

A FAQ (perguntas freqüentes) do Debian GNU/Linux.

'Autores' on page [71](#)

Versão CVS, 11 November 2001

Resumo

Este documento responde perguntas freqüentes sobre o Debian GNU/Linux.

Nota de Copyright

Copyright © 1996-2000 por Software in the Public Interest.

É concedida permissão para fazer e distribuir cópias exatas deste documento, desde que, esta nota de copyright seja preservada em todas as cópias.

É concedida permissão para copiar e distribuir versões modificadas deste documento sob as condições de cópia exata, contanto que os trabalhos derivados deste, sejam distribuídos sob termos idênticos a estes.

É concedida permissão para copiar e distribuir traduções deste documento em outros idiomas, sob as condições para versões modificadas, declarada acima, exceto pelo fato de que esta permissão deve ser incluída na tradução autorizada pela Free Software Foundation, ao invés do original em inglês.

Tradução para o idioma português (Brasil):

- Tradutor: Desconhecido
- Revisor: João Alberto de França Ferreira <joaoff@esquadro.com.br>

Qualquer comentário, sugestão, correção etc, devem ser enviados ao tradutor e revisor.

Sumário

1	Visão geral e definições.	1
1.1	O que é o Debian GNU/Linux?	1
1.2	OK, agora eu sei o que é o Debian... o que é Linux!?.	2
1.3	O que é essa coisa chamada "Hurd"?	3
1.4	Qual é a diferença entre o Debian GNU/Linux e outras distribuições Linux? Por que eu deveria escolher o Debian ao invés de outra distribuição?	3
1.5	Como o projeto Debian se encaixa (no) ou se compara com o projeto GNU da Free Software Foundation?	4
1.6	Como se pronuncia Debian e o que essa palavra significa?	4
2	Adquirindo e Instalando o Debian GNU/Linux.	5
2.1	Qual é a última versão do Debian?	5
2.2	Onde/como posso adquirir os discos de instalação do Debian?	5
2.3	Como eu instalo o Debian de CD-ROMs?	6
2.4	Eu tenho meu próprio gravador de CDs, há imagens de CD disponíveis em algum lugar?	6
2.5	Posso instalá-lo a partir de disquetes?	6
2.6	Posso pegar e instalar o Debian diretamente de um site remoto?	7
2.7	Posso adquirir e instalar o Debian por fita?	8
3	Questões sobre compatibilidade.	9
3.1	Em quais arquiteturas de hardware o Debian GNU/Linux roda?	9
3.2	Quão compatível é o Debian com outras distribuições de Linux?	10
3.3	Quão compatível, em nível de código fonte, o Debian é em relação a outros sistemas Unix?	10

3.4	Posso usar pacotes Debian (arquivos “.deb”) em meu sistema Red Hat/Slackware/...? Posso usar pacotes Red Hat (arquivos “.rpm”) em meu sistema Debian GNU/Linux?	10
3.5	O Debian pode rodar meus antiqüíssimos programas “a.out”?	11
3.6	O Debian consegue rodar meus antigos programas em libc5?	12
3.7	O Debian pode ser usado para compilar programas em libc5?	12
3.8	Como devo instalar um pacote não-Debian?	12
3.9	Por que aparece “Can’t find libX11.so.6” (“Impossível encontrar libX11.so.6”) quando tento rodar foo?	13
3.10	Por que não consigo compilar programas que exigem a libtermcap?	13
3.11	Por que não consigo instalar o AccelX?	13
3.12	Por que minhas velhas aplicações XFree 2.1 Motif travam?	13
4	Software disponível no sistema Debian.	15
4.1	Que tipos de aplicativos e software de desenvolvimento estão disponíveis para o Debian GNU/Linux?	15
4.2	Quem escreveu todos esses programas?	16
4.3	Como posso conseguir uma lista atual dos programas que foram empacotados para o projeto Debian?	16
4.4	Por que eu obtenho a mensagem “ld: cannot find -lfoo” quando compilo programas? Por que não existe nenhum arquivo libfoo.so nos pacotes de bibliotecas Debian?	16
4.5	O que está faltando no Debian GNU/Linux?	16
4.6	(Como) O Debian suporta Java?	17
4.7	Como posso verificar se estou usando um sistema Debian, e que versão é esta?	17
4.8	Como o Debian suporta idiomas diferentes do Inglês?	18
4.9	E as limitações de exportação dos EUA?	18
4.10	Onde está o Pine?	19
5	Os repositórios FTP do Debian.	21
5.1	O que são todos aqueles diretórios nos repositórios FTP do Debian?	21
5.2	Quantas distribuições Debian existem no diretório dists?	21
5.3	O que são todos esses nomes, como “slink”, “potato”, etc.?	21
5.4	E sobre a “frozen” (paralizada)?	22

5.5	Que outros codinomes foram usados no passado?	22
5.6	De onde vieram esses codinomes?	22
5.7	E sobre a “sid”?	22
5.7.1	Notas históricas sobre a “sid”.	23
5.8	O que o diretório “stable” contém?	23
5.9	O que o diretório “testing” contém?	23
5.10	O que o diretório “unstable” contém?	24
5.11	O que são todos aqueles diretórios dentro de <code>dists/stable/main</code> ?	24
5.12	Onde está o código-fonte?	25
5.13	O que existe no diretório <code>pool</code> ?	25
5.14	Ok, e sobre os outros diretórios, que não estão dentro de <code>dists</code> ou <code>pool</code> ?	26
6	Fundamentos do Sistema de Gerenciamento de Pacotes Debian	27
6.1	O que é um pacote Debian?	27
6.2	Qual é o formato de um pacote binário Debian?	28
6.3	Por que os nomes de pacotes Debian são tão longos?	28
6.4	O que é um arquivo de controle Debian?	29
6.5	O que é um <code>conffile</code> Debian?	30
6.6	O que são os scripts Debian <code>preinst</code> , <code>postinst</code> , <code>prepm</code> e <code>postrm</code> ?	30
6.7	O que é um pacote Required/Important/Standard/Optional/Extra?	31
6.8	O que é um pacote virtual?	32
6.9	O que significa dizer que um arquivo <code>Depends/Recommends/Suggests/Conflicts/Replaces/Provides</code> (Depende/Recomenda/Sugere/Conflita/Substitui/Oferece) outro pacote?	32
6.10	O que significa Pré-Dependência?	33
6.11	O que significa <code>unknown/install/remove/purge/hold</code> (desconhecido/instalar/remover/remover por completo/conservar) no estado do pacote?	33
6.12	Como coloco um pacote em “hold”?	34
6.13	Como instalo um pacote de fontes?	34
6.14	Como construo pacotes binários a partir de um pacote de códigos fonte?	35
6.15	Como eu crio pacotes Debian?	35

7	As Ferramentas de Gerenciamento de Pacotes Debian.	37
7.1	Que programas o Debian oferece para administrar seus pacotes?	37
7.1.1	dpkg	37
7.1.2	dselect	38
7.1.3	dpkg-deb	40
7.1.4	apt-get	40
7.1.5	dpkg-split	41
7.2	O Debian alega ser capaz de atualizar um programa que está sendo executado; como isto é realizado?	41
7.3	Como posso descobrir que pacotes estão instalados num sistema Debian?	41
7.4	Como posso descobrir que pacote produziu um arquivo específico?	42
8	Mantendo seu Sistema Debian Atualizado.	43
8.1	Como posso atualizar minha distribuição Debian 1.3.1 (ou anterior), baseada na libc5, para a 2.0 (ou posterior), baseada na libc6?	43
8.2	Como posso manter meu sistema Debian atualizado?	44
8.2.1	APT	44
8.2.2	dpkg-ftp	45
8.2.3	mirror	45
8.2.4	dftp	45
8.2.5	dpkg-mountable	46
8.3	Preciso mudar para o modo monousuário para atualizar um pacote?	46
8.4	Preciso manter todos aqueles arquivos .deb em meu disco?	46
8.5	Como posso manter um registro dos pacotes que adicionei ao sistema?	46
9	Debian e o kernel.	49
9.1	Posso instalar e compilar um kernel sem alguma configuração específica para o Debian?	49
9.2	Que ferramentas o Debian oferece para a construção de kernels personalizados?	49
9.3	Como faço um disquete de inicialização personalizado?	51
9.4	Que ferramentas especiais o Debian oferece para lidar com módulos?	51
9.5	Posso desinstalar um kernel antigo de forma segura? Como?	51

10 Personalizando sua instalação do Debian GNU/Linux.	53
10.1 Como posso garantir que todos os programas usem o mesmo tamanho de papel?	53
10.2 Como posso fornecer acesso aos periféricos sem comprometer a segurança?	53
10.3 Como carrego uma fonte de console na inicialização à moda Debian?	53
10.4 Como configuro os defaults de um programa para X11?	54
10.5 Parece que cada distribuição possui um método de inicialização diferente. Fale-me sobre o do Debian.	54
10.6 Parece que o Debian não usa o arquivo <code>rc.local</code> para personalizar o processo de inicialização; que recursos são fornecidos?	55
10.7 Como o sistema de gerenciamento de pacotes lida com pacotes que contêm arquivos de configuração para outros pacotes?	55
10.8 Como posso anulo um arquivo instalado por um pacote, de modo que uma versão diferente possa ser usada?	56
10.9 Como posso incluir meu pacote feito localmente, na lista de pacotes disponíveis que o sistema de gerenciamento de pacotes usa?	57
10.10 Alguns usuários gostam do <code>mawk</code> , outros do <code>gawk</code> ; alguns gostam do <code>vim</code> , outros do <code>elvis</code> ; alguns gostam do <code>trn</code> , outros do <code>tin</code> ; como o Debian lida com diversidade?	57
11 Obtendo suporte para o Debian GNU/Linux.	59
11.1 Que outra documentação existe em/para um sistema Debian?	59
11.2 Existem recursos on-line para discussão sobre o Debian?	60
11.3 Qual é o código de conduta para as listas de discussão?	61
11.3.1 Política sobre anúncios em listas de discussão.	61
11.4 Existe uma maneira rápida de procurar informações sobre o Debian GNU/Linux?	62
11.5 Existem registros de bugs conhecidos?	62
11.6 Como posso informar sobre um bug no Debian?	62
12 Contribuindo para o projeto Debian.	65
12.1 Como posso me tornar um desenvolvedor de software Debian?	65
12.2 Como posso contribuir com recursos para o projeto Debian?	65
12.3 Como posso contribuir financeiramente com o projeto Debian?	66
12.3.1 Software in the Public Interest	66
12.3.2 Free Software Foundation	66

13 Redistribuindo o Debian GNU/Linux em um produto comercial.	67
13.1 Posso fazer e vender CDs Debian?	67
13.2 O Debian pode ser empacotado com software não-livre?	67
13.3 Estou fazendo uma distribuição especial do Linux para um “mercado vertical”. Posso usar o Debian GNU/Linux como núcleo do sistema Linux e acrescentar meus próprios aplicativos em cima dele?	68
13.4 Posso colocar meu programa proprietário num “pacote” Debian, para que ele possa ser instalado facilmente em qualquer sistema Debian?	68
14 Que tipo de mudanças são esperadas para a próxima versão do Debian GNU/Linux?	69
14.1 Mais segurança.	69
14.2 Mais suporte para usuários que não possuem Inglês como idioma nativo.	69
14.3 Mais arquiteturas.	70
14.4 Kernel Linux 2.4.	70
14.5 A Biblioteca C 2.2 da GNU.	70
15 Informações gerais sobre o FAQ.	71
15.1 Autores	71
15.2 Feedback	72
15.3 Disponibilidade	72
15.4 Formato do documento	72

Capítulo 1

Visão geral e definições.

1.1 O que é o Debian GNU/Linux?

Debian GNU/Linux é uma *distribuição* particular do sistema operacional Linux, e numerosos pacotes que rodam nele.

Em princípio, os usuários poderiam obter o kernel (núcleo do sistema operacional) do Linux através da Internet ou de qualquer outro lugar e compilá-lo por si próprios. Eles poderiam então, obter o código fonte de vários aplicativos da mesma maneira, compilar os programas e então, instalá-los em seus sistemas. Para programas complexos, este processo pode ser não apenas demorado, mas também, suscetível a erros. Para evitar isto, os usuários frequentemente escolhem obter o sistema operacional e os pacotes de aplicativos de um dos distribuidores Linux. O que distingue os vários distribuidores Linux são os programas, protocolos e as práticas usadas para empacotar, instalar e monitorar pacotes de aplicativos nos sistemas dos usuários, combinados com a instalação e as ferramentas de manutenção, documentação e outros serviços.

O Debian GNU/Linux é o resultado de um esforço voluntário para criar um sistema operacional livre, de alta qualidade, compatível com Unix e completo, com um conjunto de aplicativos. A idéia de um sistema livre compatível com o Unix originou-se do projeto GNU, e muitos dos aplicativos que fazem do Debian GNU/Linux tão útil, foram desenvolvidos pelo projeto GNU.

Para o Debian, *livre* tem o significado definido pela GNU (veja As Linhas-Guia do Software Livre Debian (http://www.br.debian.org/social_contract#guidelines)). Quando falamos de software livre, estamos nos referindo a liberdade e não ao preço. *Software livre* significa que você tem a liberdade de distribuir cópias de software livre, que você recebe o código-fonte ou pode adquiri-lo se quiser, que você pode alterar o software ou usar partes dele em novos programas livres; e para saber que você pode fazer estas coisas.

O Projeto Debian foi criado por Ian Murdock em 1993, inicialmente patrocinado pelo projeto GNU da Free Software Foundation. Hoje, os projetistas do Debian entendem-no como um descendente direto do projeto GNU.

O Debian GNU/Linux é:

- **completo:** O Debian atualmente inclui mais de 4200 pacotes de software. Os usuários podem escolher quais pacotes instalar; o Debian provê uma ferramenta para este propósito. Você pode encontrar uma lista e descrições dos pacotes atualmente disponíveis no Debian em qualquer um dos sites espelhos do Debian (<http://www.br.debian.org/distrib/ftplist.html>).
- **livre para usar e redistribuir:** Não é exigido nenhum pagamento para participar de sua distribuição e desenvolvimento. Todos os pacotes que são formalmente parte do Debian GNU/Linux são livres para redistribuição, geralmente sob os termos especificados pela Licença Pública Geral GNU (GNU General Public License).
Os repositórios FTP do Debian também contêm aproximadamente 450 pacotes de software (nos diretórios `non-free` e `contrib`), que são distribuídos sob os termos específicos incluídos em cada pacote.
- **dinâmico:** com aproximadamente 800 voluntários contribuindo constantemente com novos e melhores códigos, o Debian está evoluindo rapidamente. Novas versões são periodicamente planejadas com alguns meses de intervalo, e os repositórios FTP são atualizados diariamente.

Embora o próprio Debian GNU/Linux seja software livre, ele é uma base sobre a qual novas distribuições podem ser construídas. Proporcionando um sistema básico confiável e completo, o Debian proporciona aos usuários de Linux maior compatibilidade, e permite que criadores de distribuições Linux eliminem a duplicação de esforços e se concentrem nas coisas que tornam suas distribuições especiais. veja 'Estou fazendo uma distribuição especial do Linux para um "mercado vertical". Posso usar o Debian GNU/Linux como núcleo do sistema Linux e acrescentar meus próprios aplicativos em cima dele?' on page 68 para mais informações.

1.2 OK, agora eu sei o que é o Debian... o que é Linux!?

Resumidamente, Linux é o kernel (núcleo) de um sistema operacional estilo Unix. Ele foi originalmente projetado para PCs 386 (e melhores); agora, suporte a outros sistemas, incluindo os sistemas multi-processados, estão em desenvolvimento. O Linux é desenvolvido por Linus Torvalds e por vários cientistas de computação ao redor do mundo.

Além do kernel, um sistema "Linux" geralmente possui:

- um sistema de arquivos que segue o Padrão de Hierarquia para o Sistema de Arquivos Linux (Linux Filesystem Hierarchy Standard (<http://www.pathname.com/fhs/>)).
- uma vasta gama de utilitários Unix, muitos dos quais foram desenvolvidos pelo projeto GNU e pela Free Software Foundation.

A combinação do kernel Linux, do sistema de arquivos, dos utilitários GNU e FSF, e dos outros utilitários, são projetados para alcançar conformidade com o padrão POSIX (IEEE 1003.1); veja

‘Quão compatível, em nível de código fonte, o Debian é em relação a outros sistemas Unix?’ on page 10.

Para maiores informações sobre o Linux, veja a INFO-SHEET (<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/INFO-SHEET>) de Michael K. Johnson e o META-FAQ (<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/META-FAQ>).

1.3 O que é essa coisa chamada “Hurd”?

O Hurd é um conjunto de servidores que rodam sobre o microkernel GNU Mach. Juntos, eles constroem a base para o sistema operacional GNU.

Atualmente, o Debian está apenas disponível para Linux, mas com o Debian GNU/Hurd, nós começamos a oferecer o Hurd como um desenvolvimento, servidor e plataforma, também. Porém, o Debian GNU/Hurd não foi lançado oficialmente ainda, e não será ainda por algum tempo.

Por favor, veja <http://www.gnu.org/software/hurd/> para mais informações sobre o GNU/Hurd em geral, e <http://www.br.debian.org/ports/hurd/> para mais informações sobre o Debian GNU/Hurd.

1.4 Qual é a diferença entre o Debian GNU/Linux e outras distribuições Linux? Por que eu deveria escolher o Debian ao invés de outra distribuição?

Estas características chaves distinguem o Debian de outras distribuições Linux:

O sistema de manutenção de pacotes Debian: O sistema inteiro, ou qualquer componente individual dele, pode ser atualizado sem necessidade de formatação, sem perda de arquivos de configuração personalizados e (na maioria dos casos) sem reinicializar o sistema. A maioria das distribuições Linux disponíveis hoje, têm algum tipo de sistema de manutenção de pacotes; o sistema de manutenção de pacotes Debian é único e particularmente robusto. (veja ‘Fundamentos do Sistema de Gerenciamento de Pacotes Debian’ on page 27)

Desenvolvimento aberto: Considerando que outras distribuições Linux são desenvolvidas por indivíduos, grupos pequenos e fechados ou entidades comerciais, o Debian é a única distribuição Linux que está sendo desenvolvida de forma cooperativa por vários indivíduos através da Internet, no mesmo espírito do Linux e de outros programas livres.

