüência de eventos que levaram ao crash. Você também pode usar o gdb(1) para exibir os conteúdos de várias variáveis ou estruturas a fim de examinar o estado do sistema no instante do crash.

Agora, se você é realmente louco e tem um segundo computador, você também pode configurar o gdb(1) para executar um debug remoto, tanto que você pode usar o gdb(1) em um sistema para debugar o *kernel* em outro sistema, incluindo o ajuste de breakpoints, rastreamento passo-a-passo pelo código do *kernel*, do mesmo modo

que você pode fazer com um programa do modo de usuário normal. Eu ainda não brinquei com isso pois não tive a chance de configurar duas lado a lado com o único propósito de debugging.

[Adendo de Bill: "Eu esqueci de mencionar uma coisa: se você tem DDB habilitado e o kernel em modo de debug, você pode forçar um panic (e um crash dump) apenas digitando ´panic ´ no prompt do ddb. Ele pode parar no debugger novamente durante a fase de panic. Se isso acontecer, digite ´continue´ e ele finalizará o crash dump."-ed]

P: Por que a dlsym() não funciona mais nos executáveis ELF?

R: A toolchain (cadeia de ferramentas) ELF não faz, por padrão, os símbolos definidos em um executável visível para o linkador dinâmico (dynamic linker). Conseqüentemente, a procura em nomes obtidos de chamadas com a `dlsym()` para `dlopen(NULL, flags)` irá falhar ao buscar tais símbolos.

Se a intenção é usar a `dlsym()` para buscar símbolos que possam existir nos executáveis principais do processo, é necessário linkar o programa com a opção `-export-dynamic` com o linker ELF (ld(1)).

P: Como eu posso aumentar ou reduzir o espaço de endereçamento disponível para o *kernel*?

R: Por padrão, o espaço de endereçamento (address space) do *kernel* é 256 MB no FreeBSD 3.x e 1 GB no FreeBSD 4.x. Em um servidor de rede com tráfego intensivo (por exemplo, um servidor FTP ou HTTP de muito tráfego) pode acontecer de, por exemplo, 256MB de memória não ser o suficiente.

Mas então, como aumentar esse espaço? Existem duas formas. Primeiro, é necessário dizer ao *kernel* que ele deve reservar uma grande quantidade de espaço em memória para ele mesmo. Segundo, considerando que o *kernel* é carregado no topo do espaço de endereçamento, é preciso diminuir o endereço de forma que não conflite com as páginas anteriores de memória, e que no lugar disso, ele seja carregado em seu novo local.

O primeiro objetivo é facilmente atingido aumentando as definições de valores do `NKPDE` no arquivo `src/sys/i386/include/pmap.h` . Aqui está o arquivo, como deve ser, para 1GB de endereço de memória:

```
#ifndef NKPDE
#ifdef SMP
#define NKPDE                   254     -/* addressable number ↺
of page tables/pde's */
#else
#define NKPDE                   255     -/* addressable number ↺
of page tables/pde's */
#endif  -/* SMP */
#endif
```

Para achar o valor correto de `NKPDE`, divida o número desejado (em megabytes) por quatro, então subtraia um para máquinas mono processadas e dois para máquinas com SMP.

Para atingir o segundo objetivo é necessário descobrir o endereço correto de carregamento. Para isso basta subtrair o tamanho do espaço de endereçamento desejado (em bytes) de 0x100100000; o resultado é 0xc0100000 para um endereço de espaço de 1 GB. Ajuste LOAD_ADDRESS em `src/sys/i386/conf/Makefile.i386` para esse valor, agora ajuste o contador de posição listada no inicio do `src/sys/i386/conf/kernel.script` para o mesmo valor, como a seguir:

```
OUTPUT_FORMAT("elf32-i386", -"elf32-i386", -"elf32-i386")
OUTPUT_ARCH(i386)
ENTRY(btext)
SEARCH_DIR(/usr/lib); SEARCH_DIR(/usr/obj/elf/home/src/tmp/usr/
i386-unknown-freebsdelf/lib);
SECTIONS
{
  -/* Read-only sections, merged into text segment: */
  -. = 0xc0100000 + SIZEOF_HEADERS;
  -.interp     -: { *(.interp)    -}
```

Agora recompile e reinstale seu *kernel*. Provavelmente aparecerão problemas com o ps(1) e com o top(1), executar um `make world` deve solucionar tais problemas (ou então, a recompilação manual da `libkvm`, do ps(1) e do top(1), depois de incluir o `pmap.h` alterado em `/usr/include/vm/ .`

OBS: o tamanho do espaço em memória do *kernel* deve ser um múltiplo de quatro megabytes.

[Adendo por David Greenman: Acho que o endereço de espaço do kernel precisa ser uma potência de dois, mas eu não estou certo disso. O código do processo de inicialização antigo costumava mexer com os bits de endereço de alta ordem, o que implicava em uma granularidade de 256MB]

Capítulo 19. Reconhecimentos

> Se você encontrar algum problema no FAQ ou se desejar submeter uma nova entrada, por favor envie um e-mail para o Responsável pelo FAQ <faq@FreeBSD.org >. Nós apreciamos o seu feedback e não podemos tornar este FAQ melhor sem a sua ajuda!
>
> —Grupo Central (FreeBSD Core Team)

Jordan K. Hubbard
: Atualizações e ajustes ocasionais na ordenação das informações do FAQ.

Doug White
: Pelos serviços prestados acima e além da chamada do dever em freebsd-questions

Jörg Wunsch
: Pelos serviços prestados acima e além da chamada do dever na Usenet

Garrett Wollman
: Rede (*networking*) e formatação

Jim Lowe
: Informações sobre “Multicast”

Peter da Silva
: Escravo de digitação mecânica do FAQ FreeBSD

A Equipe FreeBSD
: Reclamações, rabujices e envio de informações

E a todos os outros que nós nos esquecemos, nossas desculpas e obrigado do coração!