

# **Notas de Lançamento para o Debian GNU/Linux 5.0 (lenny), Mips**

O Projeto de Documentação do Debian (<http://www.debian.org/doc/>)

31 de agosto de 2010

---

## Notas de Lançamento para o Debian GNU/Linux 5.0 (lenny), Mips

Published 2009-02-14

This document is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License, version 2, as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

O texto da licença também pode ser encontrado em <http://www.gnu.org/copyleft/gpl.html> e `/usr/share/common-licenses/GPL-2` no Debian GNU/Linux.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Reportando bugs neste documento	3
1.2	Enviando relatórios de atualização	3
1.3	Códigos fonte para este documento	4
<b>2</b>	<b>Quais as novidades no Debian GNU/Linux 5.0</b>	<b>5</b>
2.1	Quais as novidades na distribuição?	5
2.1.1	Gerenciamento de pacotes	6
2.1.2	A seção proposed-updates	6
2.2	Melhorias do sistema	7
2.3	Grandes mudanças relacionadas ao kernel	7
2.3.1	Mudanças no empacotamento do kernel	8
2.4	Emdebian 1.0 (baseado no Debian GNU/Linux lenny 5.0)	8
2.5	Suporte a netbook	8
2.6	Java agora no Debian	8
<b>3</b>	<b>Sistema de instalação</b>	<b>9</b>
3.1	Quais as novidades do sistema de instalação?	9
3.1.1	Major changes	9
3.1.2	Instalação automatizada	10
3.1.3	Instalar Debian GNU/Linux com um display Braille	10
3.1.3.1	Exemplos	11
3.1.4	Instalar o Debian GNU/Linux com um hardware de síntese de voz	11
<b>4</b>	<b>Atualizações a partir de versões anteriores</b>	<b>13</b>
4.1	Preparando-se para a atualização	13
4.1.1	Copie seus dados (back up) e informações de configuração	13
4.1.1.1	Certifique-se de que você está em um kernel adequado	13
4.1.2	Inform users in advance	13
4.1.3	Prepare for recovery	14
4.1.3.1	Debug shell during boot using initrd	14
4.1.4	Prepare a safe environment for the upgrade	14
4.2	Checking system status	15
4.2.1	Review actions pending in package manager	15
4.2.2	Disabling APT pinning	15
4.2.3	Checking packages status	15
4.2.4	The proposed-updates section	16
4.2.5	Unofficial sources and backports	16
4.2.5.1	Using backports.org packages	16
4.3	Manually unmarking packages	16
4.4	Preparing sources for APT	17
4.4.1	Adding APT Internet sources	17
4.4.2	Adding APT sources for a local mirror	17
4.4.3	Adding APT source from CD-ROM or DVD	18
4.5	Upgrading packages	18
4.5.1	Recording the session	18
4.5.2	Updating the package list	19
4.5.3	Make sure you have sufficient space for the upgrade	19
4.5.4	Upgrade apt and/or aptitude first	20
4.5.5	Using aptitude's list of automatically-installed packages with apt	21
4.5.6	Minimal system upgrade	21
4.5.6.1	Upgrading a desktop system	21
4.5.6.2	Upgrading a system with no X support installed	21
4.5.7	Upgrading the rest of the system	22

---

4.5.8	Possible issues during upgrade	22
4.6	Upgrading your kernel and related packages	23
4.6.1	Installing the kernel metapackage	23
4.6.2	Upgrading from a 2.6 kernel	23
4.6.3	Upgrading from a 2.4 kernel	24
4.6.4	Device enumeration reordering	24
4.6.5	Boot timing issues	24
4.7	Things to do before rebooting	25
4.7.1	Rerun lilo	25
4.8	System boot hangs on Waiting for root file system	25
4.8.1	How to avoid the problem before upgrading	25
4.8.2	How to recover from the problem after the upgrade	27
4.8.2.1	Solution 1	27
4.8.2.2	Solution 2	27
4.8.2.3	Solution 3	28
4.9	Preparing for the next release	29
4.10	Deprecated packages	29
4.11	Obsolete packages	29
4.11.1	Dummy packages	30
<b>5</b>	<b>Problemas a serem considerados para o lenny</b>	<b>31</b>
5.1	Problemas potenciais	31
5.1.1	Problemas com dispositivos relacionados ao udev	31
5.1.2	Algumas aplicações deixaram de funcionar com um kernel 2.4	31
5.1.3	Alguns locais de rede não podem ser alcançados pelo TCP	31
5.1.4	Desligamento automático deixa de funcionar	31
5.1.5	Inicialização de rede assíncrona pode causar comportamento imprevisível	32
5.1.6	Problema quando usando redes sem-fio protegidas com WPA	32
5.1.7	Problemas com caracteres não-ASCII em nomes de arquivo	32
5.1.8	Som deixa de funcionar	32
5.2	Montagens de NFS agora controladas pelo nfs-common	33
5.3	Mudança do layout de teclado romeno (ro)	33
5.4	Atualizando o apache2	33
5.5	NIS e Network Manager	33
5.6	Aspectos de segurança dos produtos Mozilla	33
5.7	Suporte e mudanças na área de trabalho GNOME	34
5.8	Não há suporte padrão para Unicode no emacs21*	34
5.9	slurpd/replica deixarão de funcionar	34
5.10	Área de trabalho não usa toda a tela	34
5.11	Problema com DHCP 'failover'	34
<b>6</b>	<b>Mais informações sobre o Debian GNU/Linux</b>	<b>35</b>
6.1	Leitura complementar	35
6.2	Obtendo ajuda	35
6.2.1	Listas de discussão	35
6.2.2	Internet Relay Chat	35
6.3	Reporting bugs	35
6.4	Contributing to Debian	36
<b>A</b>	<b>Gerenciado seu sistema etch</b>	<b>37</b>
A.1	Atualizando seu sistema etch	37
A.2	Verificando sua lista de fontes (sources list)	37
<b>B</b>	<b>Colaboradores das notas de lançamento</b>	<b>39</b>
<b>C</b>	<b>Lenny dedicado a Thiemo Seufer</b>	<b>41</b>
<b>D</b>	<b>Glossário</b>	<b>43</b>
	<b>Índice Remissivo</b>	<b>45</b>

---

O Projeto de Documentação do Debian (<http://www.debian.org/doc/>)



# Capítulo 1

## Introdução

Este documento dá aos usuários da distribuição Debian GNU/Linux informações sobre grandes mudanças na versão 5.0 (codinome 'lenny').

As notas de lançamento fornecem informações sobre como atualizar de forma segura a partir da versão 4.0 (codinome etch) para a versão atual e dá aos usuários informações sobre potenciais problemas conhecidos que eles possam encontrar neste processo.

Você pode obter a versão mais recente deste documento em <http://www.debian.org/releases/lenny/releasenotes>. Em caso de dúvida, verifique a data na primeira página do documento para ter certeza de que você está lendo uma versão atualizada.

### CUIDADO



Note que é impossível listar todos os problemas conhecidos e portanto uma seleção foi feita baseada numa combinação da quantidade esperada e do impacto destes problemas.

Por favor, note que nós só damos suporte e documentamos a atualização a partir de uma versão anterior do Debian (neste caso, a atualização a partir da versão 4.0). Se você precisa atualizar a partir de versões mais antigas, nós sugerimos que você leia as edições anteriores das notas de lançamento e atualize para a versão 4.0 primeiro.

### 1.1 Reportando bugs neste documento

Nós tentamos testar todos os diferentes passos de atualizações descritos neste documento bem como antecipar todos os possíveis problemas que nossos usuários possam encontrar.

Apesar disso, se você acredita ter encontrado um bug (informação incorreta ou informação que está faltando), nesta documentação, por favor, envie um bug no [Sistema de Acompanhamento de Bugs](http://bugs.debian.org/) (<http://bugs.debian.org/>) ('Bug Tracking System') para o pacote `release-notes`.

### 1.2 Enviando relatórios de atualização

Nós apreciamos quaisquer informações dos usuários relacionadas a atualizações do etch para o lenny. Se você está interessado em compartilhar informação, por favor, envie um bug no [Sistema de Acompanhamento de Bugs](http://bugs.debian.org/) (<http://bugs.debian.org/>) ('Bug Tracking System') para o pacote `upgrade-reports` com os seus resultados. Nós pedimos que você compacte quaisquer anexos que venha a incluir (usando o `gzip`).

Por favor, inclua as seguintes informações quando enviar seu relatório de atualização:

- O estado da sua base de dados de pacotes antes e depois da atualização: a base de dados de estados do `dpkg` está disponível em `/var/lib/dpkg/status` e a informação do estado dos pacotes do `aptitude` está disponível em `/var/lib/aptitude/pkgstates`. Você deveria ter feito backup

antes da atualização conforme descrito em Seção 4.1.1, mas você também pode encontrar backups desta informação em `/var/backups`.

- Registros da sessão criados usando o comando **script**, conforme descrito em Seção 4.5.1.
- Seus logs do **apt**, disponíveis em `/var/log/apt/term.log` ou seus logs do **aptitude**, disponíveis em `/var/log/aptitude`.

#### NOTA



Você deveria separar algum tempo para revisar e remover qualquer informação sensível e/ou confidencial dos logs antes de incluí-los no relatório de bug pois a informação será disponibilizada em um banco de dados público.

## 1.3 Códigos fonte para este documento

O código fonte deste documento está no formato DocBook XML. A versão HTML é gerada usando `docbook-xsl` e `xsltproc`. A versão PDF é gerada usando `dbleatex` ou `xmlroff`. Os códigos fonte para as Notas de Lançamento estão disponíveis no repositório SVN do *Projeto de Documentação Debian*. Você pode usar a **interface web** (<http://svn.debian.org/viewsvn/ddp/manuals/trunk/release-notes/>) para acessar seus arquivos individualmente através da web e ver suas mudanças. Para mais informações sobre como acessar o SVN, por favor, consulte as **páginas de informação sobre o SVN do Projeto de Documentação Debian** (<http://www.debian.org/doc/cvs>).



## Capítulo 2

# Quais as novidades no Debian GNU/Linux 5.0

O [Wiki](http://wiki.debian.org/NewInLenny) (<http://wiki.debian.org/NewInLenny>) contém mais informações sobre este tópico. Este lançamento adiciona suporte oficial para a arquitetura ARM EABI (armel). Abaixo estão as arquiteturas oficialmente suportadas para pelo Debian GNU/Linux lenny:

- Intel x86 ('i386')
- Alpha ('alpha')
- SPARC ('sparc')
- PowerPC ('powerpc')
- ARM ('arm')
- MIPS ('mips' (big-endian) e 'mipsel' (little-endian))
- Intel Itanium ('ia64')
- HP PA-RISC ('hppa')
- S/390 ('s390')
- AMD64 ('amd64')
- ARM EABI ('armel')

Você pode ler mais sobre o estado dos portes e informações específicas sobre o porte para sua arquitetura nas [páginas web dos portes Debian](http://www.debian.org/ports/) (<http://www.debian.org/ports/>).

### 2.1 Quais as novidades na distribuição?

Esta nova versão do Debian vem com muito mais software que seu predecessor etch; a distribuição inclui mais de 7700 novos pacotes de um total de mais de 23200 pacotes. A maioria do software na distribuição foi atualizado: mais de 13400 pacotes de software (isto é 72% de todos os pacotes no etch). Além disso, um número significativo de pacotes (mais de 3100, 17% dos pacotes no etch) foram, por várias razões, removidos da distribuição. Você não verá atualizações para estes pacotes e eles serão marcados como 'obsoletos' nas interfaces de gerenciamento de pacotes.

Com esta versão, o Debian GNU/Linux atualiza do X.Org 7.1 para o X.Org 7.3.

Mais uma vez, o Debian GNU/Linux é distribuído com várias aplicações e ambiente de área de trabalho. Entre outros, estão inclusos os ambientes de área de trabalho GNOME 2.22<sup>1</sup>, KDE 3.5.10, Xfce 4.4.2 e LXDE 0.3.2.1+svn20080509. Aplicações de produtividade também foram atualizadas, incluindo as suítes de escritório OpenOffice.org 2.4.1 e KOffice 1.6.3 bem como o GNUcash 2.2.6, GNUmeric 1.8.3 e o Abiword 2.6.4.

---

<sup>1</sup> Com alguns módulos do GNOME 2.20.

Atualizações para outras aplicações de área de trabalho incluem a atualização para o Evolution 2.22.3 e Pidgin 2.4.3 (conhecido anteriormente como Gaim). A suíte Mozilla também foi atualizada: *iceweasel* (versão 3.0.6) é a versão sem marcas do navegador web Firefox e *icedove* (versão 2.0.0.19) é a versão sem marcas do cliente de e-mail Thunderbird.