Mais de 800 mantenedores voluntários de pacotes estão trabalhando em mais de 4200 pacotes e melhorando o Debian GNU/Linux. Os desenvolvedores Debian contribuem para o projeto não apenas criando novos aplicativos (na maioria dos casos), mas empacotando o software existente de acordo com os padrões do projeto, enviando relatórios de bugs

(falhas nos programas) para os desenvolvedores oficiais e provendo apoio ao usuário. Veja informações adicionais em ‘Como posso me tornar um desenvolvedor de software Debian?’ on page 65.

O Sistema de Monitoramento de Bugs: A dispersão geográfica dos desenvolvedores Debian exigiu sofisticadas ferramentas e comunicação rápida de bugs e suas soluções para acelerar o desenvolvimento do sistema. Os usuários são estimulados a enviar bugs em um estilo formal, que é rapidamente acessível pela WWW ou via e-mail. Veja informações adicionais nesta FAQ (Frequently Asked Questions - Perguntas Feitas Frequentemente) sobre a administração do ‘Existem registros de bugs conhecidos?’ on page 62.

A Política do Debian: O Debian possui uma especificação extensa de nossos padrões de qualidade, a Política do Debian (Debian Policy). Esse documento define as qualidades e padrões aos quais atrelamos os pacotes Debian.

Para informações adicionais sobre isto, por favor, veja nossa página web sobre razões para escolher o Debian (http://www.br.debian.org/intro/why_debian).

1.5 Como o projeto Debian se encaixa (no) ou se compara com o projeto GNU da Free Software Foundation?

O sistema Debian foi baseado nos ideais de software livre originalmente defendidos pela Free Software Foundation (<http://www.gnu.org/>) e em particular por Richard Stallman. As poderosas ferramentas de desenvolvimento, utilitários e aplicativos da FSF são também parte importante do sistema Debian.

O Projeto Debian é uma entidade separada da FSF, porém, nos comunicamos regularmente e cooperamos em vários projetos. A FSF pediu explicitamente que chamássemos nosso sistema de “Debian GNU/Linux”, e ficamos contentes em atender a este pedido.

O objetivo de longo prazo da FSF é desenvolver um novo sistema operacional chamado GNU, baseado no HURD (<http://www.gnu.org/software/hurd/hurd.html>).

1.6 Como se pronuncia Debian e o que essa palavra significa?

O nome do projeto é pronunciado como Deb'-ian, com ênfase na primeira sílaba. Esta palavra é uma contração dos nomes Debra e Ian Murdock, sendo este último, o fundador do projeto. (Dicionários parecem oferecer alguma ambigüidade na pronúncia de Ian, mas o mesmo prefere ee'-um)

Capítulo 2

Adquirindo e Instalando o Debian GNU/Linux.

2.1 Qual é a última versão do Debian?

Atualmente há duas versões do Debian GNU/Linux:

Versão 2.2, também conhecida como distribuição 'stable' (estável) Esta distribuição é estável e contém programas testados, ela mudará se ajustes de segurança e usabilidade forem feitos.

A distribuição 'unstable' (instável) Esta é a versão que está em desenvolvimento atualmente; ela é atualizada continuamente. Você pode pegar pacotes da versão 'unstable' em qualquer site de FTP do Debian e usá-los para atualizar seu sistema a qualquer hora. Mas você não pode esperar que o sistema seja tão bom e estável quanto antes - por isso essa versão é chamada de '**unstable**'!

Por favor, veja 'Quantas distribuições Debian existem no diretório dists?' on page 21 para mais informação.

2.2 Onde/como posso adquirir os discos de instalação do Debian?

Você pode adquirir os discos de instalação baixando os arquivos apropriados do site FTP do Debian: <ftp://ftp.br.debian.org/debian/> e de seus espelhos (<http://www.debian.org/distrib/ftplist.html>).

Os arquivos de instalação do sistema estão separados em subdiretórios, dentro do diretório `dists/stable/main`, e os nomes destes subdiretórios correspondem a cada uma das arquiteturas suportadas, como exemplo: `disks-arch` (onde *arch* é entendido como sendo a

arquitetura - i386, sparc, arm etc - confira no site uma lista exata). Em cada um desses subdiretórios, pode haver vários diretórios, cada um para uma versão do sistema de instalação, e a versão atualmente usada, está no diretório 'current' (que é uma ligação simbólica).

Veja o arquivo `readme.txt` naquele diretório, para informações adicionais.

2.3 Como eu instalo o Debian de CD-ROMs?

O Linux suporta o sistema de arquivos ISO 9660 (CD-ROM) com extensões Rock Ridge (anteriormente conhecido por "High Sierra"). Alguns distribuidores (<http://www.br.debian.org/distrib/vendors>) fornecem o Debian GNU/Linux neste formato.

Aviso: na instalação por CD-ROM, normalmente não é uma boa idéia escolher o método de acesso `cd-rom` do `dselect`. Este método é normalmente muito lento. Os métodos `mountable` e `apt`, por exemplo, são muito melhores para instalar a partir de CD-ROM (veja 'dpkg-mountable' on page 46 e 'APT' on page 44).

2.4 Eu tenho meu próprio gravador de CDs, há imagens de CD disponíveis em algum lugar?

Sim. Para facilitar aos distribuidores de CDs o fornecimento de discos de alta qualidade, nós fornecemos as imagens dos CDs Oficiais (<http://cdimage.debian.org/>).

2.5 Posso instalá-lo a partir de disquetes?

Primeiramente, um aviso: O sistema Debian GNU/Linux inteiro é muito grande para ser instalado por mídias tão pequenas quanto disquetes de 1.44MB padrão - você não achará que a instalação por disquetes seja uma experiência muito agradável.

Copie os pacotes Debian em disquetes formatados. Qualquer dos formatos DOS, ext2 (o nativo do Linux), ou minix servirá; basta usar um comando `mount` apropriado para o disquete sendo usado.

O uso de disquetes possui estas complicações:

- Nomes curtos do MS-DOS: se você está tentando colocar arquivos de pacotes Debian em discos formatados para MS-DOS, verá que seus nomes são geralmente muito longos, e não estão em conformidade com as limitações de nome de arquivos do MS-DOS. Para superar isto, você terá que usar discos formatados com o sistema de arquivos VFAT, visto que o VFAT suporta maiores nomes de arquivos.
- Arquivos grandes: alguns pacotes são maiores que 1.44 MBytes, e não caberão em apenas um disquete. Para resolver este problema, use a ferramenta `dpkg-split` (veja 'dpkg-split' on page 41), disponível no diretório `tools` em <ftp://ftp.br.debian.org/>

debian/ e em seus espelhos (<http://www.br.debian.org/distrib/ftplist.html>).

Você deve ter suporte a disquetes no kernel para poder ler e gravar no disquete; a maioria dos kernels inclui o suporte a disquetes.

Para montar um disquete sob o ponto de montagem `/floppy` (um diretório que deve ter sido criado durante a instalação), use:

- `mount -t msdos /dev/fd0 /floppy/`
se o disquete estiver no drive A: e possui um sistema de arquivos MS-DOS,
- `mount -t msdos /dev/fd1 /floppy/`
se o disquete estiver no drive B: e possui um sistema de arquivos MS-DOS,
- `mount -t ext2 /dev/fd0 /floppy/`
se o disquete estiver no drive A: e possui um sistema de arquivos ext2 (i.e., o normal do Linux).

2.6 Posso pegar e instalar o Debian diretamente de um site remoto?

Sim. Há várias maneiras de se fazer isso:

- **Usando APT:** instale o pacote `apt` e edite o arquivo `/etc/apt/sources.list`. Se você deseja instalar a versão estável, você provavelmente irá gostar de usar a seguinte fonte:

```
http://http.us.debian.org/debian stable main contrib non-free
```

Você pode substituir `us.debian.org` pelo nome de um mirror mais rápido perto de você. Detalhes sobre isso podem ser encontrados nas páginas de manuais `apt-get(8)` e `sources.list(8)`, assim como no Guia de Uso do APT (APT User's Guide), que se encontra em `/usr/share/doc/apt/guide.html/index.html`.

Então rode

```
apt-get update
```

seguido de

```
apt-get dist-upgrade
```

e seu sistema será atualizado para a mais nova versão do Debian.

- **Usando o dselect com o dpkg-ftp** Esse é um método desaprovado, instale o pacote `dpkg-ftp` (veja 'dpkg-ftp' on page 45).

Depois chame o programa `dselect`, que irá invocar o programa `dpkg-ftp`, que o guiará através da seleção de pacotes, e instalará os mesmos, sem baixá-los para sua máquina. Este método foi projetado para salvar espaço de disco e quanto tempo do usuário.

O `dselect` irá perguntá-lo sobre o nome de domínio do site de ftp anônimo que você pretende usar (se você não souber, tente `ftp.debian.org` ou `http.us.debian.org`).

Ele irá perguntar também, pelo diretório ou subdiretórios que contém os arquivos que você quer instalar. Esse diretório deve conter o arquivo chamado `Packages` (ou seu equivalente comprimido, `Packages.gz`), geralmente `dists/stable/main/binary-ARCH`, onde `ARCH` é entendido como a arquitetura da sua máquina.

Note que quando você obtém e instala os pacotes, você ainda os possui no seu diretório `/var`, portanto, lembre-se de excluir esses arquivos extras, ou de movê-los para outro local (dica: use `apt-move`), ou o espaço na sua partição pode acabar.

2.7 Posso adquirir e instalar o Debian por fita?

No momento, não há suporte para instalar pacotes diretamente de fitas. Entretanto, pode-se usar `tar`, `cpio`, ou `afio` para copiar os arquivos de pacotes Debian numa fita, e depois copiá-los no seu disco local para instalação. Da mesma maneira, disquetes contendo arquivos "tar" devem ser copiados num disco local antes que eles possam ser gerenciados pelas ferramentas de pacotes do Debian.

Capítulo 3

Questões sobre compatibilidade.

3.1 Em quais arquiteturas de hardware o Debian GNU/Linux roda?

O Debian GNU/Linux inclui os códigos-fonte completos para todos os programas inclusos, portanto, eles devem rodar em qualquer sistema que seja suportado pelo kernel Linux; veja a FAQ do Linux (<http://www.linuxdoc.org/FAQ/Linux-FAQ/>) para detalhes.

A versão atual do Debian GNU/Linux , a 2.2, contém uma completa distribuição binária para as seguintes arquiteturas:

i386: essa cobre os PCs baseados em Intel e processadores compatíveis, incluindo Intel 386, 486, Pentium, Pentium Pro, Pentium II (tanto Klamath quanto Celeron), e Pentium III, e os processadores compatíveis da AMD, Cyrix e outros, compatíveis com estes.

m68k: essa cobre as máquinas Amiga e Atari que possuem processadores Motorola 680x0 (para $x \geq 2$); com MMU.

alpha: sistemas Alpha da Compaq/Digital.

sparc: essa cobre os sistemas SPARC da Sun e a maioria dos sistemas UltraSPARC.

powerpc: essa cobre algumas máquinas IBM/Motorola PowerPC, incluindo máquinas CHRP, PowerMac e PReP.

arm: máquinas ARM e StrongARM.

O desenvolvimento de distribuições binárias do Debian para as arquiteturas Sparc64 (UltraSPARC nativo) e MIPS estão em andamento.

Para informações adicionais na carga (boot), particionamento do drive, habilitando dispositivos PCMCIA (PC Card) e questões similares, por favor, siga as instruções contidas no Manual de Instalação, o qual está disponível no nosso site web em <http://www.br.debian.org/releases/stable/i386/install>.

3.2 Quão compatível é o Debian com outras distribuições de Linux?

Desenvolvedores Debian se comunicam com outros criadores de distribuições Linux para manter compatibilidade binária entre as distribuições. A maior parte dos produtos comerciais para Linux roda tão bem no Debian quanto no sistema sobre o qual foram construídos.

O Debian GNU/Linux adere à Estrutura de Sistema de Arquivos do Linux (Linux File System Structure) (<http://www.pathname.com/fhs/>). Apesar disso, existe espaço para interpretações em algumas das regras dentro desse padrão, então podem existir diferenças entre um sistema Debian e outros sistemas Linux. A mais nova versão desse padrão FSSTND é chamada de FHS e planejamos mudar para ela em pouco tempo.

3.3 Quão compatível, em nível de código fonte, o Debian é em relação a outros sistemas Unix?

Para a maior parte dos aplicativos, o código fonte do Linux é compatível com outros sistemas Unix. Ele tem suporte a praticamente tudo o que está disponível em sistemas Unix System V e derivados livres e comerciais do BSD. No mundo Unix, porém, essa afirmação praticamente não tem valor, pois não há como prová-la. Na área de desenvolvimento de software é necessário haver compatibilidade completa ao invés de compatibilidade na “maioria” dos casos. Por causa disso, anos atrás surgiu a necessidade de padrões, e hoje em dia o POSIX.1 (IEEE Standard 1003.1-1990) é um dos principais padrões para compatibilidade em nível de código fonte entre sistemas operacionais estilo Unix.

O Linux pretende aderir ao POSIX.1, mas os padrões POSIX custam muito dinheiro e o certificado POSIX.1 (e FIPS 151-2) é realmente caro; isso dificulta o trabalho de conformidade completa com o POSIX por parte dos desenvolvedores do Linux. Os custos da certificação fazem com que seja improvável que a Debian receba um certificado de conformidade oficial mesmo que ela passe completamente pelo conjunto de validação. (O conjunto de validação agora está disponível livremente, portanto espera-se que mais pessoas trabalhem em questões acerca do POSIX.1.)

A Unifix GmbH (Braunschweig, Alemanha) desenvolveu um sistema Linux que recebeu certificação de conformidade ao FIPS 151-2 (um superconjunto do POSIX.1). Essa tecnologia esteve disponível na distribuição da própria Unifix, chamada Unifix Linux 2.0, e no Linux-FT da Lasermoon.

3.4 Posso usar pacotes Debian (arquivos “.deb”) em meu sistema Red Hat/Slackware/...? Posso usar pacotes Red Hat (arquivos “.rpm”) em meu sistema Debian GNU/Linux?

Diferentes distribuições Linux usam diferentes formatos de pacotes e programas de gerenciamento de pacotes.

Você provavelmente pode: Existe um programa para descompactar um pacote Debian num computador com outra distribuição do Linux, e normalmente irá funcionar no sentido de que os arquivos vão ser descompactados. A recíproca provavelmente também é verdadeira, ou seja, um programa para descompactar um pacote do Red Hat ou Slackware num computador baseado no Debian Linux provavelmente conseguirá descompactar o pacote e colocar a maior parte dos arquivos nos lugares corretos. Isso é consequência da existência do (e grande conformidade com o) Padrão de Hierarquia para o Sistema de Arquivos do Linux (Linux File System Standard).

Você provavelmente não quer: a maior parte dos gerenciadores de pacotes escrevem arquivos de administração quando são usados para descompactar um arquivo. Esses arquivos de administração geralmente não são padronizados. Portanto, o efeito de descompactar um pacote Debian em um computador “estranho” irá gerar efeitos imprevisíveis (certamente improdutivos) no gerenciador de pacotes daquele sistema. Do mesmo modo, utilitários de outras distribuições podem conseguir descompactar seus arquivos em sistemas Debian, mas provavelmente farão o sistema de gerenciamento de pacotes da Debian falhar quando chegar a hora de atualizar ou remover alguns pacotes, ou simplesmente listar exatamente quais pacotes estão presentes em um sistema.

Uma maneira melhor: o Padrão de Hierarquia para o Sistema de Arquivos do Linux (e portanto Debian GNU/Linux) exige que subdiretórios sob `/usr/local/` fiquem completamente sob a responsabilidade do usuário. Portanto, os usuários podem descompactar pacotes “estranhos” dentro desse diretório, e gerenciar sua própria configuração, atualização e remoção individualmente.

3.5 O Debian pode rodar meus antiqüíssimos programas “a.out”?

Você ainda tem um programa assim? :-)

Para *executar* um programa cujo binário esteja em formato `a.out` (isto é, QMAGIC ou ZMAGIC),

- Assegure-se que seu kernel tenha suporte a `a.out`, seja diretamente (`CONFIG_BINFMT_AOUT=y`) ou como módulo (`CONFIG_BINFMT_AOUT=m`). (O pacote `kernel-image` do Debian possui o módulo `binfmt_aout`)

Se seu kernel suporta binários `a.out` por módulo, garanta que o módulo `binfmt_aout` esteja carregado. Você pode fazer isso durante a inicialização incluindo a linha `binfmt_aout` no arquivo `/etc/modules`. Você pode fazer isso a partir da linha de comando executando o comando `insmod NOMEDIR/binfmt_aout.o` onde `NOMEDIR` é o nome do diretório onde os módulos que foram construídos para a versão da kernel sendo rodada estão armazenados. Em um sistema com a versão 2.0.36 da kernel, `NOMEDIR` provavelmente é `/lib/modules/2.2.17/fs/`.

- Instale o pacote `libc4`, encontrado em versões anteriores a versão 2.0 (porque naquele momento nós removemos o pacote). Você pode querer dar uma olhada num CD-ROM

antigo do Debian (o Debian 1.3.1 ainda tinha esse pacote), ou veja <ftp://archive.debian.org/debian-archive/> na internet.

- Se o programa que você quer executar for um cliente X em a.out, então instale o pacote xcompat (veja acima sobre a disponibilidade).

Se você possui um aplicativo comercial em a.out, este seria um bom momento para pedir que enviem uma atualização em ELF.

