

# Como usar o APT

Gustavo Noronha Silva <kov@debian.org>  
Tradução de Tiago Bortoletto Vaz <tiago@debian-ba.org>

1.9.3 - Novembro de 2004

## Resumo

Este documento pretende levar ao usuário um bom conhecimento das funções do utilitário de empacotamento do Debian, APT. É objetivo real deste facilitar a vida de novos usuários de Debian e ajudar aqueles que desejam se aprofundar no conhecimento da administração desse sistema. Ele foi criado para o projeto Debian para melhorar ainda mais o suporte disponível aos usuários desta distribuição.

## **Nota de Copyright**

Copyright © 2001, 2002, 2003, 2004 Gustavo Noronha Silva

Este manual é software livre; você pode redistribuí-lo e/ou modificá-lo sob os termos da Licença Pública Geral GNU conforme publicada pela Free Software Foundation; tanto a versão 2 da Licença, como (a seu critério) qualquer versão posterior.

Este programa é distribuído na expectativa de que seja útil, porém, sem nenhuma garantia; nem mesmo a garantia implícita de comerciabilidade ou adequação a uma finalidade específica. Consulte a Licença Pública Geral GNU para mais detalhes.

Uma cópia da GNU General Public License está disponível como `/usr/share/common-licenses/GPL` na distribuição Debian GNU/Linux ou na World Wide Web como Licença Pública Geral GNU. Você pode também obtê-la escrevendo para a Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

---

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Terminologia e conceitos básicos . . . . .	1
<b>2</b>	<b>Configuração básica</b>	<b>3</b>
2.1	O arquivo <code>/etc/apt/sources.list</code> . . . . .	3
2.2	Adicionando um CD-ROM no arquivo <code>sources.list</code> . . . . .	4
2.3	O arquivo de configuração do APT . . . . .	5
2.4	Usando o APT através de um proxy . . . . .	5
<b>3</b>	<b>Gerenciando pacotes</b>	<b>7</b>
3.1	Usando o Gerenciador de Pacote Debian: <code>aptitude</code> . . . . .	7
3.2	Atualizando a lista de pacotes disponíveis . . . . .	9
3.3	Fazendo buscas por pacotes . . . . .	9
3.4	Instalando e reinstalando pacotes . . . . .	10
3.5	Instalado manualmente pacotes debian que foram baixados ou criados localmente	12
3.6	Removendo pacotes . . . . .	13
3.7	Atualizando pacotes . . . . .	14
3.8	Atualizando para uma nova release . . . . .	14
3.9	Mantendo pacotes . . . . .	15
3.10	Removendo arquivos de pacotes não utilizados: <code>clean and autoclean</code> . . . . .	15
<b>4</b>	<b>Construindo a partir dos fontes</b>	<b>17</b>
4.1	Fazendo o download de pacotes-fonte . . . . .	17
4.2	Pacotes necessários para compilar pacotes-fonte . . . . .	18
4.3	Construindo um pacote de depuração . . . . .	19

---

4.4	Opções para construção de pacotes personalizados . . . . .	19
4.5	Mas ei, eu não quero usar essas funcionalidades do Debian! . . . . .	20
<b>5</b>	<b>Obtendo informações sobre pacotes.</b>	<b>23</b>
5.1	Descobrimo nome de pacotes . . . . .	23
5.2	Usando o dpkg para encontrar nomes de pacotes . . . . .	26
5.3	Como descobrir a que pacote um arquivo pertence . . . . .	26
5.4	Informação sobre um pacote na WEB . . . . .	27
<b>6</b>	<b>Truques e técnicas</b>	<b>29</b>
6.1	Como instalar pacotes “sob demanda” . . . . .	29
6.2	Fazendo cache de pacotes . . . . .	29
6.3	Decidindo qual é o melhor espelho para ser incluído no arquivo sources.list: netselect, netselect-apt . . . . .	30
6.4	Removendo arquivos de localização (locale) não utilizados: localepurge . . . . .	32
6.5	Como ficar informado sobre as modificações nos pacotes. . . . .	32
<b>7</b>	<b>Estando no outro lado: criando um repositório</b>	<b>33</b>
<b>8</b>	<b>Interfaces alternativas para usuários gerenciarem pacotes</b>	<b>35</b>
8.1	Synaptic . . . . .	35
8.2	GNOME-APT . . . . .	35
8.3	KPackage FIXME!! . . . . .	36
<b>9</b>	<b>Como lidar com erros</b>	<b>37</b>
9.1	Erros comuns . . . . .	37
9.2	Onde posso encontrar ajuda? . . . . .	38
<b>10</b>	<b>Quais distribuições suportam APT?</b>	<b>39</b>
<b>11</b>	<b>Agradecimentos</b>	<b>41</b>
<b>12</b>	<b>Novas versões deste tutorial</b>	<b>43</b>

# Capítulo 1

## Introdução

No início havia o `.tar.gz`. Os usuários tinham que compilar cada programa que eles quisessem usar em seus sistemas GNU/Linux. Quando o Debian foi criado, percebeu-se a necessidade de um sistema que oferecesse um método que gerenciasse os pacotes instalados na máquina. O nome `dpkg` foi dado a esse sistema. Assim o famoso ‘pacote’ foi introduzido ao GNU/Linux. Logo depois a Red Hat resolveu criar seu próprio sistema `rpm`.

Rapidamente um novo dilema tomou conta dos produtores de GNU/Linux. Eles necessitavam de uma maneira rápida, prática e eficiente de instalar pacotes que tivessem suas dependências gerenciadas automaticamente e que cuidassem dos seus arquivos de configuração enquanto fossem atualizados. Assim, o Debian, novamente pioneiro, criou o APT, Advanced Packaging Tool, hoje portado pela Conectiva para o uso com `rpm` e adotado por algumas outras distribuições.

Este manual não pretende entrar na área do `apt-rpm`, como ficou conhecido o APT portado pela Conectiva, mas “patches” nesse sentido serão bem vindos.

Este manual é baseado na próxima versão do Debian, a *Sarge*.

### 1.1 Terminologia e conceitos básicos

Aqui você pode encontrar algumas terminologias e conceitos básicos utilizados ao longo deste manual:

**linha de fonte APT:** uma linha de fonte APT, também chamada de APT source, é uma linha especial que você adiciona num arquivo de configuração que diz ao APT onde ele deve pegar os pacotes, veja ‘O arquivo `/etc/apt/sources.list`’ on page 3.

**pacote binário:** um pacote binário é um arquivo `.deb` preparado para ser instalado pelo gerenciador de pacotes (`dpkg`), ele pode incluir arquivos binários mas pode também conter apenas dados independentes de arquitetura – ele é chamado de pacote binário assim mesmo.

**pacote nativo debian:** pacote criado especificamente para o Debian, geralmente este tipo de pacote tem os arquivos de controle debian dentro da fonte original e toda nova versão do pacote é também uma nova versão do programa ou dados originais.

**debianizar:** verbo geralmente utilizado que significa “preparar para usar com Debian” ou, simplificando, empacotado num formato `.deb`.

**pacote fonte:** um pacote fonte é realmente uma definição abstrata para um conjunto de dois ou três arquivos que são partes do arquivo fonte `deb`: um arquivo `.dsc`, que contém informações sobre o pacote, também chamado de arquivo de controle da fonte; um arquivo `.orig.tar.gz`, que contém os fontes originais do programa deste pacote – você pode também encontrar este nomeado simplesmente como `.tar.gz`, sem o `.orig`, caracterizando-se como um pacote nativo `debian`; um arquivo `.diff.gz`, que guarda as modificações feitas nos fontes originais para o pacote “`debianizado`” – você não encontrará este tipo de arquivo num pacote nativo `debian`.

**upstream:** esta palavra geralmente significa algo que vem do desenvolvedor do software ou dados originais, ou o próprio desenvolvedor.

**pacotes virtuais:** pacotes virtuais são pacotes que não na verdade não existem, mas que são serviços genéricos “oferecidos” por alguns pacotes específicos – o exemplo mais comum é o pacote `mail-transport-agent`, no qual pacotes que necessitam de um MTA<sup>1</sup> podem especificar uma dependência deixando o usuário escolher qual MTA será utilizado.

---

<sup>1</sup>MTA significa Agente de Transporte de Email

## Capítulo 2

# Configuração básica

### 2.1 O arquivo `/etc/apt/sources.list`

Para seu funcionamento, o APT utiliza um arquivo que lista as ‘fontes’ de onde ele obterá os pacotes. Esse arquivo é o `/etc/apt/sources.list`.