Entre vários outros, esta versão também inclui as seguintes atualizações de software:

Pacote	Versão no 4.0 (etch)	Versão no 5.0 (lenny)
Apache	2.2.3	2.2.9
BIND (servidor DNS)	9.3.4	9.5.0
Cherokee (servidor web)	0.5.5	0.7.2
Courier (MTA)	0.53.3	0.60.0
Dia	0.95.0	0.96.1
Ekiga (cliente VoIP)	2.0.3	2.0.12
Exim (servidor e-mail padrão)	4.63	4.69
GNU Compiler Collection (Coleção de Compiladores GNU) como compilador padrão	4.1.1	4.3.2
GIMP	2.2.13	2.4.7
a biblioteca GNU C	2.3.6	2.7
lighttpd	1.4.13	1.4.19
maradns	1.2.12.04	1.3.07.09
MySQL	5.0.32	5.0.51a
OpenLDAP	2.3.30	2.4.11
OpenSSH	4.3	5.1p1
PHP	5.2.0	5.2.6
Postfix (MTA)	2.3.8	2.5.5
PostgreSQL	8.1.15	8.3.5
Python	2.4.4	2.5.2
Tomcat	5.5.20	5.5.26

A versão oficial do Debian GNU/Linux agora é distribuída em 4 ou 5 DVDs binários ou em 28 a 32 CDs binários (dependendo da arquitetura) e 4 DVDs de fontes ou 28 CDs de fontes. Adicionalmente, há um DVD *multi-arch* com um subconjunto da versão para as arquiteturas *amd64* e *i386*, juntamente com o código fonte. Pela primeira vez, o Debian GNU/Linux também é lançado em imagens Blu-ray, também para as arquiteturas *amd64* e *i386* e juntamente com o código fonte.

O Debian agora oferece suporte ao Linux Standards Base (LSB) versão 3.2. O Debian 4.0 oferecia suporte à versão 3.1.

### 2.1.1 Gerenciamento de pacotes

O programa preferido para gerenciamento de pacotes a partir da linha de comando é o **aptitude**, que é capaz de realizar as mesmas funções de gerenciamento de pacotes que o **apt-get** e se mostrou melhor na resolução de dependência. Se você ainda está usando o **dselect**, você deveria mudar para o **aptitude** como interface oficial para gerenciamento de pacotes.

Para o lenny um avançado mecanismo de resolução de conflitos foi implementado no **aptitude** que tentará encontrar a melhor solução se conflitos são detectados devido a mudanças nas dependências entre os pacotes.

### 2.1.2 A seção proposed-updates

Todas as mudanças para versão estável já lançada (e para a 'oldstable') passam por um período de teste antes de serem aceitas nos repositórios. Cada atualização da versão estável (ou 'oldstable') é chamada de atualização pontual. A preparação para atualizações pontuais é feita através do mecanismo `proposed-updates`.

Os pacotes podem entrar a área de `proposed-updates` de duas formas. Primeiro, pacotes com atualizações de segurança adicionadas ao `security.debian.org` são automaticamente adicionadas ao `proposed-updates`. Segundo, os desenvolvedores Debian GNU/Linux podem enviar novos pacotes diretamente para `proposed-updates`. A lista atual de pacotes pode

ser vista em <http://ftp-master.debian.org/proposed-updates.html> (<http://ftp-master.debian.org/proposed-updates.html>).

Se você deseja ajudar a testar as atualizações dos pacotes antes que eles sejam formalmente adicionados ao lançamento pontual, você pode fazê-lo adicionando a seção `proposed-updates` em seu arquivo `sources.list`:

```
deb      http://mirrors.kernel.org/debian lenny-proposed-updates main contrib
deb-src  http://mirrors.kernel.org/debian lenny-proposed-updates main contrib
```

A próxima vez que você executar o comando **aptitude update**, o sistema tomará conhecimento dos pacotes existentes na seção `proposed-updates` e os levará em consideração quando analisar pacotes para atualizar.

Este não é um recurso estritamente novo do Debian, mas é um recurso que não foi divulgado anteriormente.

## 2.2 Melhorias do sistema

Foram feitas várias mudanças na distribuição que beneficiarão novas instalações do lenny, mas podem não ser automaticamente aplicadas nas atualizações a partir do `etch`. Esta seção dá uma visão geral sobre as mudanças mais relevantes.

**SELinux na prioridade padrão (standard), mas não habilitado por padrão** O pacote necessários para o suporte ao SELinux (Security-Enhanced Linux) foram promovidos para a prioridade *standard*. Isto significa que eles serão instalador por padrão durante novas instalações. Para sistemas já instalados, você pode instalar o SELinux usando:

```
# aptitude install selinux-basics
```

Note que o suporte ao SELinux *não* é habilitado por padrão. Informações sobre a configuração e ativação do SELinux podem ser encontradas no [Wiki do Debian](http://wiki.debian.org/SELinux) (<http://wiki.debian.org/SELinux>).

**Novo daemon syslog padrão** O pacote `rsyslog` assume como daemon padrão para logs do kernel e do sistema no Debian 5.0, substituindo o `syslogd` e o `klogd`. Com as regras padrão de log, ele pode ser usado como um substituto direto; se você possui regras personalizadas, você deverá migrá-las para o novo arquivo de configuração, `/etc/rsyslog.conf`.

Usuários atualizando a partir do `etch` precisam instalar o `rsyslog` e remover o `sysklogd` manualmente. O daemon `syslog` padrão não é substituído automaticamente na atualização para o lenny.

**Melhor suporte a UTF-8** Várias aplicações serão configuradas para usar UTF-8 por padrão ou terão melhor suporte a UTF-8. Veja a página <http://wiki.debian.org/UTF8BrokenApps> (<http://wiki.debian.org/UTF8BrokenApps>) sobre aplicações que ainda têm dificuldades em lidar com UTF-8.

O [Wiki do Debian](http://wiki.debian.org/Etch2LennyUpgrade) (<http://wiki.debian.org/Etch2LennyUpgrade>) possui algumas informações adicionais sobre mudanças entre o `etch` e o lenny.

## 2.3 Grandes mudanças relacionadas ao kernel

O Debian GNU/Linux 5.0 vem com o kernel versão 2.6.26 para todas as arquiteturas.

Tivemos grande mudanças tanto no próprio kernel quando no empacotamento do kernel para o Debian. Algumas dessas mudanças complicaram o processo de atualização e podem, potencialmente, resultar em problemas durante a reinicialização do sistema após a atualização para o lenny. Esta seção dá uma visão geral das mudanças mais importantes; informações sobre como contornar potenciais problemas é incluída nos próximos capítulos.

### 2.3.1 Mudanças no empacotamento do kernel

**Novo subtipo (flavor) do OpenVZ** O Debian GNU/Linux 5.0 fornece imagens de kernel pré-construídas para o OpenVZ, uma segunda solução de virtualização alternativa ao suporte a Linux-VServer incluído no etch. As vantagens do OpenVZ incluem suporte a migração a quente (live migration) ao custo de uma carga ligeiramente maior.

**Unificação dos pacotes do kernel para x86** Em versões anteriores havia um subtipo especial de kernel para processadores 32-bit AMD Athlon/Duron/Sempron chamado `-k7`. Esta variante foi abandonada; um único subtipo `-686` agora cuida de todos os processadores AMD/Intel/VIA da classe 686.

Quando possível, pacotes de transição fictícios (‘dummy’ que dependem dos novos pacotes foram fornecidos em substituição aos pacotes abandonados.

## 2.4 Emdebian 1.0 (baseado no Debian GNU/Linux lenny 5.0)

Lenny agora contém ferramentas de construção para o Emdebian que permite que pacotes fonte Debian sejam construídos para outra plataforma (cross-build) e reduzidos para caber em sistemas embarcados ARM.

A distribuição Emdebian 1.0 contém pacotes ARM pré-construídos em quantidade suficiente para criar sistemas de arquivos raiz que podem ser personalizados para máquinas específicas e suas variantes. Kernels e módulos de kernel precisam ser fornecidos separadamente. Suporte para armel e i386 estão em desenvolvimento. Veja a [página web do Emdebian](http://www.emdebian.org/) (<http://www.emdebian.org/>) para mais informações.

## 2.5 Suporte a netbook

O Debian agora oferece suporte a netbooks, tais como o Eee PC da Asus. Para o Eee PC, veja o pacote `eeepc-acpi-scripts`. Além disso, o Debian tem um novo Ambiente Leve de Área de Trabalho X11 (Lightweight X11 Desktop Environment), `lxde`, que é benéfico para netbooks ou outros computadores com desempenho relativamente baixo.

## 2.6 Java agora no Debian

O OpenJDK Java Runtime Environment `openjdk-6-jre` e o Development Kit `openjdk-6-jdk`, necessários para execução de programas Java GUI e Webstart ou para construção de tais programas, agora estão no Debian. Os pacotes são construídos usando o suporte a construção do IcedTea e patches do projeto IcedTea.

## Capítulo 3

# Sistema de instalação

O Instalador Debian é o sistema de instalação oficial para o Debian. Ele oferece vários métodos de instalação. Os métodos disponíveis para instalar seu sistema dependem da sua arquitetura.

Imagens do instalador para o lenny podem ser encontradas juntamente com o Guia de Instalação no [site web do Debian](http://www.debian.org/releases/stable/debian-installer/) (<http://www.debian.org/releases/stable/debian-installer/>).

O Guia de Instalação também é incluído no primeiro CD/DVD dos conjuntos de CD/DVD oficiais do Debian, disponíveis em:

```
/doc/install/manual/idioma/index.html
```

Também pode ser do seu interesse verificar a [errata](http://www.debian.org/releases/stable/debian-installer/index#errata) (<http://www.debian.org/releases/stable/debian-installer/index#errata>) do `debian-installer` que contém uma lista de problemas conhecidos.

### 3.1 Quais as novidades do sistema de instalação?

Muito desenvolvimento foi feito no Instalador Debian desde seu primeiro lançamento oficial com o Debian GNU/Linux 3.1 (sarge) o que resultou tanto em melhorias nos hardwares para os quais há suporte quanto em alguns novos recursos muito interessantes.

Nas Notas de Lançamento nós listamos somente as grandes mudanças do instalador. Se você está interessado nas mudanças detalhadas desde o `etch`, por favor, verifique os anúncios de lançamento das versões beta e RC do lenny disponíveis a partir do [histórico de notícias](http://www.debian.org/devel/debian-installer/News/) (<http://www.debian.org/devel/debian-installer/News/>) do Instalador Debian.

#### 3.1.1 Major changes

**Suporte ao carregamento de firmware durante a instalação** Agora é possível carregar arquivos binários de firmware a partir de mídias removíveis quando eles são fornecidos separadamente das mídias de instalação Debian.

**Suporte à instalação a partir do Microsoft Windows** A mídia de instalação agora fornece uma aplicação que permite preparar o sistema para instalar o Debian a partir do ambiente Microsoft Windows.

#### SATA RAID support

**Atualização antecipada de pacotes com correções de segurança** Quando usado com acesso funcional à Internet, o instalador atualizará todos os pacotes que tenham sido atualizados desde o lançamento inicial do lenny. Esta atualização acontece durante o passo de instalação, antes do sistema instalado ser inicializado.

Como consequência, o sistema instalado tem menos chance de estar vulnerável a problemas de segurança que foram descobertos e corrigidos entre o momento do lançamento do lenny e a instalação.

**Suporte à *volatile*** O instalador pode opcionalmente configurar o sistema instalado para usar pacotes atualizados a partir do `volatile.debian.org`. Este repositório contém pacotes que fornecem dados que precisam ser atualizados regularmente, como definições de fusos horários, arquivos de assinatura de antivírus, etc.

**Novos portes** Agora há suporte para a arquitetura armel. Imagens para clientes Xen i386 também são fornecidas.

**Suporte para dispositivos de síntese de fala** O instalador agora inclui suporte para vários dispositivos projetados para fornecer síntese de fala, dessa forma melhorando sua acessibilidade para usuários com necessidades visuais especiais.

**Suporte a opções de montagem *relatime*** O instalador agora pode configurar partições com a opção de montagem `relatime`, de forma que os registros de tempo de acesso ('access time') a arquivos e diretórios só sejam atualizados se o registro de tempo de acesso anterior é mais recente do que os registros de tempo de modificação ('modify') ou de alteração ('change') atuais.

**Sincronização de relógio NTP durante a instalação** O relógio do computador agora é sincronizado com servidores NTP através da rede e durante a instalação de forma que o sistema instalado imediatamente tenha um relógio preciso.