3.6 O Debian consegue rodar meus antigos programas em libc5?

Sim. Basta instalar as bibliotecas libc5 necessárias, da seção oldlibs (contendo pacotes antigos incluídos para compatibilidade com aplicativos antigos).

3.7 O Debian pode ser usado para compilar programas em libc5?

Sim. Instale os pacotes libc5-altdev e altgcc (da seção oldlibs). Você pode achar os programas gcc e g++ compilados em libc5 no diretório /usr/i486-linuxlibc1/bin. Coloque-os em sua variável \$PATH para que o make e outros programas o executem primeiro.

Se você precisa compilar clientes X em libc5, instale os pacotes xlib6 e xlib6-altdev.

Esteja ciente de que o ambiente libc5 não é mais totalmente suportado por nossos outros pacotes.

3.8 Como devo instalar um pacote não-Debian?

Arquivos sob o diretório /usr/local/ não estão sob controle do sistema de gerenciamentos de pacotes Debian. Assim sendo, é boa prática colocar o código fonte de seu programa em /usr/local/src/. Por exemplo, você pode extrair os arquivos de um pacote chamado "foo.tar" dentro do diretório /usr/local/src/foo. Depois de compilá-lo, coloque os binários em /usr/local/bin/, as bibliotecas em /usr/local/lib, e os arquivos de configuração em /usr/local/etc.

Se seus programas e/ou arquivos realmente precisam ser colocados em algum outro diretório, você ainda pode mantê-los em /usr/local/ e fazer as ligações simbólicas apropriadas a partir do lugar necessário para sua localização em /usr/local/. Por exemplo, você pode fazer a ligação

```
ln -s /usr/local/bin/foo /usr/bin/foo
```

De qualquer modo, se você obtiver um pacote cujo copyright permita redistribuição, você deve pensar em fazer um pacote Debian a partir dele, e enviá-lo para o sistema Debian. Guias sobre como se tornar um desenvolvedor de pacotes estão incluídos no Debian Policy manual (veja 'Que outra documentação existe em/para um sistema Debian?' on page 59).

3.9 Por que aparece “Can’t find libX11.so.6” (“Impossível encontrar libX11.so.6”) quando tento rodar foo?

Essa mensagem de erro pode significar que o programa está ligado à versão `libc5` das bibliotecas do X11. Nesse caso você precisa instalar o pacote `xlib6`, da seção `oldlibs`.

Você pode obter mensagens de erro semelhantes que se referem ao arquivo `libXpm.so.4`, nesse caso, você precisa instalar a versão `libc5` da biblioteca XPM, do pacote `xpm4.7`, também na seção `oldlibs`.

3.10 Por que não consigo compilar programas que exigem a `libtermcap`?

A Debian usa o banco de dados `terminfo` e a biblioteca de rotas de interfaces de terminal `ncurses`, ao invés do banco de dados `termcap` e a biblioteca `termcap`. Os usuários que compilam programas que necessitam de algum conhecimento da interface de terminal devem substituir referências a `libtermcap` por referências a `libncurses`.

Para dar suporte a binários que já foram ligados à biblioteca `termcap` e para os quais você não possui o código fonte, a Debian oferece um pacote chamado `termcap-compat`. Ele possui os arquivos `libtermcap.so.2` e `/etc/termcap`. Instale esse pacote caso o programa falhe com a mensagem de erro “can’t load library ‘libtermcap.so.2’” (“impossível carregar biblioteca ‘libtermcap.so.2’”), ou caso ele reclame da falta do arquivo `/etc/termcap`.

3.11 Por que não consigo instalar o AccelX?

O AccelX usa a biblioteca `termcap` para a instalação. Veja ‘Por que não consigo compilar programas que exigem a `libtermcap`?’ on this page acima.

3.12 Por que minhas velhas aplicações XFree 2.1 Motif travam?

Você precisa instalar o pacote `motifnls`, que provê os arquivos de configuração do XFree-2.1 necessários para permitir aplicativos Motif compilados sobre o XFree-2.1 rodar sobre o XFree-3.1.

Sem esses arquivos, Alguns aplicativos Motif compilados em outras máquinas (como o Netscape) podem travar quando tentarem copiar ou colar de um ou para um campo texto, e podem também exibir outros problemas.

Capítulo 4

Software disponível no sistema Debian.

4.1 Que tipos de aplicativos e software de desenvolvimento estão disponíveis para o Debian GNU/Linux?

Assim como na maioria das distribuições Linux, o Debian GNU/Linux oferece:

- os principais aplicativos GNU para desenvolvimento de software, manipulação de arquivos e processamento de texto, incluindo gcc, g++, make, texinfo, Emacs, o shell (interpretador de comandos) Bash e vários utilitários Unix atualizados;
- Perl, Python, Tcl/Tk e vários programas correlatos, módulos e bibliotecas para cada um deles;
- TeX (LaTeX) e Lyx, dvips, Ghostscript;
- O sistema X Window, que oferece uma interface gráfica em rede para o Linux, e inumeráveis aplicativos X, incluindo o GNOME;
- um conjunto completo de outros aplicativos de rede, incluindo servidores para protocolos internet como o HTTP (WWW), FTP, NNTP (news), SMTP e POP (mail) e servidores de nomes; também disponíveis estão navegadores internet (web browsers) e ferramentas de desenvolvimento.

Mais de 3750 pacotes, variando de servidores de notícias e leitores a suporte de som, programas de FAX, programas de banco de dados e planilha eletrônica, programas de processamento de imagens, comunicação, rede, e utilidades de correio, servidores de rede (Web servers), e até programas de rádio-amadorismo são incluídos na distribuição. Outros 450 conjuntos de softwares estão disponíveis como pacotes Debian, mas não são formalmente parte do Debian devido as suas restrições de licença.

4.2 Quem escreveu todos esses programas?

Para cada pacote, os *autores* do respectivo programa são creditados no arquivo `/usr/doc/PACOTE/copyright`, onde PACOTE deve ser substituído pelo nome do pacote.

Mantenedores que empacotam esses programas para o sistema Debian GNU/Linux são listados no arquivo de controle Debian (veja ‘O que é um arquivo de controle Debian?’ on page 29) que vem com cada pacote.

4.3 Como posso conseguir uma lista atual dos programas que foram empacotados para o projeto Debian?

Uma lista completa está disponível a partir de qualquer um dos espelhos (<http://www.br.debian.org/distrib/ftplist>) do Debian.

O arquivo `indices/Packages-Master-i386.gz` contém uma lista, incluindo descrições curtas, de todos os pacotes que estão disponíveis para computadores com processadores ix86 (e compatíveis). Existem arquivos semelhantes, em `indices/Packages-Master-ARCH.gz`, onde ARCH deve ser substituído pela arquitetura apropriada, que provê uma lista semelhante de pacotes que estão disponíveis para tais computadores/arquiteturas.

A interface WWW para os pacotes Debian (<http://packages.debian.org/>) resume convenientemente os pacotes em cada uma das vinte “seções” do arquivo Debian.

4.4 Por que eu obtenho a mensagem “ld: cannot find -lfoo” quando compilo programas? Por que não existe nenhum arquivo libfoo.so nos pacotes de bibliotecas Debian?

A política do Debian requer que as ligações simbólicas (para `libfoo.so.x.y.z` ou semelhante) sejam colocados em separado, pacotes de desenvolvimento. Estes pacotes normalmente são nomeados `libfoo-dev` ou `libfooX-dev` (presumindo que o pacote de biblioteca seja nomeado `libfooX`, e X seja um número inteiro).

4.5 O que está faltando no Debian GNU/Linux?

Uma lista de pacotes que ainda precisam ser empacotados para o Debian existe, a lista “Work-Needing and Prospective Packages”. Para mais detalhes, veja ‘Como posso me tornar um desenvolvedor de software Debian?’ on page 65.

4.6 (Como) O Debian suporta Java?

O Kit de Desenvolvimento Java da Sun (JDK) está disponível atualmente na forma de pacote Debian (`jdk_VVV-RRR.deb`). O JDK permite a você rodar programas e applets Java, além de escrever os seus próprios programas e applets. Se seu kernel está configurado corretamente (veja abaixo), o JDK permitirá a você *rodar* programas Java da mesma maneira que você roda outros executáveis. O pacote JDK também inclui alguns programas de demonstração.

O kernel do Debian está configurado com suporte a Java como módulo (isto é, `CONFIG_BINFMT_JAVA=m`). Os usuários que desejarem construir seus próprios kernels (veja 'Que ferramentas o Debian oferece para a construção de kernels personalizados?' on page 49) podem omitir este suporte se desejarem. Uma vez que haja suporte a módulos no kernel, tenha certeza de que o módulo foi carregado. Você pode fazer isso durante a inicialização colocando a linha `binfmt_java` no arquivo `/etc/modules`.

Alternativamente, você pode instalar o módulo a partir da linha de comando executando o comando `insmod NOMEDIR/binfmt_java.o`, onde `NOMEDIR` é o nome do diretório em que se encontram os módulos construídos para a versão do kernel que está sendo executado. Em um sistema com a versão 2.2.17 do kernel, `NOMEDIR` provavelmente será `/lib/modules/2.2.17/fs/`. Você pode verificar se um módulo está carregado usando o comando `lsmod`.

Rodar um applet Java exige um navegador com capacidade para reconhecê-lo e executá-lo. O navegador da Netscape, que pode ser instalado como um pacote Debian, roda applets Java. (O código fonte do navegador da Netscape não está disponível publicamente. O pacote Debian do navegador fornece um wrapper (envelope) que auxilia a instalação e o gerenciamento do Netscape em um sistema Debian. Esse é um bom exemplo da integração de programas comerciais com o sistema Debian).

Uma nota final com boas e más notícias: A política de licenciamento da Sun sobre o JDK está ficando cada vez mais restritiva, portanto em pouco tempo esse pacote pode não estar disponível como parte do Debian. É possível que ele possa estar disponível da mesma maneira que o Netscape. Uma notícia melhor é que estão sendo desenvolvidos vários programas que poderão oferecer alternativas atraentes e que estarão publicamente disponíveis.

4.7 Como posso verificar se estou usando um sistema Debian, e que versão é esta?

Para ter certeza de que seu sistema foi instalado a partir dos discos base do Debian, verifique a existência do arquivo `/etc/debian_version`, que contém apenas uma linha, dando o número da versão, como definido pelo pacote `base-files`.

A existência do programa `dpkg` mostra que você pode instalar pacotes Debian em seu sistema, mas como o programa tem sido utilizado por vários outros sistemas operacionais e arquiteturas, esse não é mais um método confiável para se determinar se o sistema é Debian GNU/Linux.

Os usuários devem estar cientes, de que o sistema Debian consiste de muitas partes, cada uma das quais podem ser atualizadas de forma (quase) independente. Cada “versão” do Debian possui conteúdo bem definido e imutável. As atualizações ficam disponíveis separadamente. Para uma descrição de uma linha do status de instalação do pacote `foo`, use o comando `dpkg --get-selections foo`. (Sem argumentos, esse comando mostra as versões de todos os pacotes instalados). Para ver as versões dos pacotes instalados, execute:

```
dpkg -l
```

Para uma descrição mais completa, use:

```
dpkg --get-architecture foo
```

4.8 Como o Debian suporta idiomas diferentes do Inglês?

- O Debian GNU/Linux é distribuído com mapas de teclado para quase duas dúzias de teclados e com utilitários (no pacote `kbd`) para instalar, ver e modificar esses mapas. O processo de instalação pede ao usuário para especificar o teclado que ele usará.
- A grande maioria dos programas que empacotamos, suportam a entrada de caracteres não US-ASCII usados em outros idiomas latinos (e.g. ISO-8859-1 ou ISO-8859-2), e um grande número de pacotes suportam idiomas “multi-byte” como japonês e chinês.
- Atualmente, o suporte para páginas de manual nos idiomas alemão, espanhol, finlandês, francês, húngaro, italiano, japonês, coreano e polonês é oferecido através dos pacotes ‘`manpages-LANG`’ (onde `LANG` é o código ISO de duas letras do país). Para acessar uma página de manual NLS, o usuário precisa ajustar a variável do ‘shell’ `LC_MESSAGES` para o valor apropriado.

Por exemplo, no caso de páginas de manual em italiano, `LC_MESSAGES` deve ser ajustado para ‘`italian`’. O programa `man` irá então procurar por páginas de manual em italiano sob `/usr/share/man/it/`.

4.9 E as limitações de exportação dos EUA?

As leis dos EUA restringem a exportação de artigos de defesa, o que inclui alguns tipos de programas de criptografia. PGP e ssh, entre outros, entram nessa categoria.

Para prevenir que alguém corra riscos legais desnecessários, alguns pacotes do Debian GNU/Linux estão disponíveis apenas a partir de um site fora dos EUA <ftp://non-US.debian.org/debian-non-US/>. Existem numerosos sites espelhos, todos os quais, fora dos EUA, veja <ftp://non-US.debian.org/debian-non-US/README.non-US> para uma lista completa.

4.10 Onde está o Pine?

Devido à sua licença restritiva, ele está na área non-free. Além disso, como sua licença não permite nem mesmo que binários modificados sejam distribuídos, você deve compilá-lo por conta própria a partir do código-fonte e dos patches Debian. Para maior conveniência dos usuários que usam o `dselect`, o código-fonte para a versão Debian do pine está disponível nos pacotes `pine396-src` e `pine396-diffs`, ou `pine4-src` e `pine4-diffs` para a versão 4.*.

Note que existem muitos substitutos tanto para o pine quanto para o pico, como o `mutt` e `nano`, ambos localizados na seção `main`.

Capítulo 5

Os repositórios FTP do Debian.

5.1 O que são todos aqueles diretórios nos repositórios FTP do Debian?

Os programas que foram empacotados para o Debian GNU/Linux estão disponíveis em várias árvores de diretórios em cada espelho do Debian.

O diretório `dists` contém as “distribuições”, e é o modo canônico de acessar as versões Debian atualmente disponíveis (e versões antigas).

5.2 Quantas distribuições Debian existem no diretório `dists`?

Normalmente existem três distribuições, a distribuição “stable” (estável), a distribuição “testing” (teste) e a distribuição “unstable” (instável). Às vezes, também há a distribuição “frozen” (paralizada).

5.3 O que são todos esses nomes, como “slink”, “potato”, etc.?

São apenas “codinomes”. Quando uma distribuição Debian está em estágio de desenvolvimento, ela não possui um número de versão, e sim um codinome. O objetivo desses codinomes é facilitar o espelhamento das distribuições Debian (se um diretório de verdade como `unstable` de repente mudasse seu nome para `stable`, muita coisa teria que ser baixada novamente).

Atualmente, `stable` é uma ligação simbólica para `potato` (isto é, Debian GNU/Linux 2.2), `testing` é uma ligação simbólica para `woody` e `unstable` é uma ligação simbólica para `sid`, o que significa que `potato` é a atual distribuição estável e `woody` é a atual distribuição teste. `sid` é sempre a distribuição instável (veja ‘E sobre a “sid”?’ on the following page).

5.4 E sobre a “frozen” (paralizada)?

Quando a distribuição teste está madura o suficiente, ela é paralizada, significando que nenhum código novo é admitido, somente ajustes e consertos no código, se necessários. Além disso, uma nova árvore teste é criada no diretório `dist`s, com um novo codinome. A distribuição paralisada passa por alguns meses de testes, com atualizações intermitentes e ciclos de testes. Mantemos um registro de bugs na distribuição “frozen” que pode impedir que um pacote seja lançado. Uma vez que o número de bugs desça até níveis aceitáveis, a distribuição “frozen” torna-se estável e é lançada, e a distribuição estável anterior torna-se obsoleta (e é movida para o arquivo).

5.5 Que outros codinomes foram usados no passado?

Outros codinomes que já foram usados são: `buzz` para a versão 1.1, `rex` para a versão 1.2, `bo` para as versões 1.3.x, `hamm` para a versão 2.0, e `slink` para a versão 2.1.

5.6 De onde vieram esses codinomes?

Até agora eles vieram de personagens do filme “Toy Story” da Pixar.

- `buzz` (Buzz Lightyear) era o astronauta;
- `rex` era o tiranossauro;
- `bo` (Bo Peep) era a menina que cuidava da ovelha;
- `hamm` era o porquinho-cofre;
- `slink` (Slinky Dog) era o cachorro de brinquedo;
- `potato` era, claro, o Mr. Potato (Sr. Batata);
- `woody` era o vaqueiro.

5.7 E sobre a “sid”?

A `sid` ou `unstable` (instável) é o lugar onde a maioria dos pacotes é inicialmente inserida à distribuição. Ela nunca será lançada diretamente, porque pacotes que estão para serem lançados, terão primeiro que ser incluídos na `testing`, de maneira a ser lançada na `stable`. A `sid` contém pacotes para as arquiteturas lançadas e as que ainda não foram oficialmente lançadas.

O nome “sid” também veio do desenho animado “Toy Story”: Sid era o garoto na casa ao lado, que destruía brinquedos :-)

5.7.1 Notas históricas sobre a “sid”.

Quando a sid ainda não existia, a organização do site de FTP tinha uma falha grave: havia uma presunção de quando uma arquitetura fosse criada na unstable, ela seria lançada quando aquela distribuição se tornasse a nova stable. Para muitas arquiteturas esse não era o caso, de forma que aqueles diretórios tinham que ser movidos na hora do lançamento. Isso não era prático, pois essa tarefa diminuía a largura de banda.

Os administradores trabalharam sobre esse problema por vários anos, colocando binários de versões ainda não lançadas em diretórios especiais chamados “sid”. Para aquelas arquiteturas que não haviam sido lançadas ainda, a primeira vez que elas foram lançadas, isso se deu através de uma ligação da stable atual com a sid, e a partir desse momento, eles eram criados dentro da árvore unstable normalmente. Esse esquema era um tanto confuso para os usuários.