As entradas neste arquivo normalmente seguem o seguinte formato:

```
deb http://host/debian distribuição seção1 seção2 seção3
deb-src http://host/debian distribuição seção1 seção2 seção3
```

É claro que as entradas acima são fictícias e não devem ser usadas. A primeira palavra em cada linha, `deb` ou `deb-src`, indica o tipo de repositório: ou seja, os pré-compilados que normalmente usamos, ou se guarda pacotes fonte (`deb-src`), que são o fonte original do programa mais o arquivo de controle Debian (`.dsc`) e o `diff.gz` contendo as modificações necessárias para se ‘debianizar’ o programa, veja ‘Construindo a partir dos fontes’ on page 17.

Nós geralmente encontramos o seguinte num `sources.list` padrão do Debian:

```
# See sources.list(5) for more information, especially
# Remember that you can only use http, ftp or file URIs
# CDROMs are managed through the apt-cdrom tool.
deb http://http.us.debian.org/debian stable main contrib non-free
deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
deb http://security.debian.org stable/updates main contrib non-free

# Uncomment if you want the apt-get source function to work
#deb-src http://http.us.debian.org/debian stable main contrib non-free
#deb-src http://non-us.debian.org/debian-non-US stable/non-US main contrib no
```

Estas são as linhas necessárias para uma instalação básica do Debian. A primeira linha `deb` aponta para o repositório oficial, a segunda para um repositório non-US e a terceira para um repositório de atualizações de segurança do Debian.

As duas últimas linhas são comentadas (com um '#' na frente), então apt-get irá ignorá-las. Estas são as linhas `deb-src`, ou seja, elas apontam para pacotes fonte do Debian. Se você costuma fazer o download de fontes de programas para testar ou recompilar, descomente estas linhas.

O arquivo `/etc/apt/sources.list` pode conter vários tipos de linhas. O APT sabe lidar com repositórios `http`, `ftp`, `file` (arquivos locais, por exemplo, um diretório que contenha uma ISO montada) e `ssh`, até onde eu sei.

Não esqueça de atualizar a lista de pacotes depois de modificar o arquivo `/etc/apt/sources.list`, veja 'Atualizando a lista de pacotes disponíveis' on page 9. Você deve fazer isso para que o APT saiba quais pacotes estão disponíveis nas fontes que você especificou.

## 2.2 Adicionando um CD-ROM no arquivo `sources.list`

Se você preferir usar seu CD-ROM para instalar pacotes ou atualizar seu sistema automaticamente com APT, você pode adicioná-lo no seu arquivo `sources.list`. Para fazer isso, você pode usar o programa `apt-cdrom` desta forma:

```
# apt-cdrom add
```

com o CD-ROM do Debian no drive. Isso irá montar o CD-ROM, e se ele for um CD válido do Debian ele irá procurar por informações de pacotes no disco. Se sua configuração de CD-ROM for pouco usual, você pode também usar as seguintes opções:

```
-h          - Ajuda do programa
-d directory - Ponto de montagem do CD-ROM
-r          - Renomeia um CD-ROM reconhecido
-m          - Não montar
-f          - Modo rápido, não verifica os pacotes
-a          - Modo completo de scan
```

Por exemplo:

```
# apt-cdrom -d /home/kov/mycdrom add
```

Você pode também identificar um CD-ROM, sem adicioná-lo na sua lista:

```
# apt-cdrom ident
```

Note que este programa funciona somente se seu CD-ROM está configurado corretamente no sistema `/etc/fstab`.



## 2.3 O arquivo de configuração do APT

APT utiliza `/etc/apt/apt.conf` como seu principal arquivo de configuração. APT uses `/etc/apt/apt.conf` as its main configuration file. Contudo, como você irá notar, não existe um arquivo com este nome na instalação padrão, você pode cuidadosamente criar um e adicionar suas opções lá. Se você preferir, há uma maneira mais modular de configuração manual: você pode colocar arquivos individuais com os nomes que você escolher dentro de `/etc/apt/apt.conf.d/`.

Tenha cuidado com dois aspectos quando você escolher a forma modular: alguns pacotes Debian irão deixar suas configurações dentro daquele diretório, então você deve tentar prevenir conflitos de nomes, por exemplo, adicionando um sufixo `-local` no nome do arquivo. Além disso, a configuração é lida ordenadamente pelo nome do arquivo, então você pode adicionar um número no início do nome do seu arquivo para posicioná-lo na sequência. Por exemplo, você pode nomeá-lo como `00000myconf-local` se você quer ter certeza de que sua configuração será a primeira a ser considerada.

Para conhecer a sintaxe e opções aceitas por esses arquivos de configuração consulte a página de manual `apt.conf(5)`.

## 2.4 Usando o APT através de um proxy

Se você está usando o APT numa rede em que todas as conexões `http` e `ftp` são feitas através de um proxy, então você terá que configurar o APT para usar esse proxy. Você pode fazer isso editando o arquivo de configuração `/etc/apt/apt.conf` ou colocando um arquivo de configuração dentro do diretório `/etc/apt/apt.conf.d/`, que torna mais fácil de organizar as configurações (dê uma olhada em ‘O arquivo de configuração do APT’ on the current page).

Se esse arquivo não existe, então crie-o e adicione linhas como estas:

```
Acquire::http::proxy "http://proxy:port";
Acquire::ftp::proxy "http://proxy:port";
```

Substitua `proxy` e `port` pelas dadas pelo seu administrador da rede. Você pode também especificar o usuário e senha se o proxy necessitar, dessa forma:

```
Acquire::http::proxy "http://user:password@proxy:port";
```

Há muitas outras opções interessantes que você pode utilizar no arquivo de configuração `/etc/apt/apt.conf`, veja ‘O arquivo de configuração do APT’ on this page.



## Capítulo 3

# Gerenciando pacotes

### 3.1 Usando o Gerenciador de Pacote Debian: `aptitude`

O Debian conviveu muito tempo com um gerenciador de pacotes conhecido pela dificuldade de se entender e utilizar, chamado `dselect`. O sistema Debian está tendendo para um novo gerenciador padrão chamado `aptitude`, que é também baseado na biblioteca APT.

Se você instalou o Sarge você terá o `aptitude` instalado por padrão no sistema básico, se não for o caso, instale o pacote `aptitude`, usando por exemplo, este comando antes de continuar:

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

Tente evitar usar `apt-get` e use `aptitude`, ele sabe muito mais sobre o que APT está apt a fazer e tem algumas características particulares como marcar pacotes que foram automaticamente instalados para satisfazer dependências e assim pergunta se deve removê-los automaticamente quando eles não forem mais necessários.

O `aptitude` possui dois modos principais: um baseado numa interface ncurses em que você pode navegar por uma lista de pacotes, muito parecido com o `dselect`, e um modo baseado em linha de comando, muito parecido com o `apt-get`. A maior parte dos exemplos neste documento usa o modo de interface linha de comando, dando dicas de qual comando é usado para obter o mesmo resultado no modo de interface parecido com `dselect` do `aptitude`.

Para executar esta interface, digite `aptitude` num terminal – não há necessidade de ser root neste momento, o `aptitude` irá requisitar permissões de root automaticamente quando for necessário. Vamos ver o básico: para acessar o menu no topo da tela você tem que pressionar F10, como a segunda linha sugere. Esta segunda linha tem um resumo das funções mais utilizadas como obtendo ajuda de ajuda, saindo, atualizando lista de pacotes e instalando/removendo pacotes.

Dê uma olhada no `User Manual` (Manual do usuário) que está localizado no menu de ajuda para entender como essas coisas funcionam.

Note que o `aptitude` trabalha de forma similar a outros gerenciadores de pacotes: você pode fazer todas suas seleções e só depois então efetivar as modificações. Quase todas ações feitas no processo de instalação do pacote irão requerer que se pressione a tecla `g` duas vezes para serem efetivadas. A primeira vez que você pressiona a tecla `g` será apresentada a lista de modificações que serão feitas, então pressione `g` novamente para confirmar essas modificações ou `q` para voltar.

Similarmente, você pode pressionar a tecla `Enter` para obter mais detalhes sobre um pacote ou um grupo de pacotes – por exemplo, você pode pressionar `Enter` quando a seleção está em ‘Pacotes Novos’ para ver a lista de seções. Pressionando `Enter` numa seção você pode ver mais detalhes de pacote pressionando `Enter` nele. Note que entrando num detalhes de pacote irá mudar o contexto de visualização do `aptitude`, então você deverá pressionar `q` para voltar.

Os pacotes são categorizados por padrão com base no status de sua instalação e em outras categorias especiais. Você estará vendo isso quando entrar no `aptitude` pela primeira vez

```
--- Novos Pacotes
--- Pacotes Atualizáveis
--- Pacotes Instalados
--- Pacotes não Instalados
--- Pacotes Obsoletos e Criados Localmente
--- Pacotes Virtuais
--- Tarefas
```

Novos Pacotes tem os pacotes que estão no Debian desde a última vez que você limpou sua lista de novos pacotes usando a opção `Esquecer novos pacotes` no menu de `Ações`, ou através da tecla `"f"`.