**Novos idiomas** Graças ao grande esforço dos tradutores, o Debian agora pode ser instalado em 63 idiomas (50 usando a interface de usuário em modo texto e 13 para as quais só há suporte com a interface gráfica de usuário). Isto representa cinco idiomas a mais do que no etch. Idiomas adicionados neste lançamento incluem amhárico, marathi, irlandês, sami do norte e sérvio. Devido à falta de atualizações de tradução, um idioma foi removido neste lançamento: estoniano. Outro idioma que tinha sido desativado no etch foi reativado: gaulês.

Os idiomas que só podem ser selecionado usando o instalador gráfico pois seus conjuntos de caracteres não podem ser representados em um ambiente não-gráfico são: amhárico, bengali, dzongkha, gujarati, hindi, georgiano, khmer, malaialam, marathi, nepali, punkabi, tamil e tailandês.

**Escolha de país simplificada** A lista de seleção de país agora é agrupada por continentes, permitindo escolher um país mais facilmente, quando usuários não querem escolher os países associados com o idioma escolhido.

### 3.1.2 Instalação automatizada

Algumas mudanças mencionadas na seção anterior também implicam em mudanças no suporte do instalador a instalações automatizadas usando arquivos de pré-configuração. Isto significa que se você possuir arquivos de pré-configuração que funcionavam com o instalador do etch, você não deve esperar que eles funcionem sem nenhuma modificação com o novo instalador.

O **Guia de Instalação** (<http://www.debian.org/releases/stable/installmanual>) possui um apêndice atualizado com extensa documentação sobre o uso de pré-configuração.

### 3.1.3 Instalar Debian GNU/Linux com um display Braille

Você pode instalar o Debian GNU/Linux 5.0 (lenny) com um display Braille. O display Braille deve estar conectado a uma porta USB ou serial. Se seu display Braille está conectado a uma porta USB e supõe-se que a tabela americana de Braille deva ser usada, você pode pressionar **Enter** no menu de inicialização. O leitor de tela BrITTY é padronizado para procurar por displays Braille em portas USB e usar a tabela americana de Braille. Se o display Braille está conectado a uma porta serial ou se você quer usar uma tabela de Braille diferente, você deve primeiro pressionar a tecla **Tab** no menu de inicialização. Daí, você pode configurar o leitor de tela BrITTY com o parâmetro de kernel `brltty`.

Para o parâmetro, a seguinte sintaxe é válida:

```
brltty=driver,dispositivo,tabela
```

Todos os operandos são opcionais.

**driver** O driver para o display Braille usado. Aqui, você deve informar um código de duas letras ou a palavra `auto`. Se este argumento não é informado, o reconhecimento automático será ativado por padrão.

**dispositivo** O dispositivo pode ser informado relativo ao `/dev/` bem como uma especificação absoluta.

**tabela** Define a tabela Braille para o idioma desejado. Por padrão, a tabela US é usada.

### 3.1.3.1 Exemplos

```
brltty=ht, ttyS0, de
```

O 'Handy Tech Driver'. O display Braille está conectado à COM1. A tabela alemã de Braille é usada.

```
brltty=,, de
```

Aqui, somente a tabela alemã de Braille é especificada. Assim, o BrlTTY tentará encontrar um display Braille em uma porta USB.

## 3.1.4 Instalar o Debian GNU/Linux com um hardware de síntese de voz

### NOTA



Suporte a hardware de síntese de voz só está disponível no instalador em modo texto. No entanto, por questões de espaço, ele é ativado junto com o instalador gráfico que, de qualquer forma, precisa de mais espaço. Portanto, você precisa selecionar a opção `Graphical install` (Instalação Gráfica) no menu de inicialização.

O hardware de síntese de voz não pode ser detectado automaticamente. Você precisa anexar o parâmetro de inicialização `speakup.synth=driver` para informar ao Speakup qual driver deverá ser usado. Deve-se substituir `driver` pelo código do driver do seu dispositivo, veja <http://www.linux-speakup.org/spkguide.txt> para uma lista de nomes. Ao usar este parâmetro, a versão em modo texto do instalador será automaticamente selecionada e o suporte para a síntese de voz será automaticamente instalado no sistema alvo.





## Capítulo 4

# Atualizações a partir de versões anteriores

### 4.1 Preparando-se para a atualização

Nós sugerimos que antes de atualizar você também leia as informações em Capítulo 5. Esse capítulo cobre potenciais problemas que não estão diretamente relacionados ao processo de atualização mas que podem ser importantes de serem conhecidos antes de começar o processo.

#### 4.1.1 Copie seus dados (back up) e informações de configuração

Antes de atualizar seu sistema, é fortemente recomendado que você faça um backup completo ou, pelo menos, copie quaisquer dados ou informações de configuração que você não possa perder. As ferramentas de atualização e o processo são bem confiáveis, mas uma falha de hardware no meio de uma atualização pode resultar em um sistema severamente danificado.

As principais coisas que você terá que copiar são os conteúdos do `/etc`, `/var/lib/dpkg`, `/var/lib/aptitude/pkgstates` e a saída do comando `dpkg --get-selections "*" (as aspas são importantes)`.

O processo de atualização não modifica nada no diretório `/home`. No entanto, algumas aplicações (e.g. partes da suíte Mozilla e os ambientes de área de trabalho GNOME e KDE) sabidamente sobrescrevem configurações existentes de usuários com novos padrão quando uma nova versão da aplicação é iniciada pela primeira vez por um usuário. Como precaução, você pode ter interesse em copiar arquivos e diretório ocultos ('dotfiles') nos diretórios dos usuários. Esta cópia por ajudar a recuperar ou recriar antigas configurações. Também pode ser do seu interesse informar os usuários sobre isso.

Qualquer operação de instalação de pacote deve ser executada com privilégios de superusuário, por isso, acesse o sistema como `root` ou use o comando `su` ou `sudo` para obter os privilégios de acesso necessários.

A atualização possui algumas pré-condições; você deve verificá-las antes de começar a executar a atualização.

##### 4.1.1.1 Certifique-se de que você está em um kernel adequado

lenny's version of `glibc` will not work with kernels older than `2.6.8` on any architecture and some architectures have higher requirements. We strongly recommend that you upgrade to and test an `etch 2.6.18` or `2.6.24` kernel or a custom kernel of at least version `2.6.18` before beginning the upgrade process.

##### 4.1.2 Inform users in advance

It's wise to inform all users in advance of any upgrades you're planning, although users accessing your system via an `ssh` connection should notice little during the upgrade, and should be able to continue working.

If you wish to take extra precautions, back up or unmount the `/home` partition before upgrading.

You will probably have to do a kernel upgrade when upgrading to lenny, so a reboot will normally be necessary. Typically, this will be done after the upgrade is finished.

### 4.1.3 Prepare for recovery

Because of the many changes in the kernel between etch and lenny regarding drivers, hardware discovery and the naming and ordering of device files, there is a real risk that you may experience problems rebooting your system after the upgrade. A lot of known potential issues are documented in this and the next chapters of these Release Notes.

For that reason it makes sense to ensure that you will be able to recover if your system should fail to reboot or, for remotely managed systems, fail to bring up networking.

If you are upgrading remotely via an `ssh` link it is highly recommended that you take the necessary precautions to be able to access the server through a remote serial terminal. There is a chance that, after upgrading the kernel and rebooting, some devices will be renamed (as described in Seção 4.6.4) and you will have to fix the system configuration through a local console. Also, if the system is rebooted accidentally in the middle of an upgrade there is a chance you will need to recover using a local console.

The most obvious thing to try first is to reboot with your old kernel. However, for various reasons documented elsewhere in this document, this is not guaranteed to work.

If that fails, you will need an alternative way to boot your system so you can access and repair it. One option is to use a special rescue image or a Linux live CD. After booting from that, you should be able to mount your root file system and `chroot` into it to investigate and fix the problem.

Another option we'd like to recommend is to use the *rescue mode* of the lenny Debian Installer. The advantage of using the installer is that you can choose between its many installation methods for one that best suits your situation. For more information, please consult the section 'Recovering a Broken System' in chapter 8 of the [Installation Guide](http://www.debian.org/releases/stable/installmanual) (<http://www.debian.org/releases/stable/installmanual>) and the [Debian Installer FAQ](http://wiki.debian.org/DebianInstaller/FAQ) (<http://wiki.debian.org/DebianInstaller/FAQ>).

#### 4.1.3.1 Debug shell during boot using initrd

The `initramfs-tools` includes a debug shell<sup>1</sup> in the `initrd`s it generates. If for example the `initrd` is unable to mount your root file system, you will be dropped into this debug shell which has basic commands available to help trace the problem and possibly fix it.

Basic things to check are: presence of correct device files in `/dev`; what modules are loaded (`cat /proc/modules`); output of `dmesg` for errors loading drivers. The output of `dmesg` will also show what device files have been assigned to which disks; you should check that against the output of `echo $ROOT` to make sure that the root file system is on the expected device.

If you do manage to fix the problem, typing `exit` will quit the debug shell and continue the boot process at the point it failed. Of course you will also need to fix the underlying problem and regenerate the `initrd` so the next boot won't fail again.

### 4.1.4 Prepare a safe environment for the upgrade

The distribution upgrade should be done either locally from a textmode virtual console (or a directly connected serial terminal), or remotely via an `ssh` link.

In order to gain extra safety margin when upgrading remotely, we suggest that you run upgrade processes in the virtual console provided by the `screen` program, which enables safe reconnection and ensures the upgrade process is not interrupted even if the remote connection process fails.

#### IMPORTANTE



You should *not* upgrade using `telnet`, `rlogin`, `rsh`, or from an X session managed by `xdm`, `gdm` or `kdm` etc on the machine you are upgrading. That is because each of those services may well be terminated during the upgrade, which can result in an *inaccessible* system that is only half-upgraded.

<sup>1</sup> This feature can be disabled by adding the parameter `panic=0` to your boot parameters.

## 4.2 Checking system status

The upgrade process described in this chapter has been designed for upgrades from ‘pure’ *etch* systems without third-party packages. For the greatest reliability of the upgrade process, you may wish to remove third-party packages from your system before you begin upgrading.

This procedure also assumes your system has been updated to the latest point release of *etch*. If you have not done this or are unsure, follow the instructions in Seção [A.1](#).

### 4.2.1 Review actions pending in package manager

In some cases, the use of **apt-get** for installing packages instead of **aptitude** might make **aptitude** consider a package as ‘unused’ and schedule it for removal. In general, you should make sure the system is fully up-to-date and ‘clean’ before proceeding with the upgrade.

Because of this you should review if there are any pending actions in the package manager **aptitude**. If a package is scheduled for removal or update in the package manager, it might negatively impact the upgrade procedure. Note that correcting this is only possible if your `sources.list` still points to *etch* and not to *stable* or *lenny*; see Seção [A.2](#).

To perform this review, launch **aptitude** in ‘visual mode’ and press **g** (‘Go’). If it shows any actions, you should review them and either fix them or implement the suggested actions. If no actions are suggested you will be presented with a message saying ‘No packages are scheduled to be installed, removed, or upgraded’.

### 4.2.2 Disabling APT pinning

If you have configured APT to install certain packages from a distribution other than *stable* (e.g. from testing), you may have to change your APT pinning configuration (stored in `/etc/apt/preferences`) to allow the upgrade of packages to the versions in the new *stable* release. Further information on APT pinning can be found in `apt_preferences(5)`.

### 4.2.3 Checking packages status

Regardless of the method used for upgrading, it is recommended that you check the status of all packages first, and verify that all packages are in an upgradable state. The following command will show any packages which have a status of Half-Installed or Failed-Config, and those with any error status.

```
# dpkg --audit
```

You could also inspect the state of all packages on your system using **dselect**, **aptitude**, or with commands such as

```
# dpkg -l | pager
```

or

```
# dpkg --get-selections "*" > ~/curr-pkgs.txt
```

It is desirable to remove any holds before upgrading. If any package that is essential for the upgrade is on hold, the upgrade will fail.

Note that **aptitude** uses a different method for registering packages that are on hold than **apt-get** and **dselect**. You can identify packages on hold for **aptitude** with

```
# aptitude search "~ahold" | grep "^h"
```

If you want to check which packages you had on hold for **apt-get**, you should use

```
# dpkg --get-selections | grep hold
```

If you changed and recompiled a package locally, and didn’t rename it or put an epoch in the version, you must put it on hold to prevent it from being upgraded.

The ‘hold’ package state for **aptitude** can be changed using:

```
# aptitude hold package_name
```

Replace `hold` with `unhold` to unset the 'hold' state.

If there is anything you need to fix, it is best to make sure your `sources.list` still refers to `etch` as explained in Seção A.2.

#### 4.2.4 The proposed-updates section

If you have listed the `proposed-updates` section in your `/etc/apt/sources.list` file, you should remove it from that file before attempting to upgrade your system. This is a precaution to reduce the likelihood of conflicts.