Com o advento dos conjuntos de pacote (veja ‘O que existe no diretório pool?’ on page 25), pacotes binários começaram a ser armazenados em um local canônico nesse conjunto, qualquer que fosse a distribuição, portanto, o lançamento de uma distribuição não causa mais o consumo de grande largura de banda nos espelhos (há, na verdade, um consumo gradual da largura de banda através do processo de desenvolvimento).

5.8 O que o diretório “stable” contém?

- stable/main/: este diretório contém os pacotes que constituem formalmente a versão mais recente do sistema Debian GNU/Linux.

Todos esses pacotes estão em conformidade com as Orientações sobre Software Livre do Debian (http://www.br.debian.org/social_contract#guidelines), e são todos livremente utilizáveis e distribuíveis.

- stable/non-free/: este diretório contém pacotes cuja distribuição é restrita de tal modo que requer que os distribuidores considerem cuidadosamente as exigências de copyright especificadas.

Por exemplo, alguns pacotes têm licenças que proíbem distribuição comercial. Outros podem ser redistribuídos, mas são na verdade shareware, e não freeware. As licenças de cada um desses pacotes devem ser estudadas e possivelmente negociada antes que os pacotes sejam incluídos em qualquer redistribuição (por exemplo, em um CD-ROM).

- stable/contrib/: este diretório contém pacotes que são DFSG-free e podem ser *livremente distribuídos*, mas que por alguma razão dependem de um pacote que *não* é livremente distribuível e, portanto, disponível somente na seção non-free.

5.9 O que o diretório “testing” contém?

Pacotes são instalados no diretório “testing” depois que eles sofrem algum um certo grau de teste na unstable. Eles devem estar em sincronismo em todas as arquiteturas para que eles

foram construídos e não podem possuir dependências que os tornem instáveis; eles também devem ter poucos bugs críticos em relação as versões atualmente na “testing”. Desse modo, esperamos que a “testing” esteja sempre perto de ser uma candidata a lançamento.

5.10 O que o diretório “unstable” contém?

O diretório “unstable” contém uma imagem do atual sistema em desenvolvimento. Usuários são bem vindos a usar e testar esses pacotes, mas são avisados sobre seu estado incompleto. A vantagem de se usar a distribuição “unstable” é que você está sempre atualizado com os mais recentes lançamentos da indústria GNU/Linux, mas se ele falhar: você terá que manter ambas as partes :-)

Há também os subdiretórios main, contrib e non-free na “unstable”, separados pelos mesmos critérios da “stable”.

5.11 O que são todos aqueles diretórios dentro de `dists/stable/main`?

Dentro de cada árvore de diretório principal (`dists/stable/main`, `dists/stable/contrib`, `dists/stable/non-free`, and `dists/unstable/main`, etc.), os pacotes binários residem em subdiretórios cujos nomes indicam a arquitetura do processador para a qual eles foram compilados:

- `binary-all/`, para pacotes que são independentes da arquitetura. Isso inclui, por exemplo, scripts Perl, ou documentação simples.
- `binary-i386/`, para pacotes que são executados em máquinas PC 80x86.
- `binary-m68k/`, para pacotes que são executados em máquinas baseadas no processador Motorola 680x0. Atualmente, isto é feita principalmente para computadores Atari e Amiga, e também para algumas placas baseadas no padrão industrial VME.
- `binary-sparc/`, para pacotes que são executados em Sun SPARCStations.
- `binary-alpha/`, para pacotes que são executados em máquinas DEC Alpha.
- `binary-powerpc/`, para pacotes que são executados em máquinas PowerPC.
- `binary-arm/`, para pacotes que são executados em máquinas ARM.

Veja ‘Em quais arquiteturas de hardware o Debian GNU/Linux roda?’ on page 9 para mais informações.

5.12 Onde está o código-fonte?

O código fonte é incluído para tudo no sistema Debian. Além disso, os termos de licenças de muitos programas do sistema, *requerem* que o código-fonte seja distribuído com os programas, ou que uma oferta de obtenção do código-fonte acompanhe o programa.

Normalmente o código-fonte é distribuído através do diretório “source”, que é paralelo a todos os diretórios binários específicos a cada arquitetura, ou atualmente, no diretório `pool` (veja ‘O que existe no diretório `pool`?’ on the current page). Para obter o código-fonte sem ter que estar familiarizado com a estrutura do repositório de FTP, tente um comando como `apt-get source MeuNomeDePacote`.

Códigos-fonte podem ou não estar disponíveis para pacotes pertencentes ao diretório “contrib” e “non-free”, que não são partes formais do sistema Debian.

5.13 O que existe no diretório `pool`?

Historicamente, pacotes foram mantidos nos subdiretórios correspondentes do diretório `dists`. Isso demonstrou ser a causa de vários problemas, como o grande consumo de largura de banda nos espelhos quando grandes mudanças eram feitas.

Pacotes agora são mantidos em grandes “repositórios”, estruturados de acordo com o nome do pacote fonte. Para fazer isso administrável, o repositório é subdividido por seções (“main”, “contrib” e “non-free”) e pela primeira letra do nome do pacote fonte. Esses diretórios contêm vários arquivos: os arquivos binários para cada arquitetura, e os pacotes fonte de onde os pacotes binários são gerados.

Você pode descobrir onde cada pacote é colocado executando um comando como `apt-cache showsrc NomeDoMeuPacote` e olhando na linha “Directory:”. Por exemplo, os pacotes `apache` são armazenados em `pool/main/a/apache/`. Como existem tantos pacotes `lib*`, estes são tratados especialmente: por exemplo, pacotes `libpaper` são armazenados em `pool/main/libp/libpaper/`.

Os diretórios `dists` ainda são usados para os arquivos índices usados por programas como `apt`. Também, na hora de escrever, velhas distribuições não foram convertidas para o uso de repositórios, deste modo, você verá caminhos contendo distribuições como “potato” ou “woody” no campo de cabeçalho “Filename”.

Normalmente, você não terá que se preocupar com isso, já que o `apt` e provavelmente o `dpkg-ftp` (veja ‘Como posso manter meu sistema Debian atualizado?’ on page 44) resolverá isso automaticamente. Se você deseja mais informações, veja o FAQ dos repositórios de pacotes Debian (<http://people.debian.org/~joeyh/poolfaq>).

5.14 Ok, e sobre os outros diretórios, que não estão dentro de `dist` ou `pool`?

Os seguintes diretórios adicionais existem:

/tools/: Utilitários DOS para a criação de discos de boot, particionamento de seu disco rígido, compressão/descompressão de arquivos, e inicializar o Linux.

/doc/: Documentação Debian, FAQ do Debian, instruções de como enviar relatórios de bugs.

/indices/: Vários índices, mantenedores, arquivos Packages-master, arquivos ignorados.

/project/: Muito material somente para desenvolvedores, como:

project/experimental/: Este diretório contém pacotes e ferramentas que ainda estão sendo desenvolvidas, e ainda estão do estágio alpha de testes. Usuários não devem usar pacotes daqui, porque eles podem ser perigosos e prejudiciais até mesmo para pessoas experientes.

project/orphaned/: Pacotes que foram abandonados por seus desenvolvedores, e retirados da distribuição.

Capítulo 6

Fundamentos do Sistema de Gerenciamento de Pacotes Debian

6.1 O que é um pacote Debian?

De modo geral, pacotes contêm todos os arquivos necessários para implementar um conjunto de recursos ou comandos relacionados. Existem dois tipos de pacotes Debian:

- *Pacotes binários*, que contêm arquivos executáveis, de configuração, páginas de manual/info, informações de copyright e outras documentações. Esses pacotes são distribuídos em um formato de arquivo específico do Debian (veja ‘Qual é o formato de um pacote binário Debian?’ on the next page); eles são normalmente caracterizados pela extensão `.deb`. Pacotes binários podem ser descompactados utilizando o utilitário `dpkg`; detalhes são dados em sua página de manual.
- *Pacotes fonte*, que consistem em um arquivo `.dsc` que descreve o pacote fonte (incluindo os nomes dos seguintes arquivos), um arquivo `.orig.tar.gz` que contém o código fonte original sem modificações, em formato tar comprimido com gzip, e normalmente um arquivo `.diff.gz` que contém as mudanças específicas para o Debian em relação ao código fonte original. O utilitário `dpkg-source` empacota e desempacota pacotes fonte Debian; os detalhes são fornecidos em sua página de manual.

A instalação de software pelo sistema de pacotes usa “dependências” que são cuidadosamente designadas pelos mantenedores dos pacotes. Essas dependências são documentadas no arquivo de controle associado a cada pacote. Por exemplo, o pacote que contém o compilador GNU C (`gcc`) “depende” do pacote `binutils`, que inclui o ligador e o montador assembler. Se um usuário tentar instalar o `gcc` sem antes instalar o `binutils`, o sistema de gerenciamento de pacotes (`dpkg`) mostrará uma mensagem de erro indicando que o usuário também precisa do `binutils`, e irá deixar de instalar o `gcc`. (Apesar disso, esse recurso pode ser desativado por usuários insistentes, veja `dpkg(8)`). Veja mais em ‘O que significa dizer que um arquivo Depends/Recommends/Suggests/Conflicts/Replaces/Provides (Depende/Recomenda/Sugere/Conflita/Substitui/Oferece) outro pacote?’ on page [32](#) abaixo.

As ferramentas de empacotamento do Debian podem ser usadas para:

- manipular e administrar pacotes ou partes de pacotes;
- ajudar o usuário na divisão de pacotes que precisam ser transmitidos através de mídia limitada em tamanho como disquetes;
- ajudar desenvolvedores na construção de pacotes; e
- ajudar usuários na instalação de pacotes que estejam em um servidor FTP remoto.

6.2 Qual é o formato de um pacote binário Debian?

Um “pacote” Debian, ou um repositório de arquivos Debian contém os arquivos executáveis, bibliotecas e documentação associada a um conjunto particular de programas. Normalmente, o arquivo Debian possui um nome que termina em `.deb`.

O formato interno desses pacotes binários Debian é descrito na página de manual `deb(5)`. Esse formato interno está sujeito a mudanças (entre as principais versões do Debian GNU/Linux), portanto, use sempre `dpkg-deb(8)` para manipular arquivos `.deb`.

6.3 Por que os nomes de pacotes Debian são tão longos?

Os nomes de pacotes de binários Debian seguem a seguinte convenção:
`<foo>_<Número de Versão>-<Número da Revisão Debian>.deb`

Perceba que `foo` é supostamente o nome do pacote. Para verificação, pode-se descobrir o nome do pacote associado a um arquivo Debian particular (`.deb`) através de um dos seguintes meios:

- inspecionar o arquivo “Packages” no diretório onde ele foi armazenado em um site FTP do Debian. Esse arquivo contém uma entrada descrevendo cada pacote; o primeiro campo em cada entrada é formalmente o nome do pacote.
- usar o comando `dpkg --info foo_VVV-RRR.deb` (onde VVV e RRR são respectivamente os números da versão e da revisão do pacote em questão). Isso mostra entre outras coisas, o nome do pacote correspondente ao arquivo sendo desempacotado.

O componente VVV é o número de versão especificado pelo desenvolvedor original do programa. Não existem padrões aqui, então o número de versão pode ter formatos tão diferentes quanto “19990513” e “1.3.8pre1”.

O componente RRR é o número da revisão Debian, e é especificado pelo desenvolvedor Debian (ou um usuário individual, se ele próprio decidir construir o pacote). Esse número corresponde ao nível de revisão do pacote Debian, portanto, um novo nível de revisão geralmente significa mudanças no Makefile do Debian (`debian/rules`), no arquivo de controle Debian (`debian/control`), nos scripts de instalação e remoção (`debian/p*`), ou nos arquivos de configuração usados com o pacote.

6.4 O que é um arquivo de controle Debian?

Detalhes sobre o conteúdo de um arquivo de controle Debian podem ser encontrados no “Debian Packaging Manual”, no capítulo 4, veja ‘Que outra documentação existe em/para um sistema Debian?’ on page 59.

Resumidamente, um arquivo de controle de exemplo é mostrado abaixo para o pacote Debian “hello”:

```
Package: hello
Priority: optional
Section: devel
Installed-Size: 45
Maintainer: Adam Heath <doogie@debian.org>
Architecture: i386
Version: 1.3-16
Depends: libc6 (>= 2.1)
Description: The classic greeting, and a good example
 The GNU hello program produces a familiar, friendly greeting. It
 allows nonprogrammers to use a classic computer science tool which
 would otherwise be unavailable to them.
.
 Seriously, though: this is an example of how to do a Debian package.
 It is the Debian version of the GNU Project's 'hello world' program
 (which is itself an example for the GNU Project).
```

O campo “Package” indica o nome do pacote. Esse é o nome pelo qual o pacote pode ser manipulado pelas ferramentas de pacote, e é normalmente semelhante para, mas não necessariamente igual a primeira parte do nome do arquivo do pacote Debian.

O campo “Version” indica tanto o número de versão do desenvolvedor original quanto o nível de revisão (na última parte) do pacote Debian deste programa, como explicado em ‘Por que os nomes de pacotes Debian são tão longos?’ on the facing page.

O campo “Architecture” especifica o processador para o qual este binário em particular foi compilado.

O campo “Depends” apresenta uma lista de pacotes que devem estar instalados para que este pacote seja instalado com sucesso.

O campo “Installed-Size” indica quanto espaço em disco o pacote instalado consumirá. Este campo é usado por front-ends de instalação, a fim de mostrar se há espaço suficiente disponível para instalar o programa.

A linha “Section” indica a seção onde esse pacote Debian é armazenado nos sites FTP do Debian. Esse é um nome de subdiretório (dentro de um dos diretórios principais, veja ‘O que são todos aqueles diretórios nos repositórios FTP do Debian?’ on page 21) onde o pacote é armazenado.

O campo “Priority” indica quão importante é este pacote para instalação, de forma que softwares semi-inteligentes como o `dselect` ou o `apt` possa ordenar o pacote dentro de uma categoria de, por exemplo, pacotes opcionais instalados. Veja ‘O que é um pacote Required/Important/Standard/Optional/Extra?’ on the facing page.

O campo “Maintainer” indica o endereço eletrônico da pessoa responsável pela manutenção deste pacote.

O campo “Description” mostra um breve resumo das características do pacote.

Para mais informações sobre todos os campos possíveis que um pacote pode ter, por favor, veja o “Debian Packaging Manual”, seção 4, “Control files and their fields”.

6.5 O que é um `conffile` Debian?

`Conffiles` é uma lista de arquivos de configuração (normalmente colocados em `/etc`) que o sistema de gerenciamento de pacotes não sobrescreverá quando o pacote for atualizado. Isso assegura que valores locais para os conteúdos desses arquivos serão preservados, e isso é uma característica crítica que permite a atualização de pacotes enquanto o sistema está funcionando.

Para determinar exatamente que arquivos são preservados durante uma atualização, rode:

```
dpkg --status package
```

E olhe sob “`Conffiles:`”.

6.6 O que são os scripts Debian `preinst`, `postinst`, `prerm` e `postrm`?

Esses arquivos são scripts executáveis que são executados automaticamente antes ou depois de um pacote ser instalado. Juntamente com um arquivo chamado `control`, todos esses arquivos são parte da seção “`control`” de um arquivo Debian.

Os arquivos individuais são:

`preinst` Este script é executado antes que o pacote seja desempacotado de seu arquivo Debian (“`.deb`”). Muitos scripts ‘`preinst`’ param serviços de pacotes que estejam sendo atualizados até que sua instalação ou atualização esteja completa (após a execução correta do script ‘`postinst`’).

`postinst` Este script normalmente completa qualquer configuração exigida pelo pacote `foo` depois que `foo` tenha sido desempacotado de seu arquivo Debian (“`.deb`”). Frequentemente, scripts ‘`postinst`’ pedem informações ao usuário, e/ou avisam-no que se ele aceitar valores padrões, deve lembrar-se de reconfigurar o pacote conforme a necessidade. Muitos scripts ‘`postinst`’ executam quaisquer comandos necessários para iniciar ou reiniciar um serviço uma vez que o novo pacote tenha sido instalado ou atualizado.

pre_{rm} Este script geralmente para quaisquer daemons que estejam associados a um pacote. Ele é executado antes da remoção de arquivos associados ao pacote.

post_{rm} Este script normalmente modifica ligações ou outros arquivos associados a `foo`, e/ou remove arquivos criados pelo pacote. (Veja também ‘O que é um pacote virtual?’ on the next page).

Atualmente todos os arquivos de controle podem ser encontrados em `/var/lib/dpkg/info`. Os arquivos relevantes ao pacote `foo` começam com o nome “foo”, e possuem extensões “preinst”, “postinst”, etc., conforme apropriado. O arquivo `foo.list` naquele diretório lista todos os arquivos que foram instalados com o pacote `foo`. (Perceba que a localização desses arquivos é um aspecto interno do `dpkg`; você não deve confiar nisso).

6.7 O que é um pacote Required/Important/Standard/Optional/Extra?

A cada pacote Debian é atribuída uma *prioridade* designada pelos mantenedores da distribuição, para auxiliar o sistema de gerenciamento de pacotes. As prioridades são:

- **Required:** pacotes que são necessários para o funcionamento correto do sistema.
Isto inclui todas as ferramentas necessárias para o conserto de defeitos do sistema. Você não deve remover esses pacotes ou seu sistema poderá ficar inoperante e você provavelmente não conseguirá nem mesmo usar o `dpkg` para colocar tudo em seu devido lugar. Sistemas que possuam apenas os pacotes obrigatórios provavelmente não têm utilidade, mas têm funcionalidade suficiente para permitir que o administrador dê boot e instale mais software.
- **Important:** pacotes que devem estar presentes em qualquer sistema estilo Unix.
Outros pacotes sem os quais o sistema não será útil ou não funcionará corretamente estarão aqui. Isto *NÃO* inclui Emacs, X11, TeX ou quaisquer outros grandes aplicativos. Estes pacotes constituem apenas a infra-estrutura básica.
- **Standard:** pacotes que são comuns em qualquer sistema Linux, incluindo um sistema em modo caractere razoavelmente pequeno, mas não muito limitado.
Isto é o que será instalado normalmente se o usuário não selecionar mais nada. Não inclui muitos aplicativos grandes, mas inclui Emacs (este é mais uma peça de infra-estrutura do que um aplicativo) e uma parte razoável do TeX e LaTeX (se isso for possível sem o X).
- **Optional:** pacotes que incluem todos aqueles que você normalmente gostaria de instalar mesmo se não soubesse o que eram, ou não tivesse necessidades especiais.
Isto inclui o X11, uma distribuição completa do TeX e muitos aplicativos.
- **Extra:** pacotes que entram em conflito com outros de maior prioridade, ou provavelmente são úteis apenas se você sabe o que são ou se possuem necessidades especiais que os tornam inadequados para serem “optional”.