Pacotes Atualizáveis são pacotes instalados que têm novas versões disponíveis. Pacotes Instalados são os pacotes propriamente instalados, Pacotes não Instalados são aqueles que você não instalou mas estão disponíveis numa das fontes APT. Pacotes Obsoletos e Criados Localmente são aqueles pacotes que estão disponíveis numa fonte APT ou algum pacote criado por você mesmo. Pacotes Virtuais são pacotes que definem um serviço em vez de um software em particular e que são “providos” por vários pacotes específicos. Finalmente, Tasks são as coleções de pacotes que servem a um propósito específico, também usado pelo programa `tasksel`.

Pressionando `Enter` numa dessas opções será expandida uma árvore como esta:

```
--\ Not Installed Packages
  --- admin - Utilitários administrativos (instalar software, gerenciar usuá
  --- base - O sistema básico Debian
  [...]
```

Indo mais fundo teremos:

```
--\ games - Jogos, brinquedos e programas divertidos
```

```

    --- contrib - Programas que dependem de software não contido no Debian
    --\ principal - O repositório principal do Debian
p      3dchess                                     <nenhum>      0.8.1

```

Isto nos mostra um pacote que é “não instalado”, que é um jogo e que está disponível num repositório principal do Debian, que está oficialmente na distribuição Debian.

No centro da tela há uma linha marcada que é preenchida com algum texto quando você seleciona um pacote. Isso é o que chamamos de *breve descrição* do pacote. Abaixo dela há uma caixa de texto contendo informações mais detalhadas do pacote. Esta é a *longa descrição*. Algumas vezes a longa descrição é maior que a área do retângulo reservada pra ela, então você pode pressionar a tecla “z” para descer a barra de rolagem e a tecla “a” para subir.

Veja ‘Instalando e reinstalando pacotes’ on the next page para mais informações sobre os dados mostrados aqui.

## 3.2 Atualizando a lista de pacotes disponíveis

O sistema de empacotamento usa uma base de dados própria para deixar as informações de quais pacotes estão instalados, quais não estão instalados e quais estão disponíveis para instalação. O programa `aptitude` usa esta base de dados para saber como instalar os pacotes requisitados pelo usuário e saber que pacotes adicionais são necessários para um pacote selecionado funcionar corretamente.

Para atualizar esta lista, você deve usar o comando `aptitude update`. Este comando procura as listas de pacotes nos repositórios encontrados em `/etc/apt/sources.list`; veja ‘O arquivo `/etc/apt/sources.list`’ on page 3 para mais informações sobre esse arquivo. Você pode também usar a tecla “u” dentro da interface ncurses do `aptitude` para fazer com que ele atualize suas listas de pacotes.

É uma boa prática rodar este comando regularmente para deixar você e seu sistema informados sobre possíveis atualizações de pacotes, especialmente atualizações de segurança.

## 3.3 Fazendo buscas por pacotes

`Aptitude` oferece uma poderosa e flexível funcionalidade de busca. Você pode acessar ela pressionando a tecla “/” (slash). Será aberta uma pequena caixa de diálogo no centro da tela.

Você pode então digitar o nome do pacote. `Aptitude` irá buscar de acordo com o que você digita. Se o pacote encontrado não é exatamente o que você quer, você pode pressionar enter para omitir a caixa de diálogo e então pressionar a tecla “n” para repetir a busca pela mesma cadeia de caracteres.

Você pode usar expressões regulares <sup>1</sup> na caixa de diálogo de busca, apenas atente para o fato de que o `aptitude` tem sua própria linguagem que às vezes conflita com uma expressão regular.

<sup>1</sup>expressões regulares são strings especiais que casam sentenças com base em algumas condições, dê uma olhada na página de manual `regex(7)`

Vamos ver um exemplo, então: se você quer ter uma ocorrência exata da cadeia de caracteres “gnome” digite “^gnome\$” na caixa de diálogo de busca.

Note que buscando dessa forma você simplesmente busca por nomes de pacotes. Se você quer fazer uma busca por descrições você precisa usar “~d” à antes do termo a ser buscado. Assim, se você quiser buscar por todos os pacotes que mencionem “isolinux” nas suas descrições você deverá escrever “~disolinux” na caixa de diálogo de busca.

Você pode também buscar por todos pacotes que estão atualmente quebrados buscando por “~b”, ou buscar por todos os pacotes que foram removidos mas que os arquivos de configuração foram mantidos através de “~c”.

Para buscas mais avançadas no aptitude veja o Manual do Usuário do Aptitude, que pode ser encontrado dentro do menu de Ajuda.

Você pode também usar a linha de comando para procurar por pacotes da seguinte maneira:

```
$ aptitude search '~c'
c A abiword-common          - WYSIWYG word processor based on GTK2
c A acme                    - Enables the "multimedia buttons" found
(...)
```

Como você pode ver, você pode usar qualquer opção de busca quando está usando a linha de comando, e aptitude oferece a você algumas informações de estado dos pacotes que nós iremos discutir no decorrer deste manual.

### 3.4 Instalando e reinstalando pacotes

Finalmente, o processo que todos estavam esperando! Com seu sources.list pronto e sua lista de pacotes disponíveis atualizada, tudo que você tem que fazer é rodar o aptitude para ter seu desejado pacote instalado.

Quando você entra no aptitude você pode instalar um ou mais pacotes pressionando a tecla “+” enquanto o pacote em questão está selecionado. Você pode usar a tecla “/” para buscar pelo pacote que voce quer, veja ‘Fazendo buscas por pacotes’ on the page before. Quando você requisita a instalação de um pacote ele é marcado com uma cor verde e a linha fica parecida com a linha deste exemplo:

```
pi      celestia-gnome          <nenhum>      1.3.0
```

Isto significa que o pacote não está instalado (‘p’) e está marcado para instalação. A palavra <nenhum> significa que não há versão instalada e o número da versão depois dela é a que está sendo marcada para instalação.

Note que quando você faz isso outros pacotes serão automaticamente marcados para instalação. São as dependências, sugestões e recomendações do pacote selecionado. Eles serão muito provavelmente marcados da seguinte forma:

```
piA  gnome-bin                                <nenhum>    1.4.2
```

O 'A' maiúsculo significa que esse pacote está sendo instalado porque outro pacote depende dele. Todos os pacotes instalados dessa forma serão removidos quando não forem mais necessários. Você pode marcá-lo manualmente digitando "M" com o cursor posicionado no pacote.

Você pode obter informações detalhadas sobre o pacote que você quer instalar pressionando Enter enquanto o pacote é selecionado. Isto irá retornar para você alguma coisa como:

```
i A --\ xterm                                4.2.1-14  4.2.1-1
  Description
xterm - X terminal emulator
xterm is a terminal emulator for the X Window System.  It provides DEC VT102
and Tektronix 4014 compatible terminals for programs that cannot use the
window system directly.  This version implements ISO/ANSI colors and most of
the control sequences used by DEC VT220 terminals.
  Priority: optional
  Section: x11
  Maintainer: Branden Robinson <branden@debian.org>
  Compressed size: 547k
  Uncompressed size: 1130k
  Source Package: xfree86
  --\ Depends
    --- libc6 (>= 2.3.2.ds1-4)
    --- libfreetype6 (>= 2.1.5-1)
    --- libncurses5 (>= 5.3.20030510-1)
    --- libxaw7 (> 4.1.0)
    --- xlibs (> 4.1.0)
    --- xlibs (> 4.2.0)
  --\ Conflicts
    --- xbase (< 3.3.2.3a-2)
    --- suidmanager (< 0.50)
  --\ Replaces
    --- xbase (< 3.3.2.3a-2)
  --- Packages which depend on xterm
  --\ Versions
p A 4.3.0-0pre1v4
i A 4.2.1-14
```

Note que quase sempre que você tem três ou mais traços você pode pressionar Enter para expandí-los. Será então exibido algo como --\ . Você pode ter informações interessantes sobre o pacote, como quais pacotes dependem dele, e quais os conflitos<sup>2</sup>, por exemplo.

---

<sup>2</sup>um pacote que conflita com outro pacote ou com uma versão específica de outro pacote não pode ser instalado no mesmo tempo que este outro pacote ou versão

Também note que no final da tela (você pode descer a tela usando as teclas de setas ou page down) você poderá encontrar diferentes *Versões* disponíveis para aquele pacote. O pacote `xterm` tem duas versões disponíveis pra mim, porque eu tenho mais do que uma fonte APT oferecendo ele em diferentes versões.