#### 4.2.5 Unofficial sources and backports

If you have any non-Debian packages on your system, you should be aware that these may be removed during the upgrade because of conflicting dependencies. If these packages were installed by adding an extra package archive in your `/etc/apt/sources.list`, you should check if that archive also offers packages compiled for `lenny` and change the source line accordingly at the same time as your source lines for Debian packages.

Some users may have unofficial backported 'newer' versions of packages that *are* in Debian installed on their `etch` system. Such packages are most likely to cause problems during an upgrade as they may result in file conflicts<sup>2</sup>. Seção 4.5.8 has some information on how to deal with file conflicts if they should occur.

##### 4.2.5.1 Using backports.org packages

`backports.org` is a semi-official repository provided by Debian GNU/Linux developers, which provides newer packages for the stable release, based on a rebuild from the packages from the 'testing' archive.

The `backports.org` repository mainly contains packages from 'testing', with reduced version numbers so that the upgrade path from `etch` backports to `lenny` still works. However, there are a few backports which are made from unstable: security updates, plus the following exceptions: Firefox, the Linux kernel, OpenOffice.org, and X.Org.

If you do not use one of these exceptions, you can safely upgrade to `lenny`. If you use one of these exceptions, set the `Pin-Priority` (see `apt_preferences(5)`) temporarily to 1001 for all packages from `lenny`, and you should be able to do a safe `dist-upgrade` too. See the [backports FAQ](http://backports.org/dokuwiki/doku.php?id=faq) (<http://backports.org/dokuwiki/doku.php?id=faq>).

## 4.3 Manually unmarking packages

To prevent `aptitude` from removing some packages that were pulled in through dependencies, you need to manually unmark them as *auto* packages. This includes OpenOffice and Vim for desktop installs:

```
# aptitude unmarkauto openoffice.org vim
```

And 2.6 kernel images if you have installed them using a kernel metapackage:

```
# aptitude unmarkauto $(dpkg-query -W 'linux-image-2.6.*' | cut -f1)
```

### NOTA



You can review which packages are marked as *auto* in `aptitude` by running:

```
# aptitude search '~i~M'
```

<sup>2</sup> Debian's package management system normally does not allow a package to remove or replace a file owned by another package unless it has been defined to replace that package.

## 4.4 Preparing sources for APT

Before starting the upgrade you must set up apt's configuration file for package lists, `/etc/apt/sources.list`.

apt will consider all packages that can be found via any 'deb' line, and install the package with the highest version number, giving priority to the first line in the file (thus where you have multiple mirror locations, you'd typically first name a local hard disk, then CD-ROMs, and then HTTP/FTP mirrors).

### DICA



You might need to add an GPG checking exception for DVDs and CD-ROMs. Add the following line to `/etc/apt/apt.conf`, if it's not already in `/etc/apt/apt.conf.d/00trustedrom`:

```
APT::Authentication::TrustCDROM "true";
```

This does not work with DVD/CD-ROM image files, however.

A release can often be referred to both by its codename (e.g. `etch`, `lenny`) and by its status name (i.e. `oldstable`, `stable`, `testing`, `unstable`). Referring to a release by its codename has the advantage that you will never be surprised by a new release and for this reason is the approach taken here. It does of course mean that you will have to watch out for release announcements yourself. If you use the status name instead, you will just see loads of updates for packages available as soon as a release has happened.

### 4.4.1 Adding APT Internet sources

The default configuration is set up for installation from main Debian Internet servers, but you may wish to modify `/etc/apt/sources.list` to use other mirrors, preferably a mirror that is network-wise closest to you.

Debian HTTP or FTP mirror addresses can be found at <http://www.debian.org/distrib/ftplist> (look at the 'list of Debian mirrors' section). HTTP mirrors are generally speedier than FTP mirrors.

For example, suppose your closest Debian mirror is `http://mirrors.kernel.org`. When inspecting that mirror with a web browser or FTP program, you will notice that the main directories are organized like this:

```
http://mirrors.kernel.org/dists/lenny/main/binary-mips/...
http://mirrors.kernel.org/dists/lenny/contrib/binary-mips/...
```

To use this mirror with apt, you add this line to your `sources.list` file:

```
deb http://mirrors.kernel.org lenny main contrib
```

Note that the 'dists' is added implicitly, and the arguments after the release name are used to expand the path into multiple directories.

After adding your new sources, disable the previously existing 'deb' lines in `sources.list` by placing a hash sign (#) in front of them.

### 4.4.2 Adding APT sources for a local mirror

Instead of using HTTP or FTP package mirrors, you may wish to modify `/etc/apt/sources.list` to use a mirror on a local disk (possibly mounted over NFS).

For example, your package mirror may be under `/var/ftp/debian/`, and have main directories like this:

```
/var/ftp/debian/dists/lenny/main/binary-mips/...
/var/ftp/debian/dists/lenny/contrib/binary-mips/...
```

To use this with apt, add this line to your `sources.list` file:

```
deb file:/var/ftp/debian lenny main contrib
```

Note that the ‘dists’ is added implicitly, and the arguments after the release name are used to expand the path into multiple directories.

After adding your new sources, disable the previously existing ‘deb’ lines in `sources.list` by placing a hash sign (#) in front of them.

### 4.4.3 Adding APT source from CD-ROM or DVD

If you want to use CDs *only*, comment out the existing ‘deb’ lines in `/etc/apt/sources.list` by placing a hash sign (#) in front of them.

Make sure there is a line in `/etc/fstab` that enables mounting your CD-ROM drive at the `/cdrom` mount point (the exact `/cdrom` mount point is required for **apt-cdrom**). For example, if `/dev/hdc` is your CD-ROM drive, `/etc/fstab` should contain a line like:

```
/dev/hdc /cdrom auto defaults,noauto,ro 0 0
```

Note that there must be *no spaces* between the words `defaults,noauto,ro` in the fourth field. To verify it works, insert a CD and try running

```
# mount /cdrom      # this will mount the CD to the mount point
# ls -alF /cdrom    # this should show the CD's root directory
# umount /cdrom     # this will unmount the CD
```

Next, run:

```
# apt-cdrom add
```

for each Debian Binary CD-ROM you have, to add the data about each CD to APT’s database.

## 4.5 Upgrading packages

The recommended way to upgrade from previous Debian GNU/Linux releases is to use the package management tool **aptitude**. This program makes safer decisions about package installations than running **apt-get** directly.

Don’t forget to mount all needed partitions (notably the root and `/usr` partitions) read-write, with a command like:

```
# mount -o remount,rw /mountpoint
```

Next you should double-check that the APT source entries (in `/etc/apt/sources.list`) refer either to ‘lenny’ or to ‘stable’. There should not be any sources entries pointing to `etch`.

### NOTA



Source lines for a CD-ROM will often refer to ‘unstable’; although this may be confusing, you should *not* change it.

### 4.5.1 Recording the session

It is strongly recommended that you use the `/usr/bin/script` program to record a transcript of the upgrade session. Then if a problem occurs, you will have a log of what happened, and if needed, can provide exact information in a bug report. To start the recording, type:

```
# script -t 2>~/upgrade-lenny.time -a ~/upgrade-lenny.script
```

or similar. Do not put the typescript file in a temporary directory such as `/tmp` or `/var/tmp` (files in those directories may be deleted during the upgrade or during any restart).

The typescript will also allow you to review information that has scrolled off-screen. Just switch to VT2 (using Alt+F2) and, after logging in, use `less -R ~/root/upgrade-lenny.script` to view the file.

After you have completed the upgrade, you can stop **script** by typing `exit` at the prompt.

If you have used the `-t` switch for **script** you can use the **scriptreplay** program to replay the whole session:

```
# scriptreplay ~/upgrade-lenny.time ~/upgrade-lenny.script
```

## 4.5.2 Updating the package list

First the list of available packages for the new release needs to be fetched. This is done by executing:

```
# aptitude update
```

Running this the first time new sources are updated will print out some warnings related to the availability of the sources. These warnings are harmless and will not appear if you rerun the command again.

## 4.5.3 Make sure you have sufficient space for the upgrade

You have to make sure before upgrading your system that you have sufficient hard disk space when you start the full system upgrade described in Seção 4.5.7. First, any package needed for installation that is fetched from the network is stored in `/var/cache/apt/archives` (and the `partial/` subdirectory, during download), so you must make sure you have enough space on the file system partition that holds `/var/` to temporarily download the packages that will be installed in your system. After the download, you will probably need more space in other file system partitions in order to both install upgraded packages (which might contain bigger binaries or more data) and new packages that will be pulled in for the upgrade. If your system does not have sufficient space you might end up with an incomplete upgrade that might be difficult to recover from.

Both **aptitude** and **apt** will show you detailed information of the disk space needed for the installation. Before executing the upgrade, you can see this estimate by running:

```
# aptitude -y -s -f --with-recommends dist-upgrade
[ ... ]
XXX upgraded, XXX newly installed, XXX to remove and XXX not upgraded.
Need to get xx.xMB/yyyMB of archives. After unpacking AAAMB will be used.
Would download/install/remove packages.
```

### NOTA



Running this command at the beginning of the upgrade process may give an error, for the reasons described in the next sections. In that case you will need to wait until you've done the minimal system upgrade as in Seção 4.5.6 and upgraded your kernel before running this command to estimate the disk space.

If you do not have enough space for the upgrade, make sure you free up space beforehand. You can:

- Remove packages that have been previously downloaded for installation (at `/var/cache/apt/archives`). Cleaning up the package cache by running **apt-get clean** or **aptitude clean** will remove all previously downloaded package files.
- Remove forgotten packages. If you have `popularity-contest` installed, you can use **popcon-largest-unused** to list the packages you do not use that occupy the most space. You can also use **deborphan** or **debfooster** to find obsolete packages (see Seção 4.11 ). Alternatively you can start **aptitude** in 'visual mode' and find obsolete packages under 'Obsolete and Locally Created Packages'.



- Remove packages that take up too much space and are not currently needed (you can always reinstall them after the upgrade). You can list the packages that take up the most disk space with **dpigs** (available in the `debian-goodies` package) or with **wajig** (running `wajig size`).

You can list packages that take up most of the disk space with `aptitude`. Start **aptitude** into *visual mode*, select Views → New Flat Package List (this menu entry is available only after `etch` version), press **I** and enter `~i`, press **S** and enter `~installsize`, then it will give you nice list to work with. Doing this after upgrading `aptitude` should give you access to this new feature.

- Remove translations and localization files from the system if they are not needed. You can install the `localepurge` package and configure it so that only a few selected locales are kept in the system. This will reduce the disk space consumed at `/usr/share/locale`.
- Temporarily move to another system, or permanently remove, system logs residing under `/var/log/`.
- Use a temporary `/var/cache/apt/archives`: You can use a temporary cache directory from another filesystem (USB storage device, temporary hard disk, filesystem already in use, ...)

#### NOTA



Do not use an NFS mount as the network connection could be interrupted during the upgrade.

For example, if you have a USB drive mounted on `/media/usbkey`:

1. remove the packages that have been previously downloaded for installation:

```
# apt-get clean
```

2. copy the directory `/var/cache/apt/archives` to the USB drive:

```
# cp -ax /var/cache/apt/archives /media/usbkey/
```

3. mount the temporary cache directory on the current one:

```
# mount --bind /media/usbkey/archives /var/cache/apt/archives
```

4. after the upgrade, restore the original `/var/cache/apt/archives` directory:

```
# umount /media/usbkey/archives
```

5. remove the remaining `/media/usbkey/archives`.

You can create the temporary cache directory on whatever filesystem is mounted on your system.

Note that in order to safely remove packages, it is advisable to switch your `sources.list` back to `etch` as described in Seção [A.2](#).

#### 4.5.4 Upgrade apt and/or aptitude first

Several bug reports have shown that the versions of the `aptitude` and `apt` packages in `etch` are often unable to handle the upgrade to `lenny`. In `lenny`, `apt` is better at dealing with complex chains of packages requiring immediate configuration and `aptitude` is smarter at searching for solutions to satisfy the dependencies. These two features are heavily involved during the `dist-upgrade` to `lenny`, so it is necessary to upgrade these two packages before upgrading anything else. For `apt`, run:

```
# apt-get install apt
```

and for `aptitude` (if you have it installed) run:



```
# aptitude install aptitude
```

This step will automatically upgrade `libc6` and `locales` and will pull in SELinux support libraries (`libselinux1`). At this point, some running services will be restarted, including `xdm`, `gdm` and `kdm`. As a consequence, local X11 sessions might be disconnected.

### 4.5.5 Using aptitude's list of automatically-installed packages with apt

`aptitude` maintains a list of packages that were installed automatically (for instance, as dependencies of another package). In `lenny`, `apt` now has this feature as well.