6.8 O que é um pacote virtual?

Um pacote virtual é um nome genérico que se aplica a qualquer elemento de um grupo de pacotes, onde todos oferecem funcionalidade básica similar. Por exemplo, ambos os programas `tin` e `trn` são leitores de news, e devem então satisfazer qualquer dependência de um programa que exija um leitor de news em um sistema para funcionar ou ser útil. Diz-se que ambos oferecem o “pacote virtual” chamado `news-reader` (leitor de news).

Analogamente, `smail` e `sendmail` oferecem a funcionalidade de um “mail transport agent” (agente de transporte de correio). Diz-se então que ambos oferecem o pacote virtual “mail transport agent”. Se qualquer um deles está instalado, programas que dependam de um `mail-transport-agent` serão satisfeitos pela existência deste pacote virtual.

O Debian fornece um mecanismo que, se mais de um pacote que oferece o mesmo pacote virtual estiver instalado em um sistema, então os administradores podem selecionar um deles como o pacote preferencial. O comando pertinente é `update-alternatives`, que é descrito adiante na seção ‘Alguns usuários gostam do `mawk`, outros do `gawk`; alguns gostam do `vim`, outros do `elvis`; alguns gostam do `trn`, outros do `tin`; como o Debian lida com diversidade?’ on page 57.

6.9 O que significa dizer que um arquivo Depends/Recommends/Suggests/Conflicts/Replaces/Provides (Depende/Recomenda/Sugere/Conflita/Substitui/Oferece) outro pacote?

O sistema de pacotes Debian possui uma gama de “dependências” entre pacotes que são planejadas para indicar (em um único campo) o nível no qual o Programa A pode operar independentemente da existência do Programa B em um dado sistema:

- O Pacote A *depende* (“Depends”) do Pacote B, se B deve necessariamente estar instalado para que A possa ser executado. Em alguns casos, A depende não apenas de B, mas de uma versão de B. Neste caso, a dependência na versão é normalmente um limite mínimo, no sentido de que A depende de qualquer versão de B mais recente que uma versão específica.
- O Pacote A *recomenda* (“Recommends”) o Pacote B, se o mantenedor do pacote julga que a maioria dos usuários não usariam A sem ter também a funcionalidade oferecida por B.
- O Pacote A *sugere* (“Suggests”) o Pacote B se B contém arquivos que estão relacionados com (e geralmente melhoram) a funcionalidade de A.
- O Pacote A *conflita* (“Conflicts”) com o Pacote B quando A não funciona se B está instalado no sistema. Normalmente, conflitos são casos onde A contém arquivos que são melhorias em relação aos de B. “Conflitos” geralmente são combinados com “substituições” (“replaces”).

- O Pacote A *substitui* (“Replaces”) o Pacote B quando arquivos instalados por B são removidos e (em alguns casos) sobrescritos por arquivos de A.
- O Pacote A *oferece* (“Provides”) o Pacote B quando todos os arquivos e a funcionalidade de B estão incorporados em A. Este mecanismo oferece um modo pelo qual os usuários com espaço em disco limitado tenham apenas a parte do pacote A que eles realmente precisam.

Informações mais detalhadas sobre o uso desses termos podem ser encontradas no Packaging Manual e no Policy Manual.

6.10 O que significa Pré-Dependência?

“Pré-Dependência” é uma dependência especial. No caso da maioria dos pacotes, o `dpkg` descompactará seu arquivo (ou seja, seu arquivo `.deb`) independentemente se os arquivos dos quais ele depende existem ou não no sistema. De forma simples, descompactar significa que o `dpkg` irá extrair o conteúdo do arquivo que deveria ser instalado em seu sistema de arquivos, e o colocará no seu lugar correto. Se aquele pacote *depende* da existência de outros pacotes do seu sistema, o `dpkg` se negará a completar a instalação (executando sua ação “configure”) até que os outros pacotes sejam instalados.

Contudo, para alguns pacotes, o `dpkg` se negará a descompactá-los até que certas dependências sejam resolvidas. Diz-se que tais pacotes “pré-dependem” (“Pre-Depends”) da presença de outros pacotes. O projeto Debian ofereceu este mecanismo para dar suporte a atualização segura de sistemas de formato `a.out` para o formato `ELF`, onde a *ordem* em que os pacotes eram descompactados era crítica. Existem outras situações de atualização onde este método é útil, por exemplo, os pacotes de prioridade “required”, que dependem da `libC`.

Como antes, mais informações detalhadas sobre isto podem ser encontradas no Packaging Manual.

6.11 O que significa unknown/install/remove/purge/hold (desconhecido/instalar/remover/remover por completo/conservar) no estado do pacote?

Estes “flags” indicam o que o usuário gostaria de fazer com o pacote (como indicado pelas ações tomadas na seção “Select” do `dselect`, ou pela chamada direta ao `dpkg`).

Seus significados são:

- unknown (desconhecido) - o usuário nunca indicou se ele quer o pacote;
- install (instalar) - o usuário deseja o pacote instalado ou atualizado;

- `remove` (remover) - o usuário deseja o pacote removido, mas não deseja excluir qualquer arquivo de configuração existente;
- `purge` (remover por completo) - o usuário deseja que o pacote seja removido por completo, incluindo seus arquivos de configuração;
- `hold` (conservar) - o usuário não deseja que esse pacote seja processado, ou seja, ele deseja manter a atual versão que está instalada em seu sistema, qualquer que ela seja.

6.12 Como coloco um pacote em “hold”?

Existem duas maneiras de colocar um pacote em “hold”, com `dpkg`, ou com `dselect`.

Com o `dpkg`, você deverá apenas exportar a lista de seleções de pacotes, com:

```
dpkg --get-selections > selections.txt
```

E então, editar o arquivo resultante `selections.txt`, altere a linha que contém o pacote que você deseja manter, por exemplo, `libc6`,disso:

```
libc6                                install
```

para isto:

```
libc6                                hold
```

Salve o arquivo, e o recarregue na base de dados do `dpkg`, com:

```
dpkg --set-selections < selections.txt
```

Com o `dselect`, você deverá apenas entrar na tela de seleção de pacotes, achar o pacote que você deseja manter em seu devido estado, e pressionar a chave ‘=’ (ou ‘H’). A mudanças aconteceram imediatamente após você sair da tela de seleção de pacotes.

6.13 Como instalo um pacote de fontes?

Pacotes Debian de fontes não são “instalados” de fato, eles são apenas descompactados no diretório em que você quiser construir os pacotes de binários que eles produzem. Pacotes de código fonte são distribuídos em um diretório chamado `source`, e você pode baixá-los manualmente, ou usar

```
apt-get source foo
```

para pegá-los (veja a página de manual `apt-get(8)` em como configurar o APT para fazer isso).

6.14 Como construo pacotes binários a partir de um pacote de códigos fonte?

Você precisará dos arquivos `foo_*.dsc`, `foo_*.tar.gz` e `foo_*.diff.gz` para compilar o código fonte (atenção: não há `.diff.gz` para um pacote Debian nativo).

Uma vez que você os tenha, se você possuir o pacote `dpkg-dev` instalado, o comando:

```
dpkg-source -x foo_version-revision.dsc
```

extrairá o pacote no diretório chamado `foo-versão`.

Se você deseja apenas compilar o pacote, deve ir para o diretório `foo-versão` e executar o comando

```
debian/rules build
```

para construir o programa, depois

```
debian/rules binary
```

como root, para construir o pacote, e em seguida

```
dpkg -i ../foo-versão-revisão_arq.deb
```

para instalar o pacote recentemente construído.

6.15 Como eu crio pacotes Debian?

Para descrições mais detalhadas sobre isso, leia o *New Maintainers' Guide*, disponível no pacote `maint-guide`, ou em <ftp://ftp.debian.org/debian/doc/package-developer/maint-guide.html.tar.gz>.

Capítulo 7

As Ferramentas de Gerenciamento de Pacotes Debian.

7.1 Que programas o Debian oferece para administrar seus pacotes?

7.1.1 dpkg

Este é o principal programa de gerenciamento de pacotes. O `dpkg` pode ser chamado com muitas opções. Alguns usos comuns são:

- Descobrir todas as opções: `dpkg --help`.
- Imprimir o arquivo de controle (e outras informações) de um pacote especificado: `dpkg --info foo_VVV-RRR.deb`
- Instalar um pacote (incluindo desempacotamento e configuração) no sistema de arquivos do disco rígido: `dpkg --install foo_VVV-RRR.deb`.
- Desempacotar (mas não configurar) um pacote Debian no sistema de arquivos do disco rígido: `dpkg --unpack foo_VVV-RRR.deb`. Perceba que esta operação *não* necessariamente deixa o pacote em um estado utilizável; alguns arquivos necessitam de uma personalização adicional para rodar corretamente. Este comando remove qualquer versão do programa já instalada, e executa o script `preinst` (veja 'O que são os scripts Debian `preinst`, `postinst`, `prerm` e `postrm`?' on page 30) associado ao pacote.
- Configurar um pacote que já foi desempacotado: `dpkg --configure foo`. Entre outras coisas, esta ação executa o script `postinst` (veja 'O que são os scripts Debian `preinst`, `postinst`, `prerm` e `postrm`?' on page 30) associado ao pacote. Ele também atualiza os arquivos listados no `conffiles` desse pacote. Observe que a operação 'configure' recebe como argumento um nome de pacote (por exemplo, `foo`), e *não* um nome-de-arquivo (por exemplo, `foo_VVV-RRR.deb`).

- Extrair um único arquivo chamado “blurf” (ou um grupo de arquivos chamados “blurf*” de um pacote Debian: `dpkg --fsys-tarfile foo_VVV-RRR.deb | tar -xf - blurf*`
- Remover um pacote (mas não seus arquivos de configuração): `dpkg --remove foo.`
- Remover um pacote (incluindo seus arquivos de configuração): `dpkg --purge foo.`
- Listar o estado de instalação dos pacotes contendo a string (ou expressão regular) “foo*”: `dpkg --list 'foo*'`.

7.1.2 dselect

Este programa é uma interface baseada em menus para o sistema de gerenciamento de pacotes Debian. É particularmente útil para instalações iniciais e atualizações em grande escala

O `dselect` pode:

- guiar o usuário conforme ele escolha entre os pacotes a instalar ou remover, assegurando que não sejam instalados pacotes que entrem em conflito com outros, e que todos os pacotes necessários para que cada pacote funcione adequadamente sejam instalados;
- avisar o usuário sobre inconsistências ou incompatibilidades nas suas escolhas;
- determinar a ordem na qual os pacotes devem ser instalados;
- fazer a instalação ou remoção automaticamente; e
- guiar o usuário pelo processo de configuração exigido por cada pacote.

O `dselect` começa apresentando ao usuário um menu com 7 itens, cada um dos quais realiza uma ação específica. O usuário pode escolher uma das ações usando as teclas de setas para mover a barra destacada, e depois pressionando a tecla `<enter>` para escolher a ação destacada.

O que usuário vê depois depende da ação que ele escolheu. Se ele escolheu qualquer opção que não `Access` ou `Select`, o `dselect` simplesmente executará a ação especificada: por exemplo, se o usuário escolhesse a ação `Remove`, o `dselect` removeria todos os arquivos selecionados para remoção quando da última vez que o usuário escolheu a ação `Select`.

Tanto o item `Access` como o item `Select` levam a menus adicionais. Em ambos os casos, os menus são apresentados em uma tela dividida; a parte de cima apresenta uma lista de escolhas que pode ser rolada, ao passo que a parte de baixo apresenta uma breve explicação (“info”) para cada escolha.

Uma extensiva ajuda on-line está disponível, use a tecla `'?'` para ver uma tela de ajuda a qualquer momento.

A ordem na qual as ações são apresentadas no primeiro menu do `dselect` representa a ordem na qual um usuário normalmente escolheria para instalar pacotes. Entretanto, o usuário pode escolher quaisquer das opções do menu principal quantas vezes quiser (até mesmo nenhuma vez, dependendo do que se quer fazer).

- Comece escolhendo um “Access Method” (Método de Acesso). Este é o método pelo qual o usuário planeja acessar os pacotes Debian; por exemplo, alguns usuários possuem os pacotes Debian disponíveis em CD-ROM, enquanto outros planejam pegá-los através de ftp anônimo. O “Access Method” escolhido é gravado depois que o `dselect` termina, de modo que, se não for alterado, a opção não precisa ser chamada outra vez.
- Depois faça o **Update** (Atualização) da lista de pacotes disponíveis. Para fazer isso, o `dselect` lê o arquivo “Packages.gz” que deve estar incluso no diretório de nível mais alto do local onde os pacotes Debian a serem instalados estão armazenados. (Mas, se ele não estiver lá, o `dselect` se oferecerá para fazê-lo para você.)
- **Selecione** pacotes específicos para instalação em seu sistema. Depois de escolher esse item, o usuário é apresentado primeiramente a uma tela de ajuda (a menos que a opção de linha de comando ‘-expert’ tenha sido usada). Uma vez que o usuário saia da tela de ajuda, ele verá o menu da tela dividida para escolher pacotes a instalar (ou remover).

O topo da tela é uma janela relativamente estreita com a lista dos 4200 pacotes do Debian; a parte de baixo da tela contém descrições dos pacotes ou grupo de pacotes que estiverem destacados acima.

A pessoa pode especificar quais pacotes deveriam ser operados destacando um nome de pacote ou o “label” para um grupo de pacotes. Depois disso, você pode selecionar os pacotes:

Para ser instalado: Isso pode ser realizado pressionando-se a chave ‘+’.

Para ser removido: Pacotes podem ser removidos de duas maneiras:

- removidos: isso remove a maioria dos arquivos associados com o pacote, mas preserva os arquivos listados como arquivos de configuração (veja ‘O que é um conffile Debian?’ on page 30) e informações de configuração de pacotes. Isso é feito pressionando a chave ‘-’.
- expurgados: isso remove *todos* os arquivos que fazem partes do pacote. Isso é feito pressionando a chave ‘_’.

Perceba que não é possível remover “todos os pacotes”. Se você tentar isso, seu sistema se reduzirá aos pacotes base iniciais.

Para ser colocado em “hold” (espera) Isso é feito pressionando-se ‘=’, e ele efetivamente diz ao `dselect` para não atualizar, mesmo que a versão atualmente instalada em seu sistema não é tão recente quanto a versão disponível no repositório Debian que você está usando (isso foi especificado quando você quando você configurou o **Access Method**, e adquirido quando você usou **Update**).

Assim como você pode colocar um pacote em “hold”, você pode reverter esta opção pressionando ‘:’. Que diz ao `dselect` que o(s) pacote(s) podem ser atualizados se uma versão mais recente esteja disponível. Essa é a configuração padrão.

Você pode selecionar uma ordem diferente na qual os pacotes são apresentados, usando a chave ‘o’ para alternar entre as várias opções de classificação dos pacotes. A ordem padrão apresenta os pacotes por prioridade; dentro de cada prioridade, são apresentados

pacotes em ordem do diretório (a.k.a. sessão) do repositório onde eles estão armazenados. Dada esta ordem de classificação, alguns pacotes da seção A podem ser apresentados primeiro, seguidos por alguns pacotes da seção B, seguidos por mais pacotes (de menor prioridade) da seção A.

Você também pode ampliar os “labels” do topo da tela, usando a chave ‘v’ (verbose). Esta ação empurra a maior parte do texto que cabe dentro da tela, para fora da mesma, a direita. Para vê-lo, pressione a seta para a direita; para deslizar de volta para a esquerda, pressione a seta para a esquerda.

Se você selecionar um pacote para instalação ou remoção, por exemplo, `foo.deb`, e aquele pacote depende de (ou recomenda) outro pacote, por exemplo, `blurf.deb`, então o `dselect` apresentará a você uma subtela da tela de seleção principal. Nesta tela, você pode escolher dentre os pacotes relacionados, aceitando as ações sugeridas (para instalar ou não), ou rejeitando-as. Para remover as seleções feitas automaticamente, pressione Shift-D; para restaurar as pré-seleções feitas, pressione Shift-U. Em todo caso, você pode salvar suas seleções e retornar a tela principal de seleção pressionando Shift-Q.

- Ao retornar ao menu principal, o usuário pode então escolher o item “Install” para desempacotar e configurar os pacotes escolhidos. Alternativamente, os usuários que desejam remover arquivos podem escolher o item “Remove”. Em qualquer ponto, os usuários podem escolher “Quit” para sair do `dselect`; as escolhas do usuário são preservadas pelo `dselect`.

7.1.3 dpkg-deb

Este programa manipula arquivos de pacotes Debian (`.deb`). Alguns usos comuns são:

- Descobrir todas as opções: `dpkg-deb --help`.
- Determinar que arquivos estão contidos num arquivo de pacote Debian: `dpkg-deb --contents foo_VVV-RRR.deb`
- Extrair os arquivos contidos num arquivo Debian encontrado em um diretório especificado pelo usuário: `dpkg-deb --extract foo_VVV-RRR.deb tmp` extrai todos os arquivos de `foo_VVV-RRR.deb` no diretório `tmp/`. Isto é conveniente para examinar o conteúdo de um pacote num diretório localizado, sem instalar o pacote no sistema de arquivos raiz.