Você pode instalar as versões que você quiser pressionando “+” nela. Lembre que você precisa responder ao `aptitude` para efetivar as modificações que você fez. Você faz isso pressionando “g”. `Aptitude` irá então mostrar uma lista de ações a serem efetivadas onde você poderá ver se será feito o que você quer:

```
--\ Pacotes sendo automaticamente instalados para satisfazer dependências
ciA calctool                                <nenhum>      4.3.1
--\ Pacotes sendo mantidos
ih libgnomeprintui2.2-0                    2.4.0-1      2.4.2-1
--\ Pacotes a serem instalados
pi gwget2                                   <nenhum>      0.7-3
--\ Pacotes a serem removidos
ip qvwm                                     1:1.1.12-11:1.1.1
```

Essas informações devem ser entendidas facilmente. A coisa nova aqui é um pacote sendo mantido. Você pode ler mais sobre isso em `FIXME: hold reference`. Depois de verificar as modificações que você fez confirme pressionando a tecla “g” novamente.

Se você quiser reinstalar um pacote, digo, ter os arquivos que você removeu acidentalmente recuperados ou algo assim, você pode pressionar a tecla “L” para selecioná-lo para reinstalação. Você irá então se deparar com o mesmo processo como se você fosse instalá-lo.

Você pode também usar o `aptitude` sem a interface de usuário interativa para instalar ou reinstalar pacotes da seguinte forma:

```
# aptitude install pacote1 pacote2 ...
# aptitude --reinstall install pacote1 pacote2 ...
```

### 3.5 Instalado manualmente pacotes debian que foram baixados ou criados localmente

Se você fez o download de um arquivo `.deb` manualmente da internet ou tem um `.deb` de alguma fonte que não seja via APT você não poderá usar o `aptitude` ou algum outro programa baseado em APT para instalá-lo.

You’ll then need to use the underlying package management tool, which is the one that actually does the hard work of installing packages – even `aptitude` calls this tool for installing: `dpkg`.

Você pode instalar um pacote fazendo:

```
# dpkg -i arquivo.deb
```



Se você quiser usar a versão `unstable` do Debian ou gostaria de criar pacotes Debian, você deve ler mais sobre `dpkg`! Ele é uma ferramenta muito poderosa que tem muitas opções úteis.

### 3.6 Removendo pacotes

Se você não quer mais usar um pacote, você pode removê-lo do seu sistema usando o APT. Para fazer isso apenas digite “-” enquanto o pacote estiver selecionado no `aptitude`. Isso irá selecionar o pacote para remoção com uma linha roxa, como isso:

```
id    celestia-gnome                1.3.0-1          1.3.0-
```

Note que isso irá resultar numa linha parecida com essa, depois que o pacote é removido:

```
c    ocaml-base                      <nenhum>        3.07.2
```

Há um `c` no início da linha, em vez de um `p`. Isso significa que no meu sistema, o pacote `ocaml-base` foi removido mas os arquivos de configuração ainda permanecem. Para remover o pacote com seus arquivos de configuração você deve usar a tecla “\_”<sup>3</sup>, em vez de - isto é chamado de remoção completa (purge).

Uma observação importante: se você tentar remover um pacote que é uma dependência de um pacote instalado você irá acabar tendo pacotes quebrados, como a parte superior da tela do `aptitude` mostra:

```
aptitude 0.2.13      #Quebrado: 1    Serão liberados 208MB de espaço em disco
```

Os pacotes quebrados será marcados assim:

```
iBA  openoffice.org                1.1.0-3          1.1.0-3
```

Você pode tentar resolver os problemas de pacotes quebrados por você mesmo, tendo os pacotes atingidos removidos também, de forma que nenhum pacote quebrado seja deixado ou peça pro `aptitude` proceder, pressionando a tecla “g”. Ele então irá tentar consertar todos os pacotes quebrados. Se a solução efetuada pelo programa não satisfizer suas necessidades você poderá consertar manualmente antes de confirmar com a tecla “g”.

Você pode procurar por pacotes quebrados digitando “/” e usando “~b” como string de busca e pressionando `Enter`, você poderá buscar pelas próximas ocorrências usando a tecla “n”. Veja FIXME: busque referência!!

Como é de se esperar, você pode ter pacotes removidos ou removidos completamente (purged) através da interface de linha de comando, da seguinte forma:

```
# aptitude remove pacote1 pacote2 ...
# aptitude purge pacote1 pacote2 ...
```

<sup>3</sup>A tecla underscore, geralmente encontrada nas mesma tecla de traço é digitada pressionando-se a tecla shift junto com a tecla de traço.

### 3.7 Atualizando pacotes

Atualização de pacotes são um grande sucesso no sistema APT. Aptitude irá marcar automaticamente os pacotes que possuem novas versões, então tudo o que você tem que fazer é atualizar a lista de pacotes (veja ‘Atualizando a lista de pacotes disponíveis’ on page 9) e então pedir pro aptitude proceder, digitando a tecla “g” e, depois de rever as modificações, confirmar com “g” novamente.

Você pode checar a lista de pacotes que possuem novas versões disponíveis buscando na categoria Pacotes Atualizáveis:

```
--\ Pacotes Atualizáveis
  --\ libs - Collections of software routines
    --\ main - The main Debian archive
iu   libgnomeprintui2.2-0                               2.4.0-1    2.4.2-1
```

Pacotes marcados para atualização têm suas linhas com a cor ciano e têm o caractere u como ação que será efetivada. Se você quiser atualizar para uma nova release, então dê uma olhada na próxima seção, ‘Atualizando para uma nova release’ on the current page. Packages marked for upgrade have their lines cyan-colored and have the

### 3.8 Atualizando para uma nova release

Esta característica do APT permite que você atualize um sistema Debian inteiro através da internet ou de um novo CD (adquirido ou baixado com uma imagem ISO).

Por exemplo, suponha que você está usando a revisão 0 da versão estável (stable) do Debian e você compra um CD com a revisão 3. Você pode usar o APT para atualizar seu sistema a partir deste novo CD. Para fazer isso, use o apt-cdrom (veja a seção ‘Adicionando um CD-ROM no arquivo sources.list’ on page 4) para adicionar o CD no seu /etc/apt/sources.list e roda o aptitude para proceder com a instalação, como descrito acima.

É importante notar que o APT sempre procura pelas versões mais recentes dos pacotes. Dessa forma, se no seu /etc/apt/sources.list tiver listado um repositório que possua versões mais recentes de pacotes que as versoes do CD, APT deverá baixar os pacotes de lá.

Se você quiser atualizar seu sistema a partir da internet você apenas precisa configurar o arquivo /etc/apt/sources.list corretamente. Atualizar para uma nova revisão da versão estável é geralmente um processo normal de atualização do aptitude. Se seu /etc/apt/sources.list usa o apelido da release<sup>4</sup> você irá precisar atualizar ele para o novo nome ou mudar ele para ‘stable’.

Então, por exemplo, esta linha:

```
deb http://http.us.debian.org/debian woody main
```

<sup>4</sup>Por exemplo, o Debian 2.2 foi chamado de potato, o Debian 3.0 é o woody

Teria que ser como segue abaixo assim que o Sarge fosse lançado como estável (stable) para o computador estar atualizado com o sistema atual estável:

```
deb http://http.us.debian.org/debian sarge main
deb http://http.us.debian.org/debian stable main
```

A diferença aqui é que o primeiro irá deixar você com o sarge o tempo que você quiser, mesmo se sair uma nova versão estável. Isto pode ser útil se você quer decidir quando será feita a atualização para a nova versão estável. O segundo exemplo irá automaticamente atualizar seu sistema para a nova versão estável logo que ela for liberada.

### 3.9 Mantendo pacotes

Como dissemos na seção anterior, o `aptitude` irá marcar automaticamente pacotes para atualização. Se você quiser manter a versão atual instalada, você poderá fazer com que ele coloque o pacote como mantido.

Isto é feito pressionando a tecla “=”. Pacotes sendo mantidos em atualizações ficarão desta forma:

```
ih    alien                                8.41      8.41
```

Neste exemplo, o pacote `alien` será deixado an versão 8.41 mesmo se uma nova versão aparecer num fonte APT – note o caractere `h` à esquerda. Para tê-lo atualizado/atualizável novamente basta marcá-lo para instalação.

Você pode colocar pacotes como mantidos usando a linha de comando também, rodando:

```
# aptitude hold pacote1 pacote2 ...
```

### 3.10 Removendo arquivos de pacotes não utilizados: clean and autoclean

Quando você instala um pacote, o APT pega os arquivos necessários nos hosts listados em `/etc/apt/sources.list`, salva-os num repositório local (`/var/cache/apt/archives/`), e então procede com a instalação, veja ‘Instalando e reinstalando pacotes’ on page 10.

Dessa forma o repositório local pode crescer e ocupar muito espaço em disco. Felizmente, o APT oferece ferramtas para gerenciar seu repositório local: os métodos `clean` e `autoclean` do APT.

O método `clean` é chamado através do menu `Limpar cache de pacotes` no menu de Ações. Ele é usado para remover todos os arquivos `.deb` baixados. O método `autoclean` é chamado através da opção `Limpar arquivos obsoletos`, também no menu de Ações.