The first time the `lenny` version of `aptitude` is run, it will read in its list of automatically installed packages and convert it for use with the `lenny` version of `apt`. If you have `aptitude` installed, you should at least issue one `aptitude` command to do the conversion. One way to do this is by searching for a non-existent package:

```
# aptitude search "?false"
```

### 4.5.6 Minimal system upgrade

Because of certain necessary package conflicts between `etch` and `lenny`, running `aptitude dist-upgrade` directly will often remove large numbers of packages that you will want to keep. We therefore recommend a two-part upgrade process, first a minimal upgrade to overcome these conflicts, then a full `dist-upgrade`.

First, run:

```
# aptitude safe-upgrade
```

This has the effect of upgrading those packages which can be upgraded without requiring any other packages to be removed or installed.

The next step will vary depending on the set of packages that you have installed. These release notes give general advice about which method should be used, but if in doubt, it is recommended that you examine the package removals proposed by each method before proceeding.

Some common packages that are expected to be removed include `base-config`, `hotplug`, `xlibs`, `netkit-inetd`, `python2.3`, `xfree86-common`, and `xserver-common`. For more information about packages obsoleted in `lenny`, see Seção 4.11.

#### 4.5.6.1 Upgrading a desktop system

This step concerns systems with the `desktop` task installed, or with the GNOME or KDE packages installed.

Check whether you have the `libfam0c102` package installed:

```
# dpkg -l libfam0c102 | grep ^ii
```

If you do have it installed, run:

```
# aptitude install libfam0
```

Note that installing `libfam0` will also install the File Alteration Monitor (`fam`) as well as the RPC portmapper (`portmap`) if not already available in your system. Both packages will enable a new network service in the system although they can both be configured to be bound to the (internal) loopback network device.

#### 4.5.6.2 Upgrading a system with no X support installed

On a system with no X, no additional `aptitude install` command should be required, and you can move on to the next step.

### 4.5.7 Upgrading the rest of the system

You are now ready to continue with the main part of the upgrade. Execute:

```
# aptitude dist-upgrade
```

This will perform a complete upgrade of the system, i.e. install the newest available versions of all packages, and resolve all possible dependency changes between packages in different releases. If necessary, it will install some new packages (usually new library versions, or renamed packages), and remove any conflicting obsoleted packages.

When upgrading from a set of CD-ROMs (or DVDs), you will be asked to insert specific CDs at several points during the upgrade. You might have to insert the same CD multiple times; this is due to inter-related packages that have been spread out over the CDs.

New versions of currently installed packages that cannot be upgraded without changing the install status of another package will be left at their current version (displayed as 'held back'). This can be resolved by either using **aptitude** to choose these packages for installation or by trying `aptitude -f install package`.

### 4.5.8 Possible issues during upgrade

If an operation using **aptitude**, **apt-get**, or **dpkg** fails with the error

```
E: Dynamic MMap ran out of room
```

the default cache space is insufficient. You can solve this by either removing or commenting lines you don't need in `/etc/apt/sources.list` or increasing the cache size. The cache size can be increased by setting `APT::Cache-Limit` in `/etc/apt/apt.conf`. The following command will set it to a value that should be sufficient for the upgrade:

```
# echo 'APT::Cache-Limit "12500000";' >> /etc/apt/apt.conf
```

This assumes that you do not yet have this variable set in that file.

Sometimes it's necessary to enable the `APT::Force-LoopBreak` option in APT to be able to temporarily remove an essential package due to a Conflicts/Pre-Depends loop. **aptitude** will alert you of this and abort the upgrade. You can work around this by specifying the option `-o APT::Force-LoopBreak=1` on the **aptitude** command line.

It is possible that a system's dependency structure can be so corrupt as to require manual intervention. Usually this means using **aptitude** or

```
# dpkg --remove package_name
```

to eliminate some of the offending packages, or

```
# aptitude -f install
# dpkg --configure --pending
```

In extreme cases you might have to force re-installation with a command like

```
# dpkg --install /path/to/package_name.deb
```

File conflicts should not occur if you upgrade from a 'pure' etch system, but can occur if you have unofficial backports installed. A file conflict will result in an error like:

```
Unpacking <package-foo> (from <package-foo-file>) ...
dpkg: error processing <package-foo> (--install):
 trying to overwrite '<some-file-name>',
 which is also in package <package-bar>
dpkg-deb: subprocess paste killed by signal (Broken pipe)
Errors were encountered while processing:
<package-foo>
```

You can try to solve a file conflict by forcibly removing the package mentioned on the *last* line of the error message:

```
# dpkg -r --force-depends package_name
```

After fixing things up, you should be able to resume the upgrade by repeating the previously described **aptitude** commands.

During the upgrade, you will be asked questions regarding the configuration or re-configuration of several packages. When you are asked if any file in the `/etc/init.d` or `/etc/terminfo` directories, or the `/etc/manpath.config` file should be replaced by the package maintainer's version, it's usually necessary to answer 'yes' to ensure system consistency. You can always revert to the old versions, since they will be saved with a `.dpkg-old` extension.

If you're not sure what to do, write down the name of the package or file and sort things out at a later time. You can search in the typescript file to review the information that was on the screen during the upgrade.

## 4.6 Upgrading your kernel and related packages

This section explains how to upgrade your kernel and identifies potential issues related to this upgrade. You can either install one of the `linux-image-*` packages provided by Debian, or compile a customized kernel from source.

Note that a lot of information in this section is based on the assumption that you will be using one of the modular Debian kernels, together with `initramfs-tools` and `udev`. If you choose to use a custom kernel that does not require an `initrd` or if you use a different `initrd` generator, some of the information may not be relevant for you.

### 4.6.1 Installing the kernel metapackage

When you dist-upgrade from etch to lenny, it is strongly recommended that you install a new `linux-image-2.6-*` metapackage. This package may be installed automatically by the dist-upgrade process. You can verify this by running:

```
# dpkg -l "linux-image*" | grep ^ii
```

If you do not see any output, then you will need to install a new `linux-image` package by hand. To see a list of available `linux-image-2.6` metapackages, run:

```
# apt-cache search linux-image-2.6- | grep -v transition
```

If you are unsure about which package to select, run `uname -r` and look for a package with a similar name. For example, if you see '2.6.18-6-686', it is recommended that you install `linux-image-2.6-686`. (Note that the `k7` flavor no longer exists; if you are currently using the `k7` kernel flavor, you should install the `686` flavor instead.) You may also use `apt-cache` to see a long description of each package in order to help choose the best one available. For example:

```
# apt-cache show linux-image-2.6-686
```

You should then use `aptitude install` to install it. Once this new kernel is installed you should reboot at the next available opportunity to get the benefits provided by the new kernel version.

For the more adventurous there is an easy way to compile your own custom kernel on Debian GNU/Linux. Install the `kernel-package` tool and read the documentation in `/usr/share/doc/kernel-package`.

### 4.6.2 Upgrading from a 2.6 kernel

If you are currently running a 2.6 series kernel from etch this upgrade will take place automatically after you do a full upgrade of the system packages (as described in Seção 4.5).

If possible, it is to your advantage to upgrade the kernel package separately from the main `dist-upgrade` to reduce the chances of a temporarily non-bootable system. Note that this should only be done after the minimal upgrade process described in Seção 4.5.6.

You can also take this step if you are using your own custom kernel and want to use the kernel available in lenny. If your kernel version is not supported by `udev` then it is recommended that you upgrade after the minimal upgrade. If your version is supported by `udev` you can safely wait until after the full system upgrade.

### 4.6.3 Upgrading from a 2.4 kernel

If you have a 2.4 kernel installed, and your system relies on `hotplug` for its hardware detection you should first upgrade to a 2.6 series kernel from `etch` before attempting the upgrade. Make sure that the 2.6 series kernel boots your system and all your hardware is properly detected before you perform the upgrade. The `hotplug` package is removed from the system (in favor of `udev`) when you do a full system upgrade. If you do not do the kernel upgrade before this your system might not boot up properly from this point on. Once you have done an upgrade to a 2.6 series kernel in `etch` you can do a kernel upgrade as described in Seção 4.6.2.

If your system does not rely on `hotplug`<sup>3</sup> you can delay the kernel upgrade to after you have done a full system upgrade, as described in Seção 4.5.7. Once your system has been upgraded you can then do the following (changing the kernel package name to the one most suited to your system by substituting `flavor`):

```
# aptitude install linux-image-2.6-flavor
```

### 4.6.4 Device enumeration reordering

`lenny` features a more robust mechanism for hardware discovery than previous releases. However, this may cause changes in the order devices are discovered on your system, affecting the order in which device names are assigned. For example, if you have two network adapters that are associated with two different drivers, the devices `eth0` and `eth1` refer to may be swapped. Please note that the new mechanism means that if you e.g. exchange ethernet adapters in a running `lenny` system, the new adapter will also get a new interface name.

For network devices, you can avoid this reordering by using `udev` rules, more specifically, through the definitions at `/etc/udev/rules.d/70-persistent-net.rules`<sup>4</sup>. Alternatively you can use the `ifrename` utility to bind physical devices to specific names at boot time. See `ifrename(8)` and `iftab(5)` for more information. The two alternatives (`udev` and `ifrename`) should not be used at the same time.

For storage devices, you can avoid this reordering by using `initramfs-tools` and configuring it to load storage device driver modules in the same order they are currently loaded. To do this, identify the order the storage modules on your system were loaded by looking at the output of `lsmod`. `lsmod` lists modules in the reverse order that they were loaded in, i.e., the first module in the list was the last one loaded. Note that this will only work for devices which the kernel enumerates in a stable order (like PCI devices).

However, removing and reloading modules after initial boot will affect this order. Also, your kernel may have some drivers linked statically, and these names will not appear in the output of `lsmod`. You may be able to decipher these driver names and load order from looking at `/var/log/kern.log`, or the output of `dmesg`.

Add these module names to `/etc/initramfs-tools/modules` in the order they should be loaded at boot time. Some module names may have changed between `etch` and `lenny`. For example, `sym53c8xx_2` has become `sym53c8xx`.

You will then need to regenerate your `initramfs` image(s) by executing `update-initramfs -u -k all`.

Once you are running a `lenny` kernel and `udev`, you may reconfigure your system to access disks by an alias that is not dependent upon driver load order. These aliases reside in the `/dev/disk/` hierarchy.

### 4.6.5 Boot timing issues

If an `initrd` created with `initramfs-tools` is used to boot the system, in some cases the creation of device files by `udev` can happen too late for the boot scripts to act on.

The usual symptoms are that the boot will fail because the root file system cannot be mounted and you are dropped into a debug shell. But if you check afterwards, all devices that are needed are present in `/dev`. This has been observed in cases where the root file system is on a USB disk or on RAID, especially if LILO is used.

<sup>3</sup> You can have the kernel modules needed by your system loaded statically through proper configuration of `/etc/modules`.

<sup>4</sup> The rules there are automatically generated by the script `/etc/udev/rules.d/75-persistent-net-generator.rules` to have persistent names for network interfaces. Delete this symlink to disable persistent device naming for NICs by `udev`.

A workaround for this issue is to use the boot parameter `rootdelay=9`. The value for the timeout (in seconds) may need to be adjusted.

## 4.7 Things to do before rebooting

When `aptitude dist-upgrade` has finished, the ‘formal’ upgrade is complete, but there are some other things that should be taken care of *before* the next reboot.

### 4.7.1 Rerun lilo

If you are using `lilo` as your bootloader (it is the default bootloader for some installations of `etch`) it is strongly recommended that you rerun `lilo` after the upgrade:

```
# /sbin/lilo
```

Notice this is needed even if you did not upgrade your system’s kernel, as `lilo`’s second stage will change due to the package upgrade.

Also, review the contents of your `/etc/kernel-img.conf` and make sure that you have `do_bootloader = Yes` in it. That way the bootloader will always be rerun after a kernel upgrade.

If you encounter any issues when running `lilo`, review the symbolic links in `/` to `vmlinuz` and `initrd` and the contents of your `/etc/lilo.conf` for discrepancies.

If you forgot to rerun `lilo` before the reboot or the system is accidentally rebooted before you could do this manually, your system might fail to boot. Instead of the `lilo` prompt, you will only see `LI` when booting the system<sup>5</sup>. See Seção 4.1.3 for information on how to recover from this.

## 4.8 System boot hangs on Waiting for root file system

Procedure to recover from `/dev/hda` that became `/dev/sda` Some users have reported that an upgrade could cause the kernel not to find the system root partition after a system reboot.

In such case, the system boot will hang on the following message:

```
Waiting for root file system ...
```

and after a few seconds a bare `busybox` prompt will appear.

This problem can occur when the upgrade of the kernel introduces the use of the new generation of IDE drivers. The IDE disk naming convention for the old drivers was `hda`, `hdb`, `hdc`, `hdd`. The new drivers will name the same disks respectively `sda`, `sdb`, `sdc`, `sdd`. The problem appears when the upgrade does not generate a new `/boot/grub/menu.lst` file to take the new naming convention into account. During the boot, `Grub` will pass a system root partition to the kernel that the kernel doesn’t find.

If you have encountered this problem after upgrading, jump to Seção 4.8.2. To avoid the problem before upgrading, read ahead.

### 4.8.1 How to avoid the problem before upgrading

One can avoid this problem entirely by using an identifier for the root filesystem that does not change from one boot to the next. There are two possible methods for doing this - labeling the filesystem, or using the filesystem’s universally unique identifier (UUID). These methods are supported in Debian since the ‘`etch`’ release.

The two approaches have advantages and disadvantages. The labeling approach is more readable, but there may be problems if another filesystem on your machine has the same label. The UUID approach is uglier, but having two clashing UUIDs is highly unlikely.

For the examples below we assume the root filesystem is on `/dev/hda6`. We also assume your system has a working `udev` installation and `ext2` or `ext3` filesystems.

To implement the labeling approach:

<sup>5</sup> For more information on `lilo`’s boot error codes please see [The Linux Bootdisk HOWTO](http://tldp.org/HOWTO/Bootdisk-HOWTO/a1483.html) (<http://tldp.org/HOWTO/Bootdisk-HOWTO/a1483.html>).

1. Label the filesystem (the name must be < 16 characters) by running the command: **e2label /dev/hda6 rootfilesys**
2. Edit /boot/grub/menu.lst and change the line:

```
# kopt=root=/dev/hda6 ro
```

to

```
# kopt=root=LABEL=rootfilesys ro
```

#### NOTA



Do not remove the # at the start of the line, it needs to be there.

3. Update the kernel lines in menu.lst by running the command **update-grub**.
4. Edit /etc/fstab and change the line that mounts the / partition, e.g.:

```
/dev/hda6 / ext3 defaults,errors=remount-ro 0 1
```

to

```
LABEL=rootfilesys / ext3 defaults,errors=remount-ro 0 1
```

The change that matters here is the first column, you don't need to modify the other columns of this line.

To implement the UUID approach:

1. Find out the universally unique identifier of your filesystem by issuing: **ls -l /dev/disk/by-uuid | grep hda6**

You should get a line similar to this one:

```
lrwxrwxrwx 1 root root 24 2008-09-25 08:16 d0dfcc8a-417a-41e3-ad2e-9736317f2d8a ↔  
f2d8a -> ../../hda6
```

The UUID is the name of the symbolic link pointing to /dev/hda6 i.e.: d0dfcc8a-417a-41e3-ad2e-9736317f2d8a.

#### NOTA



Your filesystem UUID will be a different string.

2. Edit /boot/grub/menu.lst and change the line:

```
# kopt=root=/dev/hda6 ro
```

to

```
# kopt=root=UUID=d0dfcc8a-417a-41e3-ad2e-9736317f2d8 ro
```

**NOTA**

Do not remove the # at the start of the line, it needs to be there.

3. Update the kernel lines in `menu.lst` by running the command **update-grub**.

4. Edit `/etc/fstab` and change the line that mounts the `/` partition, e.g.:

```
/dev/hda6 / ext3 defaults,errors=remount-ro 0 1
```

to

```
UUID=d0dfcc8a-417a-41e3-ad2e-9736317f2d8 / ext3 defaults,errors=remount- ↵
ro 0 1
```

The change that matters here is the first column, you don't need to modify the other columns of this line.

## 4.8.2 How to recover from the problem after the upgrade

### 4.8.2.1 Solution 1

This is applicable when Grub shows you the menu interface for selecting the entry you want to boot from. If such a menu does not appear, try pressing the **Esc** key before the kernel boots in order to make it appear. If you can't get into this menu, try [Seção 4.8.2.2](#) or [Seção 4.8.2.3](#).

1. In the Grub menu, highlight the entry you want to boot from. Press the **e** key to edit the options related to this entry. You will see something like:

```
root (hd0,0)
kernel /vmlinuz-2.6.26-1-686 root=/dev/hda6 ro
initrd /initrd.img-2.6.26-1-686
```

2. Highlight the line

```
kernel /vmlinuz-2.6.26-1-686 root=/dev/hda6 ro
```

press the **e** key and replace `hdX` with `sdX` (`X` being the letter `a`, `b`, `c` or `d` depending of your system). In my example the line becomes:

```
kernel /vmlinuz-2.6.26-1-686 root=/dev/sda6 ro
```

Then press **Enter** to save the modification. If other lines show `hdX`, change these line too. Don't modify the entry similar to `root (hd0,0)`. Once all modifications are done, press the **b** key. And your system should now boot as usual.

3. Now that your system has booted, you need to fix this issue permanently. Jump to [Seção 4.8.1](#) and apply one of the two proposed procedures.

### 4.8.2.2 Solution 2

Boot from Debian GNU/Linux installation media (CD/DVD) and when prompted, pick `rescue` to launch rescue mode. Select your language, location, and keyboard mapping; then let it configure the network (no matter whether it succeeds or not). After a while, you should be asked to select the partition you want to use as root file system. The proposed choices will look something like:

```
/dev/ide/host0/bus0/target0/lun0/part1
/dev/ide/host0/bus0/target0/lun0/part2
/dev/ide/host0/bus0/target0/lun0/part5
/dev/ide/host0/bus0/target0/lun0/part6
```

If you know which partition is your root file system, choose the appropriate one. If you don't, just try with the first. If it complains about an invalid root file system partition, try the next one, and so on. Trying one after the other shouldn't harm your partitions and if you have only one operating system installed on your disks, you should easily find the right root file system partition. If you have many operating systems installed on your disks, it would be better to know exactly which is the right partition.

Once you have chosen a partition, you will be offered a range of options. Pick the option of executing a shell in the selected partition. If it complains that it cannot do that then try with another partition.

Now you should have shell access as user `root` on your root file system mounted on `/target`. You need access to the contents of the `/boot`, `/sbin` and `/usr` directories on your hard disk, which should now be available under `/target/boot`, `/target/sbin` and `/target/usr`. If these directories need to be mounted from other partitions, do so (see `/etc/fstab` if you have no idea of which partition to mount).

Jump to Seção 4.8.1 and apply one of the two proposed procedures to fix the problem permanently. Then type `exit` to leave the rescue shell and select `reboot` for rebooting the system as usual (don't forget to remove the bootable media).

#### 4.8.2.3 Solution 3

1. Boot from your favorite LiveCD distribution, such as Debian Live, Knoppix, or Ubuntu Live.
2. Mount the partition where your `/boot` directory is. If you don't know which one it is, use the output of the command `dmmsg` to find whether your disk is known as `hda`, `hdb`, `hdc`, `hdd` or `sda`, `sdb`, `sdc`, `sdd`. Once you know which disk to work on, for example `sdb`, issue the following command to see the partition table of the disk and to find the right partition: `fdisk -l /dev/sdb`
3. Assuming that you have mounted the right partition under `/mnt` and that this partition contains the `/boot` directory and its content, edit the `/mnt/boot/grub/menu.lst` file.

Find the section similar to:

```
## ## End Default Options ##

title          Debian GNU/Linux, kernel 2.6.26-1-686
root           (hd0,0)
kernel         /vmlinuz-2.6.26-1-686 root=/dev/hda6 ro
initrd        /initrd.img-2.6.26-1-686

title          Debian GNU/Linux, kernel 2.6.26-1-686 (single-user mode)
root           (hd0,0)
kernel         /vmlinuz-2.6.26-1-686 root=/dev/hda6 ro single
initrd        /initrd.img-2.6.26-1-686

### END DEBIAN AUTOMAGIC KERNELS LIST
```

and replace every `hda`, `hdb`, `hdc`, `hdd` with `sda`, `sdb`, `sdc`, `sdd`, as appropriate. Don't modify the line similar to:

```
root           (hd0,0)
```

4. Reboot the system, remove the LiveCD and your system should boot correctly.
5. When it has booted, apply one of the two proposed procedures under Seção 4.8.1 to fix the problem permanently.



## 4.9 Preparing for the next release

After the upgrade there are several things you can do to prepare for the next release.

- If the new kernel image metapackage was pulled in as a dependency of the old one, it will be marked as automatically installed, which should be corrected:

```
# aptitude unmarkauto $(dpkg-query -W 'linux-image-2.6-*' | cut -f1)
```

- Remove obsolete and unused packages as described in Seção 4.11. You should review which configuration files they use and consider purging the packages to remove their configuration files.

## 4.10 Deprecated packages

With the next release of Debian GNU/Linux 6.0 (codenamed squeeze) a bigger number of server packages will be deprecated, thus updating to newer versions of those now will save you from trouble when updating to 6.0.

This includes the following packages:

- `apache (1.x)`, successor is `apache2`
- `bind8`, successor is `bind9`
- `exim3`, successor is `exim4`
- `php4`, successor is `php5`
- `postgresql-7.4`, successor is `postgresql-8.1`
- `uw-imapd`, no successor defined yet

## 4.11 Obsolete packages

Introducing several thousand new packages, lenny also retires and omits more than two thousand old packages that were in etch. It provides no upgrade path for these obsolete packages. While nothing prevents you from continuing to use an obsolete package where desired, the Debian project will usually discontinue security support for it a year after lenny's release<sup>6</sup>, and will not normally provide other support in the meantime. Replacing them with available alternatives, if any, is recommended.

There are many reasons why packages might have been removed from the distribution: they are no longer maintained upstream; there is no longer a Debian Developer interested in maintaining the packages; the functionality they provide has been superseded by different software (or a new version); or they are no longer considered suitable for lenny due to bugs in them. In the latter case, packages might still be present in the 'unstable' distribution.

Detecting which packages in an updated system are 'obsolete' is easy since the package management front-ends will mark them as such. If you are using **aptitude**, you will see a listing of these packages in the 'Obsolete and Locally Created Packages' entry. **dselect** provides a similar section but the listing it presents might differ.

Also, if you have used **aptitude** to manually install packages in etch it will have kept track of those packages you manually installed and will be able to mark as obsolete those packages pulled in by dependencies alone which are no longer needed if a package has been removed. Also, **aptitude**, unlike **deborphan** will not mark as obsolete packages that you manually installed, as opposed to those that were automatically installed through dependencies.

There are additional tools you can use to find obsolete packages such as **deborphan**, **debfooster** or **cruft**. **deborphan** is highly recommended, although it will (in default mode) only report obsolete libraries: packages in the 'libs' or 'oldlibs' sections that are not used by any other packages. Do not blindly remove the packages these tools present, especially if you are using aggressive non-default options that are prone to produce false positives. It is highly recommended that you manually review the packages suggested for removal (i.e. their contents, size and description) before you remove them.

<sup>6</sup> Or for as long as there is not another release in that time frame. Typically only two stable releases are supported at any given time.

The [Debian Bug Tracking System](http://bugs.debian.org/) (<http://bugs.debian.org/>) often provides additional information on why the package was removed. You should review both the archived bug reports for the package itself and the archived bug reports for the [ftp.debian.org pseudo-package](http://bugs.debian.org/cgi-bin/pkgreport.cgi?pkg=ftp.debian.org&archive=yes) (<http://bugs.debian.org/cgi-bin/pkgreport.cgi?pkg=ftp.debian.org&archive=yes>).

### 4.11.1 Dummy packages

Some packages from etch have been split into several packages in lenny, often to improve system maintainability. To ease the upgrade path in such cases, lenny often provides ‘dummy’ packages: empty packages that have the same name as the old package in etch with dependencies that cause the new packages to be installed. These ‘dummy’ packages are considered obsolete packages after the upgrade and can be safely removed.

Most (but not all) dummy packages’ descriptions indicate their purpose. Package descriptions for dummy packages are not uniform, however, so you might also find **deborphan** with the `--guess` options useful to detect them in your system. Note that some dummy packages are not intended to be removed after an upgrade but are, instead, used to keep track of the current available version of a program over time.

## Capítulo 5

# Problemas a serem considerados para o lenny

### 5.1 Problemas potenciais

Algumas vezes, mudanças tem efeitos colaterais que não podem ser evitados ou que acabam expondo bugs em outros locais. Aqui, nós documentamos problemas conhecidos. Por favor, leia a errata e a documentação dos pacotes, relatórios de bugs e outras informações mencionadas em Seção 6.1.

#### 5.1.1 Problemas com dispositivos relacionados ao udev

Embora o `udev` tenha sido extensivamente testado, você pode encontrar pequenos problemas com alguns dispositivos que precisarão ser corrigidos. Os problemas mais comuns são mudança de permissões e/ou dono de um dispositivo. Em alguns casos um dispositivo não pode ser criado por padrão (e.g. `/dev/video` e `/dev/radio`).

O `udev` fornece mecanismos de configuração para lidar com estes problemas. Veja `udev(8)` e `/etc/udev` para mais informações.

#### 5.1.2 Algumas aplicações deixaram de funcionar com um kernel 2.4

Algumas aplicações no lenny podem não funcionar com um kernel 2.4, por exemplo, por necessitarem de suporte a `epoll()`, que não está disponível nos kernels 2.4. Essas aplicações podem não funcionar por completo ou não funcionar corretamente até que o sistema seja reinicializado com um kernel 2.6.

Um exemplo é o proxy HTTP `squid`.

#### 5.1.3 Alguns locais de rede não podem ser alcançados pelo TCP

Desde a versão 2.6.17, o Linux usa agressivamente redimensionamento de janela ("window scaling") que é especificado na RFC 1323. Alguns servidores têm um comportamento falho e anunciam o tamanho incorreto das janelas para si próprios. Para mais detalhes, por favor, veja os relatórios de bug [#381262](http://bugs.debian.org/381262) (<http://bugs.debian.org/381262>), [#395066](http://bugs.debian.org/395066) (<http://bugs.debian.org/395066>), [#401435](http://bugs.debian.org/401435) (<http://bugs.debian.org/401435>).

Usualmente, há duas formas de contornar estes problemas: uma é reverter os valores máximos dos tamanhos de janela TCP para valores menores (preferível) e a outra é desligar completamente o redimensionamento de janelas TCP (obsoleto). Veja os comandos de exemplo na [página de errata do debian-installer](http://www.debian.org/devel/debian-installer/errata) (<http://www.debian.org/devel/debian-installer/errata>).

#### 5.1.4 Desligamento automático deixa de funcionar

Em alguns sistemas antigos, `shutdown -h` pode não desligar o sistema (somente pará-lo). Isto acontece porque o APM precisa ser usado. Adicionar `acpi=off apm=power_off` à linha de comando do kernel, e.g. nos arquivos de configuração do `grub` ou do `lilo` deveria corrigir o problema. Por favor, veja o bug [#390547](http://bugs.debian.org/390547) (<http://bugs.debian.org/390547>) para informações adicionais.

### 5.1.5 Inicialização de rede assíncrona pode causar comportamento imprevisível

Em sistemas que usam `udev` para carregar dispositivos para interfaces de rede, é possível que dada a natureza assíncrona do `udev` que o driver de rede não seja carregado antes da execução do `/etc/init.d/networking` na inicialização do sistema. Embora a inclusão de `allow-hotplug` no `/etc/network/interfaces` (além do `auto`) garantirá que a interface de rede seja habilitada assim que ficar disponível, não há garantia que isto será terminado antes da sequência de inicialização dar inícios aos serviços de rede, os quais podem não se comportar corretamente na ausência de uma interface de rede.

### 5.1.6 Problema quando usando redes sem-fio protegidas com WPA

No `etch`, o pacote `wpa_supplicant` era instalado como um serviço do sistema, configurado via `/etc/default/wpa_supplicant` e do arquivo `/etc/wpa_supplicant.conf` fornecido pelo usuário.

No `lenny`, o `/etc/init.d/wpa_supplicant` foi descontinuado e o pacote Debian agora é integrado ao `/etc/network/interfaces`, de maneira similar a outros pacote como `wireless-tools`. Isto significa que o `wpa_supplicant` deixou de fornecer diretamente um serviço de sistema.

Para informação sobre a configuração do `wpa_supplicant`, por favor, consulte o `/usr/share/doc/wpa_supplicant/README.modes.gz`, que dá exemplos para o arquivo `/etc/network/interfaces`. Informação atualizada sobre o uso do pacote `wpa_supplicant` no Debian pode ser encontrada no [Debian Wiki](http://wiki.debian.org/WPA) (<http://wiki.debian.org/WPA>).

### 5.1.7 Problemas com caracteres não-ASCII em nomes de arquivo

Montar sistemas de arquivos `vfat`, `ntfs` ou `iso9660` com arquivos que incluem caracteres não-ASCII em seus nomes resultará em falhas quando alguém tentar usar os nomes de arquivos e a menos que a montagem seja feita com a opção `utf8`. Um indicativo pode ser a seguinte falha: 'Invalid or incomplete multibyte or wide character' ('Caractere duplo ou multibyte inválido ou incompleto'). Uma possível solução é usar `defaults,utf8` como opções de montagem para sistemas de arquivos `vfat`, `ntfs` e `iso9660` quando eles contêm nomes de arquivos com caracteres não-ASCII.

Note que o kernel Linux não tem suporte a manipulação de nomes de arquivos ignorando maiúsculas e minúsculas para `vfat` quando a opção `utf8` é usada.

### 5.1.8 Som deixa de funcionar

Em casos raros, o som pode parar de funcionar após a atualização. Se isto acontecer, repasse a lista de checagem do ALSA:

- execute o `alsacnf` como `root`,
- adicione seu usuário ao grupo `audio`,
- certifique-se de que os níveis dos canais de som estão ativos e não-mudos (usando o `alsamixer`),
- certifique-se de que o `arts` e o `esound` não estão sendo executados,
- certifique-se de que nenhum módulo OSS foi carregado,
- certifique-se de que os alto falantes estão ligados, e
- verifique se o comando

```
cat /dev/urandom > /dev/dsp
```

funciona para o `root`.

## 5.2 Montagens de NFS agora controladas pelo nfs-common

Desde a versão 2.13 do `util-linux` montagens NFS não são mais controladas pelo `util-linux`, mas pelo `nfs-common`. Como nem todos os sistemas montam compartilhamentos NFS e para evitar uma instalação padrão do `portmapper`, o `util-linux` somente sugere o `nfs-common`. Se você precisa montar compartilhamentos NFS, certifique-se de que o pacote `nfs-common` está instalado em seu sistema. O script de pré-instalação do pacote `mount` verifica se há montagens NFS e aborta se o `/usr/sbin/mount.nfs` do `nfs-common` não está presente ou se o `nfs-common` está desatualizado. Atualize o `nfs-common` ou desmonte quaisquer pontos de montagem NFS antes de atualizar o `mount`.

## 5.3 Mudança do layout de teclado romeno (ro)

Por causa da atualização do `xkb-data` versão 1.3 no `lenny` a variante padrão do layout romeno (`ro`) agora produz os caracteres `șț` corretamente (vírgula sob o caractere) ao invés de `şţ` (cedilha sob o caractere). Além disso, algumas das variantes foram renomeadas. As antigos nomes das variantes ainda funcionam, mas os usuários são encorajados a atualizar seus arquivos `/etc/X11/xorg.conf`. Mais informações bem como todos os efeitos colaterais possíveis resultantes desta mudança estão disponíveis no [wiki \(somente no idioma romeno\)](http://wiki.debian.org/L10N/Romanian/Lenny/Notes) (<http://wiki.debian.org/L10N/Romanian/Lenny/Notes>).

## 5.4 Atualizando o apache2

A configuração padrão do `apache2` mudou e em alguns casos é necessária uma intervenção manual para realizar mudanças na sua configuração. As alterações mais importantes são:

`NameVirtualHost *` foi mudado para `NameVirtualHost *:80`. Se você adicionou mais hosts virtuais baseados em nome, você precisa mudar o `<VirtualHost *>` para `<VirtualHost *:80>` em cada um deles.

As opções `User`, `Group` e `PidFile` do Apache agora são configuradas no arquivo `/etc/apache2/envvars`. Se você mudou essas opções para algo diferente do valor padrão, você terá que mudar esse arquivo. Isto também significa que iniciar o `apache2` com `apache2 -k start` não é mais possível, você terá que usar `/etc/init.d/apache2` ou `apache2ctl`.

O programa auxiliar `suexec` precisa do `mod_suexec` que agora é distribuído em um pacote separado `apache2-suexec`, que não é instalado por padrão.

Mais configurações específicas de módulos foram movidas de `/etc/apache2/apache2.conf` para `/etc/apache2/mods-available/*.conf`.

Para mais informações detalhadas, veja `/usr/share/doc/apache2.2-common/NEWS.Debian.gz` e `/usr/share/doc/apache2.2-common/README.Debian.gz`.

## 5.5 NIS e Network Manager

A versão do `ypbind` incluída no `nis` para o `lenny` contém suporte ao Network Manager. Este recurso faz com que o `ypbind` desabilite a funcionalidade de cliente NIS quando o Network Manager reporta que o computador está desconectado da rede. Como o Network Manager geralmente reportará que o computador foi desconectado quando ele não está sendo usado, usuários NIS com sistemas cliente NIS deverão certificar-se de que o suporte ao Network Manager esteja desabilitado nesses sistemas.

Isto pode ser feito desinstalando o pacote `network-manager` ou editando o arquivo `/etc/default/nis` para adicionar a opção `-no-dbus` em `YPBINDARGS`.

O uso da opção `-no-dbus` é o padrão para novas instalações do Debian, mas não era o padrão em versões anteriores.

## 5.6 Aspectos de segurança dos produtos Mozilla

Os programas Mozilla `firefox`, `thunderbird`, e `sunbird` (renomeados no Debian como `iceweasel`, `icedove` e `iceowl`, respectivamente), são ferramentas importantes para vários usuários. Infelizmente, a política de segurança dos desenvolvedores desses produtos é pedir aos usuários que atualizem para novas versões, o que conflita com a política do Debian de não distribuir grandes mudanças funcionais em atualizações de segurança. Nós não podemos prever hoje, mas durante o ciclo de vida

do lenny o Time de Segurança do Debian pode chegar a um ponto onde dar suporte a produtos Mozilla não seja possível e anuncie o fim do suporte de segurança aos produtos Mozilla. Você deve considerar estes pontos se for usar Mozilla e considerar alternativas disponíveis no Debian se a ausência de suporte de segurança for um problema pra você.

`iceape`, a versão renomeada da suíte `internet seamonkey` foi removida do lenny (exceto alguns pacotes de bibliotecas internas).

## 5.7 Suporte e mudanças na área de trabalho GNOME

There have been many changes in the GNOME desktop environment from the version shipped in etch to the version in lenny, you can find more information in the [GNOME 2.22 Release Notes](http://library.gnome.org/misc/release-notes/2.22/) (<http://library.gnome.org/misc/release-notes/2.22/>).

## 5.8 Não há suporte padrão para Unicode no emacs21\*

Emacs21 e emacs21-nox não são configurados para usar Unicode por padrão. Para mais informações e formas de contornar o problemas, por favor, veja o [Bug #419490](http://bugs.debian.org/419490) (<http://bugs.debian.org/419490>). Considere a opção de mudar para o emacs22, emacs22-gtk, ou emacs22-nox.

## 5.9 slurpd/replica deixarão de funcionar

O OpenLDAP abandonou o suporte à replicação LDAP através do serviço slurpd na versão 2.4.7. Configurações existentes precisam ser reconfigurados para o mecanismo LDAP Sync Replication (`syncrepl`). Documentação mais detalhada pode ser encontrada em <http://www.openldap.org/doc/admin24/-replication.html> (<http://www.openldap.org/doc/admin24/replication.html>).

## 5.10 Área de trabalho não usa toda a tela

O driver para Intel Mobile GM965 pode detectar incorretamente uma saída VGA e definir o tamanho da tela para um valor menor para acomodá-la. O sintoma deste bug é que o gerenciador de área de trabalho usará somente uma fração da tela. O comportamento correto pode ser forçado adicionado as seguintes linhas no arquivo de configuração `/etc/X11/xorg.conf`.

```
Section "Monitor"
  Identifier "VGA"
  Option "Ignore" "true"
EndSection
```

Por favor, veja o bug [#496169](http://bugs.debian.org/496169) (<http://bugs.debian.org/496169>) para mais informações.

## 5.11 Problema com DHCP 'failover'

Quando usando um par "failover" de servidores DHCP, os nomes dos pares precisam ser consistentes, caso contrário o DHCP falhará (crash).

Por favor, veja <http://bugs.debian.org/513506> e <https://lists.isc.org/pipermail/dhcp-users/2007-September/004538.html> para mais informações.

## Capítulo 6

# Mais informações sobre o Debian GNU/Linux

### 6.1 Leitura complementar

Além destas notas de lançamento e do guia de instalação, mais documentação sobre o Debian GNU/Linux está disponível a partir do Projeto de Documentação do Debian (DDP -- Debian Documentation Project), cujo objetivo é criar documentação de alta qualidade para usuários e desenvolvedores Debian. Documentação, incluindo a Referência Debian, o Guia de Novos Mantenedores Debian e o Debian FAQ estão disponíveis, além de muitos outros. Para todos os detalhes dos recursos existentes veja o [site web do DDP](http://www.debian.org/doc/ddp) (<http://www.debian.org/doc/ddp>).

Documentação para pacotes individuais é instalada em `/usr/share/doc/pacote`. Isto pode incluir informação de copyright, detalhes específicos do Debian e documentação do autor do software.

### 6.2 Obtendo ajuda

Há várias fontes de ajuda, aconselhamento e suporte para usuários Debian, mas estas só deveriam ser consideradas se a pesquisa na documentação pelo problema exauriu todos os recursos. Esta seção fornece uma pequena introdução aos recursos que podem ser úteis para novos usuários Debian.

#### 6.2.1 Listas de discussão

As listas de discussão de maior interesse para usuários Debian são as listas `debian-user` (inglês) e outras listas `debian-user-idioma` (para outros idiomas). Para informações sobre estas listas e detalhes sobre como se inscrever veja <http://lists.debian.org/>. Por favor, verifique antes nos arquivos se já existem respostas para suas perguntas antes de enviar algo e também respeite a etiqueta padrão para listas.

#### 6.2.2 Internet Relay Chat

O Debian possui um canal IRC dedicado para o suporte a ajuda de usuários Debian localizado na rede de IRC OFTC. Para acessar o canal, aponte seu cliente de IRC favorito para `irc.debian.org` e entre no canal `#debian` (inglês). Também é possível usar o canal `#debian-br` para obter suporte em português do Brasil.

Please follow the channel guidelines, respecting other users fully. The guidelines are available at the [Debian Wiki](http://wiki.debian.org/DebianIRC) (<http://wiki.debian.org/DebianIRC>).

For more information on OFTC please visit the [website](http://www.oftc.net/) (<http://www.oftc.net/>).

### 6.3 Reporting bugs

We strive to make Debian GNU/Linux a high quality operating system, however that does not mean that the packages we provide are totally free of bugs. Consistent with Debian's 'open development'



philosophy and as a service to our users, we provide all the information on reported bugs at our own Bug Tracking System (BTS). The BTS is browseable at <http://bugs.debian.org/>.

If you find a bug in the distribution or in packaged software that is part of it, please report it so that it can be properly fixed for future releases. Reporting bugs requires a valid email address. We ask for this so that we can trace bugs and developers can get in contact with submitters should additional information be needed.

You can submit a bug report using the program **reportbug** or manually using email. You can read more about the Bug Tracking System and how to use it by reading the reference documentation (available at `/usr/share/doc/debian` if you have `doc-debian` installed) or online at the **Bug Tracking System** (<http://bugs.debian.org/>).

## 6.4 Contributing to Debian

You do not need to be an expert to contribute to Debian. By assisting users with problems on the various user support **lists** (<http://lists.debian.org/>) you are contributing to the community. Identifying (and also solving) problems related to the development of the distribution by participating on the development **lists** (<http://lists.debian.org/>) is also extremely helpful. To maintain Debian's high quality distribution, **submit bugs** (<http://bugs.debian.org/>) and help developers track them down and fix them. If you have a way with words then you may want to contribute more actively by helping to write **documentation** (<http://www.debian.org/doc/ddp>) or **translate** (<http://www.debian.org/international/>) existing documentation into your own language.

If you can dedicate more time, you could manage a piece of the Free Software collection within Debian. Especially helpful is if people adopt or maintain items that people have requested for inclusion within Debian. The **Work Needing and Prospective Packages database** (<http://www.debian.org/devel/wnpp/>) details this information. If you have an interest in specific groups then you may find enjoyment in contributing to some of Debian's subprojects which include ports to particular architectures, **Debian Jr.** (<http://www.debian.org/devel/debian-jr/>) and **Debian Med** (<http://www.debian.org/devel/debian-med/>).

In any case, if you are working in the free software community in any way, as a user, programmer, writer or translator you are already helping the free software effort. Contributing is rewarding and fun, and as well as allowing you to meet new people it gives you that warm fuzzy feeling inside.



# Apêndice A

## Gerenciado seu sistema etch

Este apêndice contém informação sobre como assegurar-se de que você consegue instalar ou atualizar pacotes do etch antes de atualizar para o lenny. Isto só será necessário em situações específicas.

### A.1 Atualizando seu sistema etch

Basicamente, isto não é diferente de qualquer outras atualização do etch que você tem feito. A única diferença é que você precisa ter certeza de que sua lista de pacotes ainda contém referências para o etch conforme explicado em Seção [A.2](#).

Se você atualizar o seu sistema usando um espelho Debian, ele automaticamente atualizará para a última versão pontual do etch.

### A.2 Verificando sua lista de fontes (sources list)

Se qualquer uma das linhas no seu arquivo `/etc/apt/sources.list` fizer referência a 'stable', você já está efetivamente 'usando' o lenny. Se você já executou `apt-get update`, você ainda pode voltar atrás sem problemas seguindo o procedimento abaixo.

Se você também já instalou pacotes do lenny, provavelmente não há razão para instalar pacotes do etch. Neste caso, você terá que decidir se quer continuar ou não. É possível rebaixar ('downgrade') um pacote, mas isto não é abordado neste documento.

Abra o arquivo `/etc/apt/sources.list` com seu editor favorito (como `root`) e verifique todas as linhas começando com `deb http:` ou `deb ftp:` para determinar se existe uma referência a 'stable'. Se você encontrar qualquer uma, mude o `stable` para `etch`.

Se você possui linhas começando como `deb file:`, você mesmo terá que verificar se a localização indicada contém um repositório do etch ou do lenny.

#### IMPORTANTE



Não mude linhas que começam com `deb cdrom:`. Fazer isso invalidaria a linha e você teria que executar o **apt-cdrom** novamente. Não se assuste se uma linha de código-fonte 'cdrom' apontar para 'unstable'. Embora confuso, isto é normal.

Se você fizer quaisquer mudanças, salve o arquivo e execute

```
# apt-get update
```

para atualizar a lista de pacotes.



## Apêndice B

# Colaboradores das notas de lançamento

Várias pessoas ajudaram com as notas de lançamento, incluindo, mas não se limitando a

Adam Di Carlo, Andreas Barth, Andrei Popescu, Anne Bezemer, Bob Hilliard, Charles Plessy, Christian Perrier, Daniel Baumann, Eddy Petrișor, Emmanuel Kasper, Esko Arajärvi Frans Pop, Giovanni Rapagnani, Gordon Farquharson, Javier Fernández-Sanguino Peña, Jens Seidel, Jonas Meurer, Josip Rodin, Justin B Rye, LaMont Jones, Luk Claes, Martin Michlmayr, Michael Biebl, Moritz Mühlenhoff, Noah Meyerhans, Noritada Kobayashi, Osamu Aoki, Peter Green, Rob Bradford, Samuel Thibault, Simon Bilenlein, Simon Paillard, Stefan Fritsch, Steve Langasek, Tobias Scherer, Vincent McIntyre, e W. Martin Borgert.

Este documento foi traduzido em vários idiomas. Muito obrigado aos tradutores!

Traduzido para português do Brasil por: Felipe Augusto van de Wiel



## Apêndice C

# Lenny dedicado a Thiemo Seufer

O Projeto Debian perdeu um membro ativo de sua comunidade. Thiemo Seufer morreu em 26 de dezembro de 2008 em um trágico acidente de carro.

Thiemo estava envolvido com o Debian de várias formas. Ele mantinha vários pacotes e era o principal apoiador dos portes Debian para a arquitetura MIPS. Ele também foi um membro de nosso time de kernel, bem como um membro do time do Instalador Debian. Suas contribuições foram muito além do projeto Debian: ele também trabalhou no porte MIPS do kernel Linux, na emulação MIPS do qemu e muitos outros pequenos projetos para serem nomeados aqui.

Sentiremos falta da habilidade de compartilhar com outros, do vasto conhecimento técnico, da dedicação e do trabalho de Thiemo. Suas contribuições não serão esquecidas. O alto padrão do trabalho de Thiemo torna difícil de igualá-lo.

Para honrar suas contribuições com o Debian, o projeto dedica o lançamento do Debian GNU/Linux 5.0 'Lenny' a Thiemo.



# Apêndice D

## Glossário

### **ACPI**

Advanced Configuration and Power Interface

### **ALSA**

Advanced Linux Sound Architecture

### **APM**

Advanced Power Management

### **CD**

Compact Disc

### **CD-ROM**

Compact Disc Read Only Memory

### **DHCP**

Dynamic Host Configuration Protocol

### **DNS**

Domain Name System

### **DVD**

Digital Versatile Disc

### **GIMP**

GNU Image Manipulation Program

### **GNU**

GNU's Not Unix

### **GPG**

GNU Privacy Guard

### **IDE**

Integrated Drive Electronics

### **LDAP**

Lightweight Directory Access Protocol

### **LILO**

Linux LOader

### **LSB**

Linux Standards Base

### **LVM**

Logical Volume Manager

**MTA**

Mail Transport Agent

**NFS**

Network File System

**NIC**

Network Interface Card

**NIS**

Network Information Service

**OSS**

Open Sound System

**RAID**

Redundant Array of Independent Disks

**RPC**

Remote Procedure Call

**SATA**

Serial Advanced Technology Attachment

**USB**

Universal Serial Bus

**UUID**

Universally Unique Identifier

**VGA**

Video Graphics Array

**WPA**

Wi-Fi Protected Access



# Índice Remissivo

## A

Abiword, 5  
Apache, 6

## B

BIND, 6  
Blu-ray, 6

## C

CD, 6  
Cherokee, 6  
Courier, 6

## D

Dia, 6  
DocBook XML, 4  
DVD, 6

## E

Ekiga, 6  
Emdebian, 8  
Evolution, 6  
Exim, 6

## F

Firefox, 6

## G

Gaim, 6  
GCC, 6  
GIMP, 6  
GNOME, 5  
GNUCash, 5  
GNUmeric, 5

## I

IcedTea, 8

## J

Java, 8

## K

KDE, 5  
KOffice, 5

## L

LILO, 24  
Linux Standards Base, 6  
LXDE, 5

## M

Microsoft Windows, 9  
Mozilla, 6, 33  
MySQL, 6

## N

Network Manager, 33  
NIS, 33

## O

OpenJDK, 8  
OpenOffice.org, 5  
OpenSSH, 6  
OpenVZ, 8

## P

packages  
  apache, 29  
  apache2, 29  
  apache2-suexec, 33  
  apt, 4, 17, 19–21  
  aptitude, 6, 20, 21  
  base-config, 21  
  bind8, 29  
  bind9, 29  
  dblatex, 4  
  debian-goodies, 20  
  doc-debian, 36  
  docbook-xsl, 4  
  eeepc-acpi-scripts, 8  
  emacs22, 34  
  emacs22-gtk, 34  
  emacs22-nox, 34  
  exim3, 29  
  exim4, 29  
  fam, 21  
  firefox, 33  
  glibc, 13  
  grub, 31  
  hotplug, 21, 24  
  iceape, 34  
  icedove, 6, 33  
  iceowl, 33  
  iceweasel, 6, 33  
  initramfs-tools, 14, 23, 24  
  kernel-package, 23  
  libc6, 21  
  libfam0, 21  
  libfam0c102, 21  
  libselinux1, 21  
  lilo, 25, 31  
  linux-image-\*, 23  
  linux-image-2.6-686, 23  
  localepurge, 20  
  locales, 21  
  lxde, 8  
  mount, 33  
  netkit-inetd, 21  
  network-manager, 33  
  nfs-common, 33  
  nis, 33  
  php4, 29  
  php5, 29  
  popularity-contest, 19  
  portmap, 21

postgresql-7.4, 29  
postgresql-8.1, 29  
python2.3, 21  
release-notes, 3  
rsyslog, 7  
seamonkey, 34  
squid, 31  
sunbird, 33  
sysklogd, 7  
thunderbird, 33  
udev, 23, 24, 31, 32  
upgrade-reports, 3  
util-linux, 33  
uw-imapd, 29  
wireless-tools, 32  
wpa\_supplicant, 32  
xfree86-common, 21  
xkb-data, 33  
xlibs, 21  
xmlroff, 4  
xserver-common, 21  
xsltproc, 4

PHP, 6  
Pidgin, 6  
Postfix, 6  
PostgreSQL, 6

## S

SELinux, 7, 21

## T

Thunderbird, 6  
Tomcat, 6

## U

Unicode, 34  
usuários com necessidades visuais especiais, 10

## V

virtualization, 8  
VServer, 8

## W

WPA, 32

## X

Xfce, 5