Perceba que qualquer pacote que foi desempacotado usando o `dpkg-deb --extract` será instalado incorretamente, você deverá usar `dpkg --install` ao invés.

Informações adicionais são dadas na página de manual `dpkg-deb(1)`.

7.1.4 apt-get

O `apt-get` oferece um modo simples para instalar pacotes a partir da linha de comando. Ao contrário do `dpkg`, o `apt-get` não entende arquivos `.deb`, ele trabalha com os nomes dos

pacotes e pode somente instalar arquivos `.deb` de uma fonte especificada no arquivo `/etc/apt/sources.list`.

Para mais informação, instale o pacote `apt` e leia `apt-get(8)`, `sources.list(5)` e `/usr/share/doc/apt/guide.html/index.html`.

7.1.5 `dpkg-split`

Este programa divide pacotes grandes em arquivos menores (por exemplo, para gravá-los em um conjunto de disquetes), e também pode ser usado para juntar um conjunto de arquivos num único arquivo. Pode ser usado apenas em um sistema Debian (ou seja, um sistema contendo o pacote `dpkg`), pois ele chama o programa `dpkg-deb` para analisar o pacote Debian em seus componentes.

Por exemplo, para dividir um arquivo `.deb` grande em `N` partes,

- Execute o comando `dpkg-split --split foo.deb`. Isto produzirá `N` arquivos, cada um com aproximadamente 460 KBytes no diretório atual.
- Copie esses `N` arquivos em disquetes.
- Copie o conteúdo dos disquetes no disco rígido desejado na outra máquina.
- Junte essas partes de arquivos usando `dpkg-split --join "foo*"`.

7.2 O Debian alega ser capaz de atualizar um programa que está sendo executado; como isto é realizado?

O kernel do sistema Debian GNU/Linux suporta a substituição de arquivos mesmo que eles estejam sendo usados.

Nós também oferecemos um programa chamado `start-stop-daemon` que é usado para iniciar daemons em tempo de boot, ou para parar daemons quando o runlevel do kernel é alterado (por exemplo, de `multi-user` para `single-user` ou para desligar). O mesmo programa é instalado para parar de rodar daemons, e reiniciá-los quando necessário.

7.3 Como posso descobrir que pacotes estão instalados num sistema Debian?

Para descobrir o estado de todos os pacotes instalados num sistema Debian, execute o comando:

```
dpkg --list
```

Isso mostra um resumo de uma linha para cada pacote, dando um símbolo de status (explicado no cabeçalho), o nome do pacote, a versão que é *instalada*, e uma breve descrição.

Para descobrir o status de pacotes cujos nomes correspondem a qualquer padrão da string começando com “foo”, execute o comando:

```
dpkg --list 'foo*'
```

Para adquirir um relatório mais completo para um pacote em particular, execute o comando:

```
dpkg --status NomeDoPacote
```

7.4 Como posso descobrir que pacote produziu um arquivo específico?

Para identificar o pacote que produziu o arquivo chamado `foo`, execute um dentre os comandos abaixo:

- `dpkg --search NomeDoArquivo`

Isto procura por `nome_do_arquivo` em pacotes instalados. (isto é atualmente equivalente a procurar todos os arquivos que possuam a extensão `.list` no diretório `/var/lib/dpkg/info/`, e ajustando a saída para mostrar os nomes de todos os pacotes que o contém).

- `zgrep foo Contents-ARCH.gz`

Isto procura por arquivos que possuam a substring `foo` em seus nomes. Os arquivos `Contents-ARQ.gz` (onde `ARQ` representa a arquitetura requerida) residem nos principais diretórios de pacotes (`main`, `non-free`, `contrib`) em um site FTP. Um arquivo `Contents` refere-se somente aos pacotes da árvore de subdiretórios onde ele reside. Além do mais, um usuário poderia ter que procurar mais que um arquivo `Contents` para encontrar o pacote contendo o arquivo `foo`.

Esse método possui a vantagem sobre o `dpkg --search` em que ele achará arquivos em pacotes que não estão atualmente instalados em seu sistema.

Capítulo 8

Mantendo seu Sistema Debian Atualizado.

Uma das metas do Debian é prover um caminho consistente de atualização e um processo seguro para realizar a mesma tarefa, e sempre fazemos o possível para que as atualizações de versões anteriores possam ser feitas da melhor forma possível. No caso em que haja alguma nota importante para acrescentar ao processo de atualização, o pacote irá alertar o usuário, e possivelmente oferecer uma solução para possíveis problemas.

Você também deverá ler as Release Notes (Notas de lançamento), documento que descreve os detalhes para atualizações específicas, enviado em todos os CDs do Debian, e disponível na WWW em <http://www.debian.org/releases/stable/i386/release-notes/>.

8.1 Como posso atualizar minha distribuição Debian 1.3.1 (ou anterior), baseada na libc5, para a 2.0 (ou posterior), baseada na libc6?

Há vários modos de atualizar:

- Usando um simples script de shell chamado `autoup.sh`, que atualiza os pacotes mais importantes. Depois que o `autoup.sh` terminou seu serviço, você pode usar o `dselect` para instalar os pacotes restantes em massa. Este é provavelmente o método recomendado, mas não o único.

Atualmente, a última versão de `autoup.sh` pode ser encontrada nos seguintes lugares:

- <http://www.br.debian.org/releases/2.0/autoup/>
- <http://www.taz.net.au/autoup/>
- <http://csanders.vicnet.net.au/autoup/>

- Seguindo atentamente o `libc5-libc6-Mini-HOWTO` (<http://ftp.debian.org/debian/doc/libc5-libc6-Mini-HOWTO.txt>) e atualizando os pacotes mais importantes na mão. O `autoup.sh` é baseado neste Mini-HOWTO, logo este método deve funcionar mais ou menos como o `autoup.sh`.
- Usando um `apt` baseado em `libc5`. APT significa “A Package Tool” (“Uma Ferramenta de Pacotes”), e pode substituir o `dselect` algum dia. Atualmente, ele funciona apenas como uma interface de linha de comando, ou como um método de acesso para o `dselect`. Você pode encontrar uma versão para `libc5` no diretório `dists/slink/main/upgrade-older-i386` nos repositórios Debian.
- Usando apenas o `dselect`, sem atualizar nenhum pacote à mão primeiro. É altamente recomendável que você NÃO use este método se puder evitar, porque atualmente o `dselect` sozinho não instala os pacotes na ordem ótima. O APT funciona muito melhor e é mais seguro.

8.2 Como posso manter meu sistema Debian atualizado?

A pessoa poderia simplesmente fazer um `ftp` anônimo a um repositório Debian, examinar os diretórios até achar o arquivo desejado, pegá-lo, e finalmente instalá-lo usando o `dpkg`. Observe que o `dpkg` instalará os arquivos atualizados em seu lugar correto, mesmo num sistema que esteja rodando. Às vezes, um pacote revisado precisa da instalação de uma versão revisada de outro pacote. Nesse caso, a instalação falhará a menos que o outro pacote seja instalado.

Muitas pessoas acham que essa abordagem consome muito tempo, já que o Debian se desentolve muito rapidamente - normalmente, uma dúzia ou mais pacotes são atualizados toda semana. Este número é maior logo antes do lançamento de uma nova versão. Para lidar com esta avalanche, muitas pessoas preferem usar programas automatizados. Vários pacotes diferentes estão disponíveis para este propósito:

8.2.1 APT

O APT é um sistema de administração de pacotes de software, isto é, binários Debian e pacotes fonte. O `apt-get` é a ferramenta de linha de comando para trabalhar com pacotes, e o método APT do `dselect` é uma interface para o APT através do `dselect`, ambos oferecem um modo mais simples e seguro de instalar e atualizar pacotes.

Para usar o método APT do `dselect`, execute o `dselect`, escolha a opção “0” (“Choose de access method to use” - “Escolha o método de acesso a ser usado”), destaque a opção “`apt`” e especifique a fonte do APT. Para fontes FTP e HTTP, você precisará do URL onde os diretórios Debian residem, nome da versão (você pode usar os nomes `stable/unstable`) e a seção da distribuição.

Se você quer usar os CDs para instalar os pacotes, insira cada CD do Debian no CD-ROM e rode o `apt-cdrom`. Então, quando selecionar o método APT você não necessitará alterar a lista de fontes.

As características do APT “ordenação completa de instalação”, “capacidade de múltiplas fontes” e várias outras características únicas, veja o User Guide em `/usr/share/doc/apt/guide.html/index.html`.

8.2.2 `dpkg-ftp`

Este é um método antigo. O APT é o método recomendado no momento.

Este é um método de acesso para o `dselect`. Ele pode ser chamado de dentro do `dselect`, permitindo assim que o usuário baixe arquivos e os instale diretamente em um único passo. Para fazer isso, execute o programa `dselect`, escolha a opção “0” (“Choose de access method to use” - “Escolha o método de acesso a ser usado”), destaque a opção “ftp” e especifique o hostname e diretório remoto. O `dpkg-ftp` baixará automaticamente os arquivos que forem selecionados (tanto nesta sessão do `dselect`, quanto nas anteriores).

Observe que, ao contrário do programa `mirror`, o `dpkg-ftp` não pega tudo que está num site espelho. Ao invés disso, ele baixa apenas os arquivos que você escolheu (da primeira vez que iniciou o `dpkg-ftp`), e que precisam ser atualizados.

8.2.3 `mirror`

Este script Perl, e seu (opcional) administrador de programas chamado `mirror-master`, pode ser usado para buscar partes específicas de uma árvore de diretórios de um host especificado *via* FTP anônimo.

O `mirror` é particularmente útil para baixar grandes volumes de softwares. Depois da primeira vez em que os arquivos foram baixados de um site, um arquivo chamado `.mirrorinfo` é armazenado no host local. Mudanças no filesystem remoto são detectadas automaticamente pelo `mirror`, que compara esse arquivo a um arquivo similar no sistema remoto e baixa somente os arquivos que sofreram mudanças.

O programa `mirror` é geralmente útil na atualização de cópias locais de árvores de diretórios remotas. Os arquivos procurados não precisam ser arquivos Debian (Como o `mirror` é um script Perl, ele pode também rodar em sistemas não Unix). Embora o programa `mirror` ofereça mecanismos para exclusão de nomes de arquivos que correspondam as strings especificadas pelos usuários, esse programa é muito útil quando o objetivo é baixar a árvore de diretórios inteira, ao invés de pacotes selecionados.

8.2.4 `dftp`

Este script em Perl pode ser usado para buscar pacotes Debian de um host específico. Ele começa baixando os arquivos `Packages.gz` dos diretórios especificados pelo usuário (por exemplo, `stable`, `contrib`, `non-free`) e mostra uma lista de pacotes. Estes são colocados em várias seções: novos upgrades, downgrades, novos pacotes, upgrades ignorados e pacotes ignorados. Daí, o usuário escolhe os pacotes desejados e o `dftp` os baixa e os instala. Isto faz com que seja

muito fácil ter seu sistema Debian 100% atualizado sem ter que baixar os pacotes que você não vai instalar.

O `dftp` pode ser usado para chamar o `dselect` (veja 'dselect' on page 38), fornecendo assim uma maneira integrada de buscar e atualizar os pacotes Debian de um sistema. Após o término da instalação, outro comando `dftp` pode ser usado para remover os arquivos de pacotes (".deb"). Mudanças no sistema de arquivos remoto são rastreadas automaticamente pelo `dftp`, que compara o arquivo `Packages.gz` local com os arquivos no sistema remoto.

8.2.5 `dpkg-mountable`

O `dpkg-mountable` adiciona um método de acesso chamado "mountable" a lista do `dselect`, que permite você instalar de qualquer sistema de arquivos especificado pelo `/etc/fstab` (por exemplo, uma partição de disco rígido normal, ou um servidor NFS), o qual ele irá automaticamente montar ou desmontar para você se necessário.

Ele também possui algumas características extras não encontradas nos métodos padrão do `dselect`, como abastecimento de uma árvore de arquivos local (tanto paralelo à distribuição principal, quanto totalmente separado), e somente obtendo pacotes que são necessários, ao invés do demorado exame de diretórios, como também registrando todas as ações do `dpkg` no método de instalação.

8.3 Preciso mudar para o modo monousuário para atualizar um pacote?

Não. Os pacotes podem ser atualizados, até em sistemas em execução. O Debian tem um programa chamado `start-stop-daemon`, que é chamado para parar, e depois reiniciar, se necessário, processos em execução, durante a atualização de um pacote.

8.4 Preciso manter todos aqueles arquivos .deb em meu disco?

Não. Se você tiver baixado os arquivos para seu disco (que não é absolutamente necessário, veja acima as descrições do `dpkg-ftp` ou do `dftp`), então, após você ter instalado os pacotes, você pode removê-los de seu sistema.

8.5 Como posso manter um registro dos pacotes que adicionei ao sistema?

O `dpkg` mantém um registro dos pacotes que foram desempacotados, configurados, removidos, e/ou apagados, mas não mantém (atualmente) um registro da atividade do terminal que ocorreu enquanto o pacote estava sendo manipulado. Alguns usuários superam esse problema simplesmente usando `tee`, assim:

```
dpkg -iGOEB -R main/binary non-free/binary contrib/binary | \  
tee -a /root/dpkg.log
```

O mesmo comando escrito usando opções longas:

```
dpkg --install --refuse-downgrade --selected-only \  
--skip-same-version --auto-deconfigure \  
--recursive main/binary non-free/binary contrib/binary | \  
tee -a /root/dpkg.log
```


Capítulo 9

Debian e o kernel.

9.1 Posso instalar e compilar um kernel sem alguma configuração específica para o Debian?

Sim. Mas você terá que entender a política do Debian em relação aos cabeçalhos.

As bibliotecas C do Debian são construídas com as versões *estáveis* mais recentes dos cabeçalhos do **kernel**.

Por exemplo, a versão Debian-1.2 usava a versão 5.4.13 dos cabeçalhos. Esta prática contrasta com os pacotes fonte do kernel Linux, distribuídos em todos os sites de FTP, que usam até mesmo as mais recentes versões dos cabeçalhos. Os cabeçalhos do kernel, distribuídos com os fontes do kernel, estão localizados em `/usr/include/linux/include/`.

Se você precisa compilar um programa com os cabeçalhos do kernel que são mais atuais do que aqueles providos por `libc6-dev`, então você terá que adicionar `-I/usr/src/linux/include/` na sua linha de comando quando estiver compilando. Isto veio em um ponto, por exemplo, com o empacotamento do “automounter daemon” (`amd`). Quando novos kernels mudaram algo em relação ao tratamento interno a NFS, o `amd` precisou saber sobre eles. Isto necessitou a inclusão dos mais recentes cabeçalhos do kernel.

9.2 Que ferramentas o Debian oferece para a construção de kernels personalizados?

Usuários que desejarem (ou precisarem) construir um kernel personalizado são incentivados a baixar o pacote `kernel-package`. Este pacote contém os scripts para construir o pacote do kernel, e oferece a capacidade de criar um pacote imagem do kernel somente rodando o comando

```
make-kpkg kernel_image
```

no diretório de mais alto nível dos fontes do kernel. Ajuda é obtida executando o comando

```
make-kpkg --help
```

e através da página de manual `make-kpkg(8)`.

Usuários devem baixar separadamente o código fonte do kernel mais recente (ou de seu kernel preferido) do seu repositório Linux favorito, a menos que um pacote `kernel-source-versão` esteja disponível (onde “`version`” significa a versão do kernel).

Instruções detalhadas sobre o uso do pacote `kernel-package` são dadas no arquivo `/usr/doc/kernel-package/README`. Resumidamente, a pessoa deve:

- Desempacotar os fontes do kernel, e dar `cd` para o diretório recém-criado.
- Modificar a configuração do kernel usando um destes comandos:
 - `make config` (para uma interface tty uma-linha-de-cada-vez).
 - `make menuconfig` (para uma interface de menus baseada em `ncurses`). Observe que, para usar esta opção, o pacote `libncurses5-dev` deve estar instalado.
 - `make xconfig` (para uma interface X11). Usar esta opção requer que os pacotes relevantes ao X e ao Tcl/Tk sejam instalados.

Qualquer dos passos acima gera um novo `.config` no diretório do fonte do kernel.

- Execute o comando: `make-kpkg -rev Custom.N kernel_image`, onde `N` é um número de revisão designado pelo usuário. O novo arquivo Debian assim formado teria revisão `Custom.1`, por exemplo, `kernel-image-2.2.14_Custom.1_i386.deb` para o kernel Linux 2.2.14.
- Instale o pacote criado.
 - Rode `dpkg --install /usr/src/kernel-image-VVV_Custom.N.deb` para instalar o kernel propriamente dito. O script de instalação irá:
 - * executar o boot loader, LILO (se estiver instalado),
 - * instalar o kernel personalizado em `/boot/vmlinuz_VVV-Custom.N`, e configurar as ligações simbólicas apropriadas ao kernel mais recente.
 - * sugerir ao usuário que crie um disquete de inicialização. Este disquete conterá apenas o kernel. Veja ‘Como faço um disquete de inicialização personalizado?’ on the next page.
 - Para usar um boot loader secundário como o `grub` ou `loadlin`, copie essa imagem para outros lugares (por exemplo, para `/boot/grub` ou para uma partição MS-DOS).

9.3 Como faço um disquete de inicialização personalizado?

Essa tarefa é muito facilitada pelo pacote Debian `boot-floppies`, normalmente encontrado na seção `admin` do repositório FTP do Debian. Scripts de shell desse pacote produzem disquetes de inicialização no formato `SYSLINUX`. Esses disquetes são formatados como `MS-DOS` cujos `MBRs` foram alterados para inicializarem o Linux diretamente (ou qualquer outro sistema operacional que tenha sido definido no arquivo `syslinux.cfg` do disquete). Outros scripts desse pacote produzem discos `root` de emergência e podem até reproduzir os “base disks” (os discos do sistema básico).

Você achará mais informações sobre isto no arquivo `/usr/doc/boot-floppies/README` após instalar o pacote `boot-floppies`.

9.4 Que ferramentas especiais o Debian oferece para lidar com módulos?

O pacote Debian `modconf` fornece um script shell (`/usr/sbin/modconf`) que pode ser usado para personalizar a configuração dos módulos. Este script apresenta uma interface baseada em menus, requisitando ao usuário detalhes sobre os drivers de dispositivos carregáveis de seu sistema. As respostas são usadas para personalizar o arquivo `/etc/modules.conf` que lista “alias” (apelidos), e outros argumentos (que devem ser usados em conjunto com vários módulos) através de arquivos em `/etc/modules` (que lista os módulos que devem ser carregados na hora do boot).

Como os (novos) arquivos `Configure.help`, que agora estão disponíveis para auxiliar a construção de kernels personalizados, o pacote `modconf` vem com uma série de arquivos de ajuda (em `/usr/lib/modules_help/`) que fornecem informações detalhadas sobre os argumentos apropriados para cada um dos módulos.

9.5 Posso desinstalar um kernel antigo de forma segura? Como?

Sim. O script `kernel-image-NNN.prerm` verifica se o kernel que você está usando atualmente é o mesmo que você está querendo desinstalar. Portanto, você pode remover pacotes de kernel indesejáveis usando este comando:

```
dpkg --purge --force-remove-essential kernel-image-NNN
```

(claro, substitua “NNN” com sua versão de kernel e número de revisão)

Capítulo 10

Personalizando sua instalação do Debian GNU/Linux.

10.1 Como posso garantir que todos os programas usem o mesmo tamanho de papel?

O arquivo `/etc/papersize` contém o nome do tamanho padrão do papel (como carta ou A4). Ele pode ser sobrescrito usando a variável de ambiente `PAPERSIZE`. Para detalhes, veja a página de manual `papersize(5)`.

10.2 Como posso fornecer acesso aos periféricos sem comprometer a segurança?

Vários arquivos de dispositivo no diretório `/dev` pertencem a grupos pré-definidos. Por exemplo, `/dev/fd0` pertence ao grupo `floppy`, e o `/dev/dsp` pertence ao grupo `audio`.

Se você deseja que um certo usuário tenha acesso a um desses dispositivos, somente adicione o usuário ao grupo ao qual o dispositivo pertence, por exemplo, faça:

```
adduser user group
```

Desse modo, você não terá que usar `chmod` no arquivo de dispositivo.

10.3 Como carrego uma fonte de console na inicialização à moda Debian?

O pacote `kbd` e `console-tools` suportam isso, edite o arquivo `/etc/kbd/config` ou o arquivo `/etc/console-tools/config`.

10.4 Como configuro os defaults de um programa para X11?

Os programas X do Debian irão instalar seus dados de recursos para os aplicativos em `/etc/X11/app-defaults/`. Se quiser personalizar aplicativos do X globalmente, coloque suas personalizações nesses arquivos. Eles são marcados como arquivos de configuração, portanto, seus conteúdos serão preservados durante atualizações.

10.5 Parece que cada distribuição possui um método de inicialização diferente. Fale-me sobre o do Debian.

Como todos os Unices, o Debian é inicializado executando-se o programa `init`. O arquivo de configuração para o `init` (que é o `/etc/inittab`) especifica que o primeiro script a ser executado deve ser o `/etc/init.d/rcS`. Este script verifica e monta sistemas de arquivos, carrega módulos, inicia os serviços de rede (chamando o script `/etc/init.d/network`), ajusta o relógio, faz outras inicializações, e daí, executa todos os scripts (exceto os que tiverem `'` no nome-de-arquivo) em `/etc/rc.boot/`. Qualquer script no diretório anterior é normalmente reservado para uso do administrador do sistema, e o uso deles em pacotes é desaprovado.

Após completar o processo de inicialização, o `init` executa todos os scripts “start” do diretório especificado pelo runlevel padrão (este runlevel é dado pela entrada `id` no `/etc/inittab`). Como a maioria dos Unices compatíveis com o System V, o Linux possui 7 runlevels:

- 0 (para o sistema),
- 1 (modo monousuário),
- 2 até 5 (vários modos multiusuários), e
- 6 (reinicializa o sistema).

Sistemas Debian vêm com `id=2`, que indica que o runlevel padrão será o ‘2’ quando o sistema entrar no modo multiusuário, e os scripts de `/etc/rc2.d/` serão executados.

Na verdade, os scripts em quaisquer dos diretórios `/etc/rcN.d/` são apenas ligações simbólicas para scripts em `/etc/init.d/`. Porém, os *nomes* dos arquivos em cada um dos diretórios `/etc/rcN.d/` são escolhidos de modo a indicar a *maneira* em que os scripts de `/etc/init.d/` serão executados. Especificamente, antes de entrar em qualquer runlevel, todos os scripts que começam com a letra ‘K’ são executados; esses scripts “matam” (terminam) os serviços. Depois, todos os scripts que começam com a letra ‘S’ são executados; esses scripts iniciam serviços. O número de dois dígitos após o ‘K’ ou ‘S’ indica a ordem na qual o script é executado. Os scripts com números menores são executados primeiro.

Esta abordagem funciona porque todos os scripts em `/etc/init.d/` aceitam um argumento que pode ser tanto ‘start’ (iniciar), ‘stop’ (parar), ou ‘reload’ (recarregar), ‘restart’ (reiniciar) ou ‘force-reload’ (força/obriga a recarga) e fazem a tarefa indicada pelo argumento. Esses scripts

podem ser usados mesmo depois que o sistema tiver sido inicializado, para controlar os vários processos.

Por exemplo, com o argumento 'reload' o comando

```
/etc/init.d/sendmail reload
```

envia ao daemon do sendmail um sinal para reler seu arquivo de configuração.

10.6 Parece que o Debian não usa o arquivo `rc.local` para personalizar o processo de inicialização; que recursos são fornecidos?

Suponha que um sistema precisa executar o script `foo` na inicialização, ou uma entrada para um runlevel (System V) específico. Então, o administrador do sistema deve:

- Colocar o script `foo` no diretório `/etc/init.d/`.
- Executar o comando `update-rc.d` do Debian com os argumentos apropriados, para estabelecer as ligações entre os diretórios `rc?.d` e `/etc/init.d/foo` (especificados na linha de comando). Aqui, '?' é um número de 0 a 6 e corresponde a cada um dos runlevels do System V.
- Reinicializar o sistema.

O comando `update-rc.d` estabelecerá as ligações entre os arquivos dos diretórios `rc?.d` e o script em `/etc/init.d/`. Cada ligação começará com um 'S' ou um 'K', seguido de um número, seguido do nome do script. Os scripts de `/etc/rcN.d/` que começam com a letra 'S' são executados quando se entra no runlevel N. Os scripts que começam com a letra 'K' são executados quando se deixa o runlevel N.

Pode-se, por exemplo, fazer com que o script `foo` seja executado na inicialização, colocando-o em `/etc/init.d/` e instalando as ligações com `update-rc.d foo defaults 19`. O argumento 'defaults' refere-se aos runlevels padrões, que vão de 2 a 5. O argumento '19' assegura que `foo` será chamado antes de quaisquer scripts que tenham números iguais ou maiores a 20.

10.7 Como o sistema de gerenciamento de pacotes lida com pacotes que contêm arquivos de configuração para outros pacotes?

Alguns usuários gostariam de criar, por exemplo, um novo servidor instalando um grupo de pacotes Debian e um pacote gerado localmente que consiste em arquivos de configuração. Isso geralmente não é uma boa idéia, porque o `dpkg` não saberá que esses arquivos de configuração

existem se eles estiverem num pacote diferente, e pode escrever configurações conflitantes quando um pacote do “grupo” inicial for atualizado.

Ao invés disso, crie um pacote local que modifica os arquivos de configuração do “grupo” de pacotes Debian em questão. Daí, o `dpkg` e o resto do sistema de gerenciamento de pacotes verá que os arquivos foram modificados pelo “administrador” local e não tentará sobrescrevê-los quando esses pacotes forem atualizados.

10.8 Como posso anulo um arquivo instalado por um pacote, de modo que uma versão diferente possa ser usada?

Suponha que um administrador ou um usuário local queira usar um programa “login-local” ao invés do programa “login” fornecido pelo pacote Debian `login`.

Não faça:

- Sobrescreva `/bin/login` com `login-local`.

O sistema de gerenciamento de pacotes não saberá sobre esta mudança, e simplesmente sobrescreverá seu `/bin/login` personalizado quando `login` (ou qualquer pacote que forneça o `/bin/login`) seja instalado ou atualizado.

Ao invés disso, faça

- Execute:

```
dpkg-divert --divert /bin/login.debian /bin/login
```

para que todas as futuras instalações de pacotes `login` do Debian escrevam o arquivo `/bin/login` com o nome `/bin/login.debian`.

- Então execute:

```
cp login-local /bin/login
```

para mover seu programa construído localmente ao local correto.

Detalhes são dados na página de manual `dpkg-divert(8)`.

10.9 Como posso incluir meu pacote feito localmente, na lista de pacotes disponíveis que o sistema de gerenciamento de pacotes usa?

Execute o comando:

```
dpkg-scanpackages BIN_DIR OVERRIDE_FILE [PATHPREFIX] > my_Packages
```

onde:

- BIN-DIR é um diretório onde arquivos de pacotes Debian (que normalmente têm extensão “.deb”) são armazenados.
- OVERRIDE_FILE é um arquivo que é editado pelos mantenedores da distribuição e é normalmente armazenado num repositório de FTP da Debian, em `indices/override.main.gz` para os pacotes Debian na distribuição “main”. Você pode ignorar isso para arquivos locais.
- PATHPREFIX é um texto *opcional* que pode ser prefixado ao arquivo `Packages.new` sendo produzido.

Uma vez que você tenha construído o arquivo `my_Packages`, conte ao sistema de gerenciamento de pacotes sobre ele, usando o comando:

```
dpkg --merge-avail my_Packages
```

Se você está usando o APT, você pode adicionar o repositório local ao seu arquivo `sources.list(5)` também.

10.10 Alguns usuários gostam do mawk, outros do gawk; alguns gostam do vim, outros do elvis; alguns gostam do trn, outros do tin; como o Debian lida com diversidade?

Existem vários casos onde dois pacotes fornecem duas versões diferentes de um programa, ambos fornecendo a mesma funcionalidade. Usuários podem preferir uma ao invés de outra fora de hábito, ou por causa da interface de usuário de um pacote ser um tanto mais agradável do que a interface de outro. Outros usuários do mesmo sistema podem realizar diferentes escolhas.

O Debian usa um sistema de pacotes “virtual” que permite aos administradores do sistema escolher (ou deixar que os usuários escolham) suas ferramentas favoritas quando houver duas

ou mais que forneçam a mesma funcionalidade básica, e ainda satisfazer requerimentos de dependência de pacotes sem especificar um pacote específico.

Por exemplo, podem existir duas versões diferentes de leitores de news num sistema. O pacote do servidor de news pode “recomendar” que haja *algum* leitor de news no sistema, mas a escolha entre `tin` ou `trn` fica a cargo do usuário individual. Isto é satisfeito fazendo com que ambos os pacotes `tin` e `trn` forneçam o pacote virtual `news-reader`. O programa a ser chamado é determinado através de uma ligação que aponta de um arquivo com o nome do pacote virtual `/etc/alternatives/news-reader` para o arquivo escolhido, por exemplo, `/usr/bin/trn`.

Uma única ligação é insuficiente para dar suporte ao uso total de um programa alternativo; normalmente, páginas de manual, e possivelmente outros arquivos de suporte devem ser selecionados também. O script em Perl `update-alternatives` fornece uma maneira de assegurar que todos os arquivos associados a um pacote especificado sejam escolhidos como o padrão do sistema.

Por exemplo, para verificar quais executáveis fornecem o ‘x-window-manager’, rode:

```
update-alternatives --display x-window-manager
```

Se você deseja mudá-lo, rode:

```
update-alternatives --config x-window-manager
```

E siga as instruções na tela (basicamente, pressione o número próximo a entrada que deseja).

Se um pacote não se registrou como um gerenciador de janelas por alguma razão (registre um bug se isso for um erro), ou se você usa um gerenciador de janelas do diretório `/usr/local`, as seleções na tela não conterão suas entradas preferidas. Você pode atualizar a ligação através das opções de linha de comando, assim:

```
update-alternatives --install /usr/bin/x-window-manager \  
x-window-manager /usr/local/bin/wmaker-cvs 50
```

O primeiro argumento para a opção ‘`--install`’ é a ligação simbólica que aponta para `/etc/alternatives/NAME`, onde `NAME` é o segundo argumento. O terceiro argumento é o programa para o qual `/etc/alternatives/NAME` deveria apontar, e o quarto argumento é a prioridade (grandes valores significam que a alternativa será mais provavelmente escolhida automaticamente).

Para remover uma alternativa que você adicionou, simplesmente rode:

```
update-alternatives --remove x-window-manager /usr/local/bin/wmaker-cvs
```

Capítulo 11

Obtendo suporte para o Debian GNU/Linux.

11.1 Que outra documentação existe em/para um sistema Debian?

- Instruções de instalação para a versão atual: veja <http://www.br.debian.org/releases/stable/i386/install>.
- O Manual de empacotamento é a documentação principal sobre os aspectos técnicos da criação de pacotes Debian, tanto binários como fontes.
Você pode encontrá-lo no pacote `packaging-manual`, ou em <ftp://ftp.debian.org/debian/doc/package-developer/packaging.html.tar.gz>.
- O Policy Manual (Manual de políticas) documenta as exigências políticas para a distribuição, por exemplo, a estrutura e conteúdos do arquivo Debian, vários sobre o planejamento do sistema operacional, assim como requisitos técnicos que cada pacote tem que satisfazer para ser incluído da distribuição.
Obtenha-o através do pacote `debian-policy`, ou em <ftp://ftp.debian.org/debian/doc/package-developer/policy.html.tar.gz>.
- Documentação sobre pacotes Debian instalados: a maioria dos pacotes tem arquivos que são desempacotados em `/usr/doc/PACKAGE`.
- Documentação sobre o projeto Linux: o pacote `doc-linux` instala todas as mais recentes versões dos HOWTOs e mini-HOWTOs do Linux Documentation Project (Projeto de documentação do Linux) (<http://www.linuxdoc.org>).
- Páginas de manual estilo Unix: a maioria dos comandos tem páginas de manual escritas no estilo dos arquivos “man” do Unix. Elas são referenciadas pela seção do diretório “man” em que residem: por exemplo, `foo(3)` refere-se à página de manual que reside em `/usr/man/man3`, e pode ser chamada através do comando: `man 3 foo`, ou apenas `man foo` se a seção 3 for a única a conter uma página sobre `foo`.

Pode-se saber qual diretório de `/usr/share/man/` contém uma certa página de manual executando `man -w foo`.

Novos usuários do Debian devem observar que as páginas de manual de muitos comandos gerais do sistema não estão disponíveis até que estes pacotes sejam instalados:

- `man-db`, que contém o próprio programa `man`, e outros programas para manipular as páginas de manual.
- `manpages`, que contém o sistema de páginas de manual. (veja ‘Como o Debian suporta idiomas diferentes do Inglês?’ on page 18).
- Páginas “info” estilo GNU: documentação para vários comandos, especialmente ferramentas GNU, não está disponível em páginas de manual, e sim em arquivos “info” que podem ser lidos pela ferramenta GNU `info`, ou executando `M-x info` dentro do GNU Emacs, ou com algum outro visualizador de páginas Info.

Sua principal vantagem em relação às páginas de manual originais é que usam um sistema de hipertexto. Entretanto, *não* precisam da WWW; o `info` pode ser executado a partir de um console em modo texto. Ele foi projetado por Richard Stallman e precedeu a WWW.

Note que você pode acessar muita documentação em seu sistema usando um browser WWW, através dos comandos “`dwww`” ou “`dhelp`”, encontrados em seus respectivos pacotes.

11.2 Existem recursos on-line para discussão sobre o Debian?

Existem várias listas de discussão relacionadas ao Debian.

Num sistema que possua o pacote `doc-debian` instalado, existe uma lista completa de listas de discussão em `/usr/share/doc/debian/mailling-lists.txt`.

Para se inscrever na `debian-X` (para anúncios sobre o X, usuários, etc.), envie um mail para `debian-X-request@lists.debian.org` com a palavra “`subscribe`” no campo “Assunto:”. Se você possui um browser WWW que possua suporte a formulários, você pode se inscrever na lista usando o formulário on-line (<http://www.debian.org/MailingLists/subscribe>). Você também pode se retirar da lista utilizando o mesmo formulário.

O endereço de e-mail do administrador da lista é `<listmaster@lists.debian.org>`, em caso de você ter algum problema.

Arquivos das listas de discussão do Debian estão disponíveis via WWW em <http://lists.debian.org/>.

Usuários podem enviar perguntas a mantenedores de pacotes individuais, desde que seus endereços de e-mail sejam fornecidos no arquivo de controle (veja a seção ‘O que é um arquivo de controle Debian?’ on page 29) que é incluído em cada pacote. A pessoa pode também aprender o nome e e-mail do mantenedor procurando nos arquivo “`Packages`”, pois este arquivo é somente um conjunto dos arquivos de controle dos pacotes que se encontram em uma árvore

de diretórios particular. Para extrair um arquivo de controle de um pacote Debian particular, use o comando:

```
dpkg --info NomeDoPacote_VVV-RRR.deb
```

Usuários devem enviar perguntas não específicas ao Debian para um dos grupos USENET do Linux, que são nomeados `comp.os.linux.*` ou `linux.*`. Existem várias listas de newsgroups Usenet de Linux e outros recursos relacionados na WWW, por exemplo, nos endereços Linux Online (<http://www.linux.org/docs/usenet.html>) e Linux Journal (<http://www.linuxjournal.com/help/online.html>).

11.3 Qual é o código de conduta para as listas de discussão?

Quando usar as listas de discussão do Debian, por favor, siga estas regras:

- Não brigue; não é agradável. As pessoas que desenvolvem o Debian são Voluntárias, doando seus tempos, energia e dinheiro numa tentativa de desenvolverem juntas o projeto Debian.
- Não use linguajar inadequado;
- Tenha certeza de que está utilizando a lista adequada. *Nunca* envie sua inscrição para a própria lista.¹
- Veja a seção ‘Como posso informar sobre um bug no Debian?’ on the following page para informações sobre o relato de bugs.

11.3.1 Política sobre anúncios em listas de discussão.

As listas de discussão do Debian GNU/Linux aceitam publicidade comercial através de pagamento. Nós oferecemos isenção se você puder nos mostrar o comprovante de pagamento da doação de \$1000 (U.S.) ou mais para a Free Software Foundation ordenado pela “Debian”. Uma doação por anúncio, por favor. Se você não deseja doar, simplesmente envie seu anúncio para a lista, e o operador das listas de discussão irá cobrar \$1999 (U.S.). O operador da lista irá doar esse total, menos as despesas recolhimento desse total, para a FSF.

Através do ato de enviar seu anúncio, você concorda em aceitar a responsabilidade sobre a taxa, você concorda em indenizar o operador das listas de discussão contra qualquer reivindicação legal de você ou de outros com relação a seu anúncio, e você concorda em pagar qualquer despesa legal e empresarial que incorra no recente recolhimento. Nossa obrigação para com você é limitada ao esforço de boa-fé de entrega de sua mensagem.

Taxas reduzidas e/ou isenção de taxas estão disponíveis para anúncios relacionados ao Debian. Você deve consultar com antecedência o operador da lista de discussão se deseja obter uma redução da taxa ou isenção.

¹Use o endereço `nome_da_lista-REQUEST@lists.debian.org` para isso.

11.4 Existe uma maneira rápida de procurar informações sobre o Debian GNU/Linux?

Há vários mecanismos de busca que servem para localizar documentações relacionadas ao Debian.

- Site de pesquisa WWW do Debian (<http://www.br.debian.org/search>).
- Serviço de pesquisa de news DejaNews (<http://www.dejanews.com/>). Para descobrir que experiências as pessoas tiveram para achar drivers para controladores Western Digital, experimente procurar por esta frase: `linux & WD` Isso diz ao DejaNews para que relate quaisquer artigos contendo ambas as strings “linux” E “WD”. (como nota: quando um dos autores deste texto o usou, descobriu que sua placa WD (que ele possuía há apenas 6 meses) tinha sido declarada obsoleta pela Adaptec, agora que eles compraram a WD. Portanto, não havia mais drivers disponíveis (bendita Adaptec)).
- Quaisquer dos mecanismos de busca na web (como o AltaVista (<http://www.altavista.com/>) ou o Google (<http://www.google.com>)) também podem ser usados para procurar a Usenet (embora não sejam tão atualizados quanto o DejaNews).

Por exemplo, procurando pela string “`cgi-perl`” uma descrição mais detalhada é retornada para esse pacote do que uma breve descrição em seu arquivo de controle.

11.5 Existem registros de bugs conhecidos?

A distribuição Debian GNU/Linux possui um sistema de monitoramento de bugs (bug tracking system (BTS)) que armazena detalhes de bugs relatados por usuários e desenvolvedores. A cada bug é dado um número, e é mantido armazenado até que seja marcado como tendo sido resolvido.

Cópias dessas informações estão disponíveis em <http://www.br.debian.org/Bugs/>.

Um servidor de mail oferece acesso a base de dados do sistema de monitoramento de bugs via correio eletrônico. Para adquirir as instruções, envie um e-mail a `<request@bugs.debian.org>` com “help” no corpo da mensagem.

11.6 Como posso informar sobre um bug no Debian?

Se você encontrou um bug no Debian, por favor, leia as instruções para informar um bug no Debian. Essas instruções podem ser obtidas em um de vários métodos:

- Por FTP anônimo. Os sites espelhos do Debian possuem as instruções no arquivo `doc/bug-reporting.txt`.

- A partir da WWW. Uma cópia das instruções é apresentada em <http://www.br.debian.org/Bugs/Reporting>.
- Em qualquer sistema Debian que tenha o pacote `doc-debian` instalado. As instruções estão no arquivo `/usr/doc/debian/bug-reporting.txt`.

Você pode usar os pacotes `bug` ou `reportbug` que irá guiá-lo através pelo processo de registro e enviará as mensagens para o endereço apropriado, com alguns detalhes adicionais sobre o seu sistema adicionados automaticamente.

Se você deseja enviar o relatório com um MUA, envia a mensagem para `<submit@bugs.debian.org>`, onde a primeira linha contenha algo do tipo

```
Package: NomeDoPacote
```

(substitua “NomeDoPacote” com o nome do pacote). O resto da mensagem deve conter a descrição do bug (por favor, faça isso de forma mais ou menos detalhada), a versão do Debian que você está usando, e as versões pertinentes daquele pacote.

Espere adquirir um conhecimento automático de seu relatório de bug. Será automaticamente dado a ele um número de monitoramento de bug, adicionado ao registro de bugs e remetido a lista de discussão `debian-bugs-dist`.

Se alguém fosse identificar um bug que era comum a muitos programas, então, ao invés de reportar dúzias de relatórios de bugs similares, alguém poderia preferir enviar bugs individuais para `<maintonly@bugs.debian.org>` (ao invés do endereço `submit@...`) para avisar somente o mantenedor do respectivo pacote, e então enviar um relatório reduzido as listas de discussão `debian-devel` e/ou de `debian-bugs-dist`.

Adicionalmente, lá existe um verificador de pacotes Debian, chamado Lintian (<http://www.br.debian.org/lintian/>), que é projetado para verificar mecanicamente os pacotes Debian em relação a violações da política e erros comuns de empacotamento. Portanto, se você encontrar um bug que aparentemente aparece em outros pacotes também, será melhor você entrar em contato com o mantenedor do Lintian em `<lintian-maint@debian.org>` de forma que uma nova verificação para o Lintian seja escrita, ao invés de reportar um bug diretamente. Isso prevenirá que o bug apareça em futuras versões do pacote, ou em qualquer outro pacote da distribuição.

Você também pode usar `<quiet@bugs.debian.org>`, para enviar um relatório de bug ao BTS, sem os ter que enviar para a `debian-bugs-dist` ou para o mantenedor. Esse endereço é usado muito raramente, por exemplo, quando você quer enviar alguma informação menor para seu relatório, que deve somente ser adicionada ao seu registro, ou quando você quer adicionar algo ao registro do BTS, mas já enviou isso para o mantenedor.

Capítulo 12

Contribuindo para o projeto Debian.

Doações de tempo (para desenvolver novos pacotes, manter pacotes existentes, ou fornecer suporte ao usuário), recursos (para espelhar os repositórios de FTP e WWW), e dinheiro (para pagar novos equipamentos que possam ser usados em testes e hardware novo para os repositórios) podem ajudar o projeto.

12.1 Como posso me tornar um desenvolvedor de software Debian?

O desenvolvimento do Debian está aberto a todos, e é preciso que haja novos usuários com as habilidades adequadas e/ou vontade de aprender para manter pacotes existentes que foram abandonados por seus mantenedores anteriores, para desenvolver novos pacotes, e para fornecer suporte ao usuário.

Todos os detalhes sobre como se tornar um desenvolvedor Debian podem ser encontradas no pacote `developers-reference`. Logo, você deve instalar esse pacote e o ler atentamente.

Uma versão on-line do `developers-reference` está disponível em nosso site web (<http://www.debian.org/doc/packaging-manuals/developers-reference/>).

12.2 Como posso contribuir com recursos para o projeto Debian?

Como o projeto pretende fazer um corpo substancial de software rapidamente facilmente acessível através do globo, espelhos são necessários urgentemente. É desejável, mas não absolutamente necessário espelhar todos os arquivos. Por favor, visite a página tamanho dos espelhos Debian (<http://www.debian.org/mirror/size>) para informações sobre o espaço em disco necessário.

A maior parte do espelhamento é feita automaticamente por scripts, sem intervenção humana. Entretanto, às vezes acontece uma falha ou mudança no sistema que requer intervenção humana.

Se você tem uma conexão de alta velocidade à Internet, os recursos para espelhar toda ou parte da distribuição, e está disposto a dedicar seu tempo (ou encontrar alguém) para fazer uma manutenção regular do sistema, por favor entre em contato com <debian-admin@debian.org>.

12.3 Como posso contribuir financeiramente com o projeto Debian?

A pessoa pode fazer doações individuais a uma das duas organizações que são críticas ao desenvolvimento do projeto Debian.

12.3.1 Software in the Public Interest

Software in the Public Interest (SPI) é uma organização sem fins lucrativos do tipo IRS 501(c)(3), formada quando a FSF retirou seu patrocínio à Debian. O propósito da organização é desenvolver e distribuir software livre.

Nossos objetivos são bem parecidos com os da FSF, e estimulamos os programadores a usar a GNU General Public License (Licença Pública Geral GNU) em seus programas. Entretanto, nosso foco é ligeiramente diferente, no sentido de que estamos construindo e distribuindo um sistema Linux que difere em muitos detalhes técnicos do sistema GNU planejado pela FSF. Ainda nos comunicamos com a FSF, e cooperamos mandando para eles alterações nos softwares GNU e recomendando a nossos usuários fazer doações a FSF e ao projeto GNU.

A SPI pode ser contactada através do endereço: <http://www.spi-inc.org/>.

12.3.2 Free Software Foundation

Atualmente, não há nenhuma conexão formal entre o Debian e a Free Software Foundation. Entretanto, a Free Software Foundation é responsável por alguns dos componentes de software mais importantes do Debian, incluindo o compilador C da GNU, o GNU Emacs, e grande parte da biblioteca run-time do C, que é usada por todos os programas no sistema. A FSF abriu caminho ao software livre: eles escreveram a General Public License (Licença Geral Pública), que é usada em muitos dos programas do Debian, e inventaram o projeto “GNU”, para criar um sistema Unix inteiramente livre. O Debian deve ser considerado um descendente direto do sistema GNU.

A FSF pode ser contactada através do endereço: <http://www.fsf.org/>.

Capítulo 13

Redistribuindo o Debian GNU/Linux em um produto comercial.

13.1 Posso fazer e vender CDs Debian?

Vá em frente. Você não precisa de permissão para distribuir qualquer coisa que tenhamos *lançado*, de modo que você pode masterizar seu CD logo que o beta-teste terminar. Você não tem que nos pagar qualquer coisa. Claro que, todos os fabricantes de CD têm que honrar as licenças dos programas em Debian. Por exemplo, muitos dos programas são autorizados debaixo dos termos da licença GPL, que exige a distribuição do código fonte dos programas juntamente com seus binários.

Também, publicaremos uma lista dos fabricantes de CDs que doam dinheiro, software e tempo ao projeto Debian, e incentivamos os usuários a comprar desses fabricantes. Logo, fazer doações é uma boa propaganda.

13.2 O Debian pode ser empacotado com software não-livre?

Sim. Embora todos os componentes principais sejam software livre, nós fornecemos um diretório non-free (não-livre) para programas que não possam ser redistribuídos livremente.

Fabricantes de CDs *podem ou não* ser capazes de distribuir os programas que colocamos nesse diretório, dependendo dos termos da licença ou seus acordos particulares com os autores desse software. Os fabricantes de CDs também podem distribuir, no mesmo CD, software não-livre que obtiverem de outras fontes. Não há nada de novo nisso: muitos fabricantes já distribuem software livre e proprietário no mesmo CD. É claro que ainda incentivamos os autores de software a lançar seus programas como software livre.

13.3 Estou fazendo uma distribuição especial do Linux para um “mercado vertical”. Posso usar o Debian GNU/Linux como núcleo do sistema Linux e acrescentar meus próprios aplicativos em cima dele?

Sim. Por exemplo, uma pessoa está fazendo uma distribuição “Linux for Hams” (Linux para rádio-amadores), com programas especializados em rádio-amadorismo. Ele está partindo do Debian como “sistema base”, e acrescentando programas para controlar o transmissor, seguir satélites, etc. Todos os programas que ele acrescenta são empacotados com o sistema de pacotes Debian, de modo que seus usuários conseguirão atualizar facilmente quando ele lançar CDs posteriores.

Existem várias outras distribuições derivadas do Debian no mercado atualmente, como a Corel Linux (<http://www.corel.com/>) e a Storm Linux (<http://www.stormix.com/>), que são direcionados a diferentes tipos de públicos do que o do Debian GNU/Linux original, mas usam grande parte dos componentes em seus produtos.

O Debian também provê um mecanismo para permitir a desenvolvedores e administradores de sistemas instalar versões locais de arquivos selecionados de um modo tal que eles não sejam sobrescritos quando outros pacotes são atualizados. Isso é discutido mais adiante na pergunta ‘Como posso anulo um arquivo instalado por um pacote, de modo que uma versão diferente possa ser usada?’ on page 56.

13.4 Posso colocar meu programa proprietário num “pacote” Debian, para que ele possa ser instalado facilmente em qualquer sistema Debian?

Vá em frente. A ferramenta de pacotes é software livre; os pacotes podem ou não ser software livre, contudo, ela pode instalar todos eles.

Capítulo 14

Que tipo de mudanças são esperadas para a próxima versão do Debian GNU/Linux?

14.1 Mais segurança.

O Debian contém suporte a senhas shadow desde a versão 1.3. Além disso, a Linux Library of Pluggable Authentication Modules (a.k.a. libpam (<http://www.kernel.org/pub/linux/libs/pam/>);) que permite administradores de sistema escolher modos de autorização em uma aplicação base específica está disponível, e inicialmente preparado para autenticar via senhas shadow.

Incluindo total suporte para métodos avançados de autenticação como o Kerberos, RSBAC e outros em progresso.

14.2 Mais suporte para usuários que não possuem Inglês como idioma nativo.

O Debian já tem algum suporte para usuários cuja língua nativa não é o Inglês, veja 'Como o Debian suporta idiomas diferentes do Inglês?' on page 18.

Esperamos encontrar pessoas que proverão apoio para ainda mais idiomas, e traduzi-los. Alguns programas já suportam internacionalização, portanto, precisamos de tradutores de catálogo de mensagens. Muitos programas ainda precisam ser internacionalizados corretamente.

O projeto de tradução GNU <ftp://ftp.gnu.org/pub/gnu/ABOUT-NLS> trabalha na internacionalização dos programas GNU.

14.3 Mais arquiteturas.

Atualmente, está disponível uma versão completa do Debian para as arquiteturas Intel x86 (*i386*), Motorola 680x0 (*m68k*), Alpha (*alpha*), SPARC (*sparc*), PowerPC (*powerpc*) e ARM (*arm*).

Sistemas Debian completos para outras arquiteturas como MIPS e SPARC64 são esperados para breve.

14.4 Kernel Linux 2.4.

Esperamos que o próximo lançamento, atualmente de codinome *woody*, inclua a última versão do Kernel Linux 2.4.x.

Claro que, ele ainda oferecerá suporte e conterà pacotes para as versões 2.0.x e 2.2.x.

14.5 A Biblioteca C 2.2 da GNU.

Esperamos que a versão *woody* inclua a biblioteca LibC da GNU, versão 2.2.x.

Capítulo 15

Informações gerais sobre o FAQ.

15.1 Autores

A primeira edição deste FAQ foi feita e mantida por J.H.M. Dassen (Ray) e Chuck Stickelman. Autores do reescrito Debian GNU/Linux FAQ são Susan G. Kleinmann e Sven Rudolph. Depois deles, o FAQ foi mantido por Santiago Vila. O mantenedor atual é Josip Rodin.

Partes da informação vieram de:

- Do anúncio de lançamento do Debian-1.1, por Bruce Perens (<http://www.perens.com/>).
- The Linux FAQ, por Ian Jackson (<http://www.chiark.greenend.org.uk/~ijackson/>).
- Arquivos das listas de discussão Debian (<http://lists.debian.org/>),
- O manual dos programadores do `dpkg` e o manual da política do Debian (veja ‘Que outra documentação existe em/para um sistema Debian?’ on page 59)
- muitos desenvolvedores, voluntários e “beta testers”, e ...
- e as recordações de seus autores

Os autores gostariam de agradecer todos aqueles que ajudaram a tornar este documento possível.

Todas as garantias são negadas. Todas as marcas registradas são propriedades de seus respectivos donos.

15.2 Feedback

Comentários e adições a este documento sempre são bem-vindos. Por favor, envie um e-mail para <doc-debian@packages.debian.org>, ou submeta um relatório de bugs (problemas) contra o pacote doc-debian.

15.3 Disponibilidade

A mais recente versão deste documento pode ser vista nas páginas WWW do Debian em <http://www.br.debian.org/doc/FAQ/>.

Também está disponível para download nos formatos texto simples, HTML e PostScript no servidor de FTP do Debian (<ftp://ftp.debian.org/debian/>) e em um de seus espelhos (<http://www.debian.org/distrib/ftplist>) no diretório doc/FAQ/ (<ftp://ftp.debian.org/debian/doc/FAQ/>).

Os arquivos SGML originais, usados para criar este documento, estão disponíveis no pacote fonte doc-debian, ou no servidor CVS em: `:pserver:anonymous@cvs.debian.org:/cvs/debian-doc/ddp/manuals.sgml/faq`

15.4 Formato do documento

Este documento foi escrito usando o DebianDoc SGML DTD (reescrito do SGML em LinuxDoc). O sistema DebianDoc SGML nos permite criar arquivos em uma variedade de formatos a partir de um original, por exemplo, este documento pode ser visto como HTML, texto simples, TeX DVI, PostScript, PDF ou GNU info.

Utilitários de conversão para DebianDoc SGML estão disponíveis no pacote `debiandoc-sgml`.