O método `autoclean` é para ser usado pelos que gostam de deixar uma cópia dos pacotes atuais instalados. Ele remove somente aqueles arquivos `.deb` que não são mais oferecidos por nenhuma fonte APT e são, além disso, obsoletos devido a uma nova versão.

## Capítulo 4

# Construindo a partir dos fontes

### 4.1 Fazendo o download de pacotes-fonte

É comum no mundo do software livre estudar código fonte ou mesmo fazer correções para códigos bugados. Construir a partir dos fontes pode também ser útil se você quer habilitar funcionalidades que estão desabilitadas no pacote oficial, ou desabilitar algumas que estão habilitadas. Para fazer isso você terá que fazer o download do código fonte do programa. O sistema APT oferece uma maneira fácil de obter o código fonte de muitos programas contidos na distribuição, incluindo todos os arquivos necessários para criar um `.deb` para o programa.

Outro uso comum de pacotes-fonte no Debian é para adaptar uma versão mais recente de um programa, a partir de uma distribuição instável (`unstable`), por exemplo, para usar com uma distribuição estável (`stable`). Compilar um pacote no estável irá gerar `.debs` com dependências ajustadas para os pacotes disponíveis nesta distribuição.

Para efetivar isso, a entrada `deb-src` no seu `/etc/apt/sources.list` deve ser apontada para `unstable`. Deve também estar habilitada (descomentada). Veja a seção ‘O arquivo `/etc/apt/sources.list`’ on page 3.

O Gerenciador de Pacotes Debian, Aptitude, é voltado para pacotes binários. Para usar pacotes-fonte nós teremos que usar o `apt-get`. Para baixar pacotes-fonte, você usará o seguinte comando:

```
$ apt-get source nomedopacote
```

Isto irá baixar três arquivos: um `.orig.tar.gz`, um `.dsc` e um `.diff.gz`. No caso de pacotes nativos debian, este último não é baixado e o primeiro provavelmente não terá “`orig`” no nome.

O arquivo `.dsc` é utilizado pelo `dpkg-source` para descompactar o pacote fonte dentro do diretório `nomedopacote-versao`. Em cada pacote-fonte baixado existe um diretório `debian/` que contém os arquivos necessários para se criar o pacote `.deb`.

Para auto-construir o pacote quando ele é baixado, apenas adicione `-b` na linha de comando, como abaixo:

```
$ apt-get -b source nomedopacote
```

Se você decidir não criar o `.deb` no momento do download, você poderá criar ele depois rodando o seguinte comando dentro do diretório que foi criado para o pacote depois do download:

```
$ dpkg-buildpackage -rfakeroot -uc -b
```

Observe que para construir pacotes você irá precisar pelo menos dos pacotes: `devscripts`, `dpkg-dev`, `debhelper`, `fakeroot`, dê uma olhada em ‘Instalando e reinstalando pacotes’ on page 10 e instale-os antes de proceder. A maioria dos pacotes irão depender de ferramentas de compilação, então pode ser útil instalar o pacote `build-essential` também. Outros pacotes podem ser necessários, veja ‘Pacotes necessários para compilar pacotes-fonte’ on this page para mais informações.

Para instalar o pacote construído pelos comandos acima deve-se usar o gerenciador de pacotes diretamente. Veja ‘Instalado manualmente pacotes debian que foram baixados ou criados localmente’ on page 12.

Há uma diferença entre o método de fonte do `apt-get` e seus outros métodos. O método de fonte (source method) pode ser usado por usuários normais, sem necessidade de poderes especiais de root. Os arquivos são baixados para o diretório no qual o comando `apt-get source pacote` foi chamado.

## 4.2 Pacotes necessários para compilar pacotes-fonte

Geralmente, cabeçalhos (headers) específicos e bibliotecas compartilhadas precisam estar presentes para um pacote-fonte ser compilado. Todos os pacotes têm um campo nos seus arquivos de controle (control) chamado ‘Build-Depends:’ que indica quais pacotes adicionais são necessários para os pacotes serem construídos a partir dos fontes. Alguns pacotes básicos são também necessários, veja ‘Fazendo o download de pacotes-fonte’ on the preceding page antes de continuar.

o APT tem uma forma simples de baixar esses pacotes. Apenas rode `apt-get build-dep pacote`, onde ‘pacote’ é o nome do pacote que você está querendo construir. Por exemplo:

```
# apt-get build-dep gmc
Lendo Lista de Pacotes... Pronto
Construindo Árvore de Dependências... Pronto
Os NOVOS pacotes a seguir serão instalados:
  comerr-dev e2fslibs-dev gdk-implib-dev implib-progs libgnome-dev libgnorba-de
  libgpmgl-dev
0 pacotes atualizados, 7 pacotes novos instalados, 0 a serem removidos e 1 não
É preciso fazer o download de 1069kB de arquivos.
Depois de desempacotamento, 3514kB adicionais de espaço em disco serão usados
Quer continuar? [S/n]
```

Os pacotes que serão instalados são os pacotes necessários para `gmc` ser construído corretamente. É importante observar que este comando não fará o download do pacote fonte a ser compilado. Você irá ainda precisar rodar `apt-get source` para pegar ele.

Se tudo que você quer é checar quais pacotes são necessários para construir um dado pacote, existe um método `showpkg` para o comando `apt-cache` (veja 'Obtendo informações sobre pacotes.' on page 23), que irá mostrar, além de outras informações, a linha `Build-Depends` daquele pacote.

```
# apt-cache showsrc pacote
```

### 4.3 Construindo um pacote de depuração

Se você quiser construir um pacote para propósitos de depuração, por exemplo, reportar um bug – ou consertar um, você pode usar variáveis de ambientes que são suportadas pela maioria dos pacotes Debian.

Para construir um pacote que contém binários com símbolos de depuração<sup>1</sup> tudo o que você precisa fazer é prefixá-lo com `DEB_BUILD_OPTIONS=nostrip`. Otimizações podem também tornar depurações mais difíceis, então você pode desativá-las adicionando a string `noopt` na variável `DEB_BUILD_OPTIONS`. Exemplo:

```
$ DEB_BUILD_OPTIONS="nostrip noopt" dpkg-buildpackage -rfakeroot -uc -b
```

### 4.4 Opções para construção de pacotes personalizados

Se o que você quer é personalizar a forma como o pacote é construído, você terá que editar o arquivo `debian/rules`. Isto significa: dentro do diretório principal criado quando o pacote fonte é extraído haverá um diretório `debian`, que contém muitos arquivos. Um deles é especial: o arquivo `rules`.

Esse arquivo é normalmente um Makefile que tem alvos para a configurar (`configure`), construir (`build`), instalar (`install`) e criar o pacote. Por exemplo, se eu quero ter o pacote `luola`<sup>2</sup> sem som eu posso, depois de fazer o download e extrair seus fontes editar o arquivo `debian/rules` dessa forma:

```
[...]
configure: configure-stamp
configure-stamp:
    dh_testdir
```

<sup>1</sup>Geralmente, o Debian remove os símbolos de depuração de todas as bibliotecas e aplicativos que distribui. Isto significa que todos os símbolos de depuração são removidos.

<sup>2</sup>Um jogo muito bom de batalha naval que falta ter suporte a rede. Alguém pra pegar? =)

```
# Add here commands to configure the package.
./configure $(confflags) \
--prefix=/usr \
--mandir=share/man \
--infodir=share/info \
--datadir=share/games \
--bindir=games \
--enable-sound
# --enable-sdl-gfx

touch configure-stamp
[...]
```

Está vendo o `--enable-sound`? Se eu removê-lo ou repassá-lo com `--disable-sound` e então reconstruir o pacote usando a técnica descrita em ‘Fazendo o download de pacotes-fonte’ on page 17 eu terei um pacote luola construído para não usar som.

Se você realmente quer trabalhar com pacotes fonte no dia-a-dia eu sugiro que leia o Guia dos Novos Mantenedores Debian (<http://www.debian.org/doc/maint-guide/>) e o Manual de Políticas Debian (<http://www.debian.org/doc/debian-policy/>), principalmente. Outros documentos disponíveis no Canto dos Desenvolvedores Debian (<http://www.debian.org/devel/index.pt.html>) podem ser úteis também.

## 4.5 Mas ei, eu não quero usar essas funcionalidades do Debian!

Algumas vezes, pessoas querem usar uma versão específica de um programa disponível somente em código fonte, sem pacote Debian. Mas o sistema de empacotamento pode ser um problema nesse caso. Suponha que você quer compilar uma nova versão do seu servidor de email. Tudo está bem, mas muitos pacotes no Debian dependem de um MTA (Agente de Transporte de Email). Desde que você tenha instalado algo que você mesmo compilou, o sistema de empacotamento não terá conhecimento sobre isso.

É aí que o `equivs` entra em cena. Para usá-lo, instale o pacote com esse nome. `Equivs` cria um pacote vazio que “simula” as dependências, fazendo com que o sistema de empacotamento acredite que as dependências estão satisfeitas.

Antes de começarmos, é bom lembrar que existem maneiras mais seguras de compilar um programa que já é empacotado para Debian com diferentes opções, e que nesse caso não se deve usar `equivs` para repassar dependências se não se sabe o que está fazendo. Veja ‘Fazendo o download de pacotes-fonte’ on page 17 para mais informações.

Continuando com o exemplo do MTA, você apenas instalou seu novo `postfix` compilado e foi instalar o pacote `mutt`. De repente você descobre que o `mutt` que instalar outro MTA. Mas você já tem o seu instalado.

Vá para algum diretório (`/tmp`, por exemplo) e rode:



```
# equivs-control nome
```

Repasse *nome* pelo nome do arquivo de controle (control) que você quer criar. O arquivo será criado como se segue:

```
Section: misc
Priority: optional
Standards-Version: 3.5.10

Package: <enter package name; defaults to equivs-dummy>
Version: <enter version here; defaults to 1.0>
Maintainer: Your Name <yourname@foo.com>
Pre-Depends: <packages>
Depends: <packages>
Recommends: <packages>
Suggests: <package>
Provides: <(virtual)package>
Architecture: all
Copyright: <copyright file; defaults to GPL2>
Changelog: <changelog file; defaults to a generic changelog>
Readme: <README.Debian file; defaults to a generic one>
Extra-Files: <additional files for the doc directory, commaseperated>
Description: <short description; defaults to some wise words>
    long description and info
.
second paragraph
```

Nós apenas precisamos modificar isso para o que nós queremos. Tenha atenção no formato dos campos e nas suas descrições, aqui não é necessário explicar cada um, vamos fazer o que é preciso:

```
Section: misc
Priority: optional
Standards-Version: 3.0.1

Package: mta-local
Provides: mail-transport-agent
```

Sim, isso é tudo. O mutt depende do mail-transport-agent, que é um pacote virtual oferecido por todos MTAs, o “registro” mta-local é como um mail-transport-agent usando o campo Provides.

Agora você somente precisa construir o pacote:

```
# equivs-build nome
dh_testdir
touch build-stamp
dh_testdir
dh_testroot
dh_clean -k
# Add here commands to install the package into debian/tmp.
touch install-stamp
dh_testdir
dh_testroot
dh_installdocs
dh_installchangelogs
dh_compress
dh_fixperms
dh_installdeb
dh_gencontrol
dh_md5sums
dh_builddeb
dpkg-deb: building package `name' in `../nome_1.0_all.deb'.
```

The package has been created.

Attention, the package has been created in the current directory,

E instale o `.deb` resultante. Veja ‘Instalado manualmente pacotes debian que foram baixados ou criados localmente’ on page [12](#).

Como se pode ver, há diversas utilidades para `equivs`. Uma é poder criar um pacote `meus-favoritos`, que é dependência de programas que você usualmente instala, por exemplo. Libere sua imaginação, mas seja cuidadoso.

É importante observar que há exemplos de arquivos de controle (`control`) em `/usr/share/doc/equivs/examples`. Dê uma olhada neles.

## Capítulo 5

# Obtendo informações sobre pacotes.

Existem alguns programas que fazem interface para o sistema APT, que o faz significativamente fácil para listar os pacotes que estão disponíveis para instalação ou que já estão instalados, como também procurar em que seção um pacote está, qual sua prioridade, sua descrição, etc.

Mas... nosso objetivo aqui é aprender como usar o APT puro. Então como você busca o nome de um pacote que você quer instalar?

Isto é que este capítulo tem a intenção de responder. Vamos ver nossas opções.

### 5.1 Descobrindo nome de pacotes

Por exemplo, suponha que você quer reviver os bons tempos de Atari 2600. Você quer usar o APT para instalar um emulador Atari, e então baixar alguns jogos. Você pode fazer:

```
$ aptitude search atari
p   atari-fdisk-cross          - Partition editor for Atari (running on
p   atari800                   - Atari emulator for X/curses/SDL
p   console-keymaps-atari      - Keymaps for Atari keyboards.
```

Nós encontramos diversos pacotes relacionados com o que estávamos procurando, juntos com breves descrições. A letra p no início da linha é similar à notação utilizada na interface curses aptitude. Isso significa, então, que o pacote não está instalado. É importante notar que o aptitude somente procura por nomes de pacotes por padrão. Você pode usar todas as opções de busca oferecidas pelo aptitude, que você pode aprender lendo o Manual de Usuário do aptitude. Para buscar por descrições, por exemplo, você poderia usar:

```
$ aptitude search ~datari
p   aranym                     - Atari Running on Any Machine
p   atari-fdisk-cross          - Partition editor for Atari (running on
p   atari800                   - Atari emulator for X/curses/SDL
```

```

p   circuslinux           - The clowns are trying to pop balloons t
p   circuslinux-data      - Data files for circuslinux
p   console-keymaps-atari - Keymaps for Atari keyboards.
[...]
p   stella                - Atari 2600 Emulator for SDL & X windows
[...]

```

Agora retornou muitos mais pacotes, como podemos ver. Agora que nós encontramos uma lista de possíveis soluções para nosso problema, vamos nos aprofundar neles:

```

$ aptitude show stella
Package: atari-fdisk-cross
Version: 0.7.1-5
Priority: extra
Section: otherosfs
Maintainer: Roman Hodek <roman@hodek.net>;
Uncompressed Size: 106k
Description: Partition editor for Atari (running on non-Atari)
 Atari-fdisk allows you to create and edit the partition table of a disk
 partitioned in Atari format. It supports both the AHDI 5.0 and ICD variation
 of the Atari format. It is an interactive tool with a menu similar to PC
 fdisk, and also supports most options of PC fdisk.

```

Nessa saída você tem muitos detalhes sobre o pacote que você quer (ou não quer) instalar, junto com a descrição completa do pacote. Você pode também usar o programa `apt-cache` para obter informações como estas. Se você escolher usar essa ferramenta, poderá mostrar múltiplas versões de um pacote que está disponível, por exemplo:

```

$ apt-cache show muine
Package: muine
Version: 0.5.0-1
Priority: optional
Section: gnome
Maintainer: Link Dupont <link@subpop.net>
Depends: gstreamer-gconf (>= 0.6.4), libatk1.0-0 (>= 1.4.1), libbonobo2-0 (>=
Architecture: i386
Filename: ./pool/main/m/muine/muine_0.5.0-1_i386.deb
Size: 164314
Installed-Size: 692
MD5sum: 9885f13e5ef4f76b3bf6fe7bb3ea8634
Description: Simple music player
 Muine is an innovative music player. It has a simple interface designed to
 allow the user to easily construct playlists from albums and/or single songs
 Its goal is to be simply a music player, not to become a robust music
 management application.

```

```

Package: muine
Priority: optional
Section: gnome
Installed-Size: 492
Maintainer: Link Dupont <link@subpop.net>
Architecture: i386
Version: 0.4.0-8
Depends: gstreamer-gconf (>= 0.6.4), libatk1.0-0 (>= 1.4.1), libbonobo2-0 (>=
Filename: pool/main/m/muine/muine_0.4.0-8_i386.deb
Size: 139864
MD5sum: e746709ad6a6fcc1e9c46b46b3d6e5b2
Description: Simple music player
  Muine is an innovative music player. It has a simple interface designed to
  allow the user to easily construct playlists from albums and/or single songs
  Its goal is to be simply a music player, not to become a robust music
  management application.

```

Para mais informações gerais sobre um pacote, você pode usar:

```

# apt-cache showpkg penguin-command
Package: penguin-command
Versions:
1.4.5-1(/var/lib/apt/lists/download.sourceforge.net_debian_dists_unstable_ma

Reverse Depends:
Dependencies:
1.4.5-1 - libc6 (2 2.2.1-2) libpng2 (0 (null)) libsdl-mixer1.1 (2 1.1.0) libs
Provides:
1.4.5-1 -
Reverse Provides:

```

E para somente achar quais pacotes ele depende:

```

# apt-cache depends penguin-command
penguin-command
  Depends: libc6
  Depends: libpng2
  Depends: libsdl-mixer1.1
  Depends: libsdl1.1
  Depends: zlib1g

```

Resumindo, nós temos uma série de possibilidades que podemos usar para encontrar o nome de um pacote que queremos.

## 5.2 Usando o dpkg para encontrar nomes de pacotes

Uma das formas de localizar o nome de um pacote é conhecendo o nome de um arquivo importante do pacote. Por exemplo, para encontrar o pacote que oferece um arquivo ".h" em particular que você necessita para compilação você pode rodar:

```
$ dpkg -S stdio.h
libc6-dev: /usr/include/stdio.h
libc6-dev: /usr/include/bits/stdio.h
perl: /usr/lib/perl/5.6.0/CORE/nostdio.h
```

ou:

```
$ dpkg -S /usr/include/stdio.h
libc6-dev: /usr/include/stdio.h
```

Observe que este método funciona somente para procurar nomes de pacotes que estão instalados no seu sistema. Você terá que usar `auto-apt` (veja 'Como instalar pacotes "sob demanda"' on page 29) ou `apt-file` (veja 'Como descobrir a que pacote um arquivo pertence' on this page) para procurar por arquivos em pacotes que não estão instalados. Você pode também dar uma olhada no sítio <http://packages.debian.org/>. Para listar nomes de pacotes instalados no seu sistema, que é útil, por exemplo se você planeja limpar seu disco rígido, você pode rodar:

```
$ dpkg -l | grep mozilla
ii  mozilla-browser 0.9.6-7          Mozilla Web Browser
```

O problema com esse comando é que ele "quebra" o nome do pacote. No exemplo acima, o nome completo do pacote é `mozilla-browser`. Para consertar isso, você pode usar a variável de ambiente `COLUMNS` desta forma:

```
$ COLUMNS=132 dpkg -l | grep mozilla
ii  mozilla-browser 0.9.6-7          Mozilla Web Brows
```

## 5.3 Como descobrir a que pacote um arquivo pertence

MISSING: If you want to install a package, and you can't find out what it is called by searching with `apt-cache`, but know the filename of the program itself, or some other filename that belongs to the package, then you can use `apt-file`, which is included in the package of same name to find the package name. This is done like this:

```
$ apt-file search nomedoarquivo
```

Ele funciona como o `dpkg -S`, mas mostrará também para você pacotes não instalados que contêm o arquivo. Ele poderia também ser usado para buscar qual pacote contém arquivos cabeçalho (includes) necessários que estejam faltando quando se compila programas, embora `auto-apt` seja um método muito melhor para resolver isso, veja ‘Como instalar pacotes “sob demanda”’ on page 29.

Você pode também listar o conteúdo de um pacote, rodando:

```
$ apt-file list nomedopacote
```

`apt-file` mantém uma base de dados dos arquivos que todos os pacotes contém, igualmente ao que `auto-apt` faz ele precisa ser atualizado. Isto é feito rodando:

```
# apt-file update
```

Por padrão, `apt-file` usa a mesma base de dados que o `auto-apt` está usando, veja ‘Como instalar pacotes “sob demanda”’ on page 29.

## 5.4 Informação sobre um pacote na WEB

Há vários recursos na WEB com informações sobre os pacotes disponíveis na distribuição Debian, a maioria direcionada principalmente para Desenvolvedores Debian, mas muitas delas podem ser úteis para usuários também.

As Páginas de Pacotes Debian (<http://packages.debian.org/>) deixam você buscar por pacotes que estão disponíveis em várias arquiteturas suportadas pelo Debian e, também, buscar por conteúdo de todos os pacotes Debian. Há informações sobre dependências e outras relações entre pacotes. Há também links para download dos fontes dos pacotes e o pacotes binários para todas as arquiteturas. Você pode usar um atalho como `http://packages.debian.org/nomedopacote` para ter acesso mais rápido ao link do pacote.

O Sistema de Rastreamento de Pacotes (Package Tracking System) (<http://packages.qa.debian.org/>) oferece informação sobre o que aconteceu a um pacote recentemente, quais são os itens a serem feitos (TODO), do ponto de vista de um mantenedor Debian oferece um sumário de bugs reportados e algumas mais informações muito úteis. Uma das coisas boas no Sistema de Rastreamento de Pacotes é que ele permite que você se “inscreva” em um pacote para acompanhar todos os emails que o mantenedor frequentemente recebe sobre o pacote. Desta forma você pode acompanhar o desenvolvimento de pacotes que são cruciais para seu trabalho ou diversão. O atalho `http://packages.qa.debian.org/nomedopacote` funciona aqui também.

Finalmente, o Sistema de Rastreamento de Bugs (Bug Tracking System) (<http://bugs.debian.org/>) oferece informações sobre bugs conhecidos nos pacotes distribuídos pelo Debian. Isto pode ser útil para você encontrar porque algo não está funcionando da forma que

you were expecting and thus search for solutions or work on them through the bug logs. Additionally, read these pages and their documentation to be able to send bug reports to Debian. The shortcut `http://bugs.debian.org/nomedopacote` also works here, as expected, but the BTS (Bug Tracking System) also accepts other shortcuts very useful like, for example, `http://bugs.debian.org/src:nomedopacote` to show all error reports (bug reports) in all binary packages provided by a specific source package.



## Capítulo 6

# Truques e técnicas

### 6.1 Como instalar pacotes “sob demanda”

Você está compilando um programa e, de repente, boom! Há um erro porque ele precisa de um arquivo `.h` que você não tem. O programa `auto-apt` pode te salvar desse cenário, ele é disponibilizado no pacote de mesmo nome. Ele pede a você para instalar pacotes se eles são necessários, parando o processo e continuando após o pacote estar instalado.

O que você faz, basicamente, é rodar:

```
# auto-apt run comando
```

Onde ‘comando’ é o comando a ser executado que pode necessitar de algum pacote não disponível. Por exemplo:

```
# auto-apt run ./configure
```

Ele irá então pedir para instalar os pacotes necessários e chamar o `apt-get` automaticamente. Se você está rodando o `X`, uma interface gráfica irá substituir a interface padrão em modo texto.

`Auto-apt` mantém base de dados que precisam ser atualizadas, para ele ser eficaz. Isto é feito chamando os comandos `auto-apt update`, `auto-apt updatedb` e `auto-apt update-local`.

### 6.2 Fazendo cache de pacotes

Se você está lidando com diversas máquina numa rede pode ser útil ter um sistema de cache de pacotes, então você não terá que baixar os pacotes para cada máquina. Este manual irá focar no pacote `apt-proxy`, que faz exatamente isso. Existem outras soluções, como `apt-cacher`. Você pode experimentar cada uma para escolher a mais adequada para suas necessidades.

Primeiro de tudo, instale o pacote `apt-proxy`. Ele irá registrar no `inetd` para ouvir requisições na porta 9999, você pode precisar reiniciar serviço `inetd`.

Agora você terá que editar o arquivo `/etc/apt-proxy/apt-proxy.conf`. Ele lista os servidores reais `apt-proxy` que serão utilizados para fazer o download das listas de pacotes e dos pacotes. Você pode usar `http`, `ftp` e `rsync` como métodos de transferência. O arquivo vem com um “backend” padrão assim:

```
add_backend /debian/ \
    $APT_PROXY_CACHE/debian/ \
    http://ftp.us.debian.org/debian/ \
    http://ftp.de.debian.org/debian/ \
    http://ftp.uk.debian.org/debian/ \
    +ftp.us.debian.org::debian/
```

Isto significa que sempre que um cliente tentar pegar alguma coisa de `/debian/ apt-proxy` usará seu cache, os espelhos `http` Debian listados e um servidor `rsync`, que é preferencial para baixar as listas de pacotes (o sinal “+” significa preferencia para arquivos de controle - control).

Como, então, um cliente irá usar este backend? Adicionando a seguinte linha no arquivo `/etc/apt/sources.list` dos clientes (incluindo o bloco em que o `apt-proxy` está instalado):

```
deb http://servidor:9999/debian/ stable main
```

A linha se parece muito com uma linha normal. A diferença aqui é que você coloca o servidor `apt-proxy` onde você normalmente colocaria um espelho `http` ou `ftp`, usa um valor de porta (9999) e então seleciona o backend (`/debian/`). Depois de ter feito toda essa configuração inicial atualize a lista de pacotes em uma das máquinas e atualize ela primeiro, dessa forma será feito o download somente uma vez. Depois que a primeira máquina estiver atualizada, atualize as outras.

Você pode usar o comando `apt-proxy-import` para importar os pacotes que estão dentro do seu cache APT atual. (`/var/cache/apt/archives/`) rodando `apt-proxy-import /var/cache/apt/archives`. Note que você deve ter executado o processo de `update` em pelo menos um cliente para inicializar o cache `apt-proxy` antes de usar o `apt-proxy-import`.

Você pode aprender mais sobre `apt-proxy` lendo os comentários que compõem o arquivo `/etc/apt-proxy/apt-proxy.conf`. Para configurar proxies `http` e `ftp` para o `apt-proxy` usar, por exemplo, você encontrará exemplos de configuração no final do arquivo.

### 6.3 Decidindo qual é o melhor espelho para ser incluído no arquivo `sources.list`: `netselect`, `netselect-apt`

Uma dúvida muito frequente, principalmente entre os novos usuários é: “qual espelho Debian incluir no `sources.list`?”. Há muitas maneiras de decidir qual o espelho Os mais experien-

tes provavelmente têm um script que mede o tempo de ping em diversos espelhos. Mas existe um programa que faz isso para nós: **netselect**.

Para instalar o netselect, como se faz normalmente:

```
# aptitude install netselect
```

Executá-lo sem parâmetros irá exibir ajuda. Executando ele com uma lista de hosts (espelhos) separados por espaço, ele retornará uma pontuação e um dos hosts. Esta pontuação leva em consideração o tempo estimado do ping e o número de hops (hosts tais que uma requisição de rede irá passar até chegar ao destino) e é inversamente proporcional à velocidade de download estimada (assim, quanto menor, melhor). O host retornado é aquele que tem a menor pontuação (a lista completa de pontuação pode ser visualizada adicionando a opção -vv). Veja este exemplo:

```
# netselect ftp.debian.org http.us.debian.org ftp.at.debian.org download.unes
365 ftp.debian.org.br
#
```

Isto significa que, dos espelhos incluídos como parâmetros para o netselect, ftp.debian.org.br foi o melhor, com uma pontuação de 365. (Atenção!! Como isso foi feito no meu computador e a topografia da rede é extremamente diferente dependendo do ponto de conexão, este valor não tem necessariamente a melhor velocidade em outros computadores). This means that, from the mirrors included as parameters to netselect, ftp.debian.org.br was the best, with an score of 365. (Attention!! As it was done on my computer and the network topography is extremely different depending on the contact point, this value is not necessarily the right speed in other computers).

Agora, basta colocar o espelho mais rápido encontrado, no arquivo /etc/apt/sources.list (veja 'O arquivo /etc/apt/sources.list' on page 3) e siga as dicas em 'Gerenciando pacotes' on page 7.

**Note:** a lista de espelhos podem sempre ser encontradas no arquivo [http://www.debian.org/mirror/mirrors\\_full](http://www.debian.org/mirror/mirrors_full).

A partir da versão 0.3, o pacote netselect inclui o script **netselect-apt**, que faz o processo acima automaticamente. Basta entrar com a distribuição como parâmetro (o padrão é stable) e o arquivo sources.list será gerado com os melhores espelhos principais (main) e não-EUA (non-US) e será salvo no diretório corrente. O exemplo seguinte gera um sources.list de uma distribuição estável (stable):

```
# ls sources.list
ls: sources.list: File or directory not found
# netselect-apt stable
(...)
# ls -l sources.list
sources.list
#
```

**Lembre-se:** o arquivo `sources.list` é gerado no diretório corrente, e deve ser movido para o diretório `/etc/apt`.

## 6.4 Removendo arquivos de localização (locale) não utilizados: `localepurge`

Muitos usuários Debian usam somente um locale. Um usuário Debian brasileiro, por exemplo, geralmente usa o locale `pt_BR` toda vez e não tem sequer conhecendo sobre o `es`.

`localepurge` é uma ferramenta muito útil para estes usuários. Você pode liberar bastante espaço tendo somente os locales que você realmente usa. Basta executar `apt-get install localepurge`.

É muito fácil para configurá-lo, questionamentos do `debconf` guiam passo a passo o usuário para a configuração. Tenha muito cuidado ao responder a primeira questão, respostas incorretas podem remover todos os arquivos locais, inclusive os que você usa. A única forma de recuperar esses arquivos é reinstalando todos os pacotes que os provêm.

## 6.5 Como ficar informado sobre as modificações nos pacotes.

Todo pacote instala no diretório de documentação (`/usr/share/doc/nomedopacote`) um arquivo chamado `changelog.Debian.gz` que contém a lista de modificações feitas desde a última versão. Você pode ler esses arquivos com `zless`, por exemplo, mas isso não é algo tão fácil, depois de uma atualização completa do sistema, iniciar a busca por changelogs para todo pacote atualizado.

Há uma forma para automatizar esta tarefa através de uma ferramenta chamada `apt-listchanges`. Para iniciar com ela é preciso instalar o pacote `apt-listchanges`. Durante a instalação do pacote, o `Debconf` irá configurá-lo. Responda as questões como você quiser.

A opção “Devo deixar `apt-listchanges` ser automaticamente executado pelo `apt`?” é muito útil porque ela mostra uma lista de modificações feitas para cada pacote que está sendo instalado pelo `apt` durante uma atualização e deixa você analisá-las antes de continuar. A opção “`apt-listchanges` deve perguntar por confirmação depois de mostrar as modificações?” é útil porque ela pergunta se você quer continuar a instalação depois de ler a lista de modificações. Se você disse que não quer continuar `apt-listchanges` irá retornar um erro e o `apt` irá abortar a instalação.

Tão logo o `apt-listchanges` estiver instalado, os pacotes baixados (ou pegos de um CD ou disco montado) pelo `apt`, serão exibidas as listas de modificações feitas para aqueles pacotes antes de instalá-los.

## Capítulo 7

# **Estando no outro lado: criando um repositório**



## Capítulo 8

# Interfaces alternativas para usuários gerenciarem pacotes

Aptitude é o que este como-fazer recomenda como a principal interface para gerenciador de pacotes para seu sistema Debian. Isto deve-se, como foi mostrado ao longo deste documento, que o aptitude tem uma série de características que faz a vida do administrador mais fácil.

Há, porém, outras interfaces para usuários, inclusive gráficas, como Synaptic ou GNOME-APT. Neste capítulo nós estaremos nos aprofundando nelas.

### 8.1 Synaptic

Synaptic foi originalmente escrita por Alfredo Kojima para a Conectiva. Hoje é mantida por pessoas da Conectiva e Debian. Ela foi originalmente escrita em wing, um toolkit gráfico do Window Maker, mas foi reescrita em Gtk+.

Para usar o Synaptic você tem que instalar o pacote `synaptic`. Outra opção, se você quiser dar uma olhada no sistema de `debtags` (<http://deb-usability.alioth.debian.org/debtags/index.html>), um novo sistema para organizar pacotes que está sendo desenvolvido pela iniciativa Debian Usability (<http://deb-usability.alioth.debian.org/>), é instalar o pacote `synaptic-debtags`, que oferece um testador para essa iniciativa.

Synaptic é provavelmente a mais utilizada das interfaces gráficas para APT, aquela que os usuários finais irão provavelmente ter menos problemas.

### 8.2 GNOME-APT

GNOME-APT foi inicialmente escrito por Havoc Pennington e Diego Lages para o sistema Debian. Ele deixou de ser mantido por algum tempo mas pessoas deixaram ele vivo para o Woody. Para o Sarge o pacote ganhou um novo mantenedor que o reviveu. Ao contrário do

Synaptic, GNOME-APT é para aquelas pessoas que, enquanto curtem uma agradável interface GTK+, preferem flexibilidade à usabilidade.

Você pode usar GNOME-APT instalando o pacote `gnome-apt`.

### **8.3 KPackage FIXME!!**



---

## Capítulo 9

# Como lidar com erros

### 9.1 Erros comuns

Erros sempre acontecerão, muitos deles causados por usuários desatenciosos. Segue uma lista de alguns dos erros reportados mais frequentes e como lidar com eles.

Se você recebe uma mensagem como esta abaixo quando tenta executar `apt-get install pacote...`

```
Lendo Lista de Pacotes... Pronto
Construindo Árvore de Dependências... Pronto
W: Couldn't stat source package list 'http://people.debian.org unstable/ Pack
W: You may want to run apt-get update to correct these missing files
E: Couldn't find package penguineyes
```

you forgot to run `apt-get update` after your last modification to the file `/etc/apt/sources.list`.

If the error seems like this:

```
E: Não foi possível abri arquivo de trava /var/lib/dpkg/lock - open (13 Permi
E: Unable to lock the administration directory (/var/lib/dpkg/), are you root
```

When you try any `apt-get` method that is not `source`, you do not have root permission, this is, you are running as a normal user.

There is a similar error above that happens when you run two copies of `apt-get` at the same time, or if you try to run `apt-get` while a `dpkg` process is active. The only method that can be used simultaneously with others is the `source` method.

If you are in the middle of an installation and you encounter the impossibility to install or remove packages, try running these two commands:

```
# apt-get -f install
# dpkg --configure -a
```

E tente novamente. Pode ser necessário rodar o segundo comando mais de uma vez. Esta é uma importante lição para os aventureiros que usam o 'unstable'.

Se você receber o erro "E: Dynamic MMap ran out of room" quando estiver rodando `apt-get update`, adicione a seguinte linha no `/etc/apt/apt.conf`:

```
APT::Cache-Limit 10000000;
```

## 9.2 Onde posso encontrar ajuda?

Se você estiver com dúvidas, consulte a extensa documentação disponível para o sistema de empacotamento Debian. `--help's` e páginas de manual pode ser grandes ajudas para você, tais como as documentações contidas nos diretórios `/usr/share/doc`, como `/usr/share/doc/apt`.

Se esta documentação não sanar suas dúvidas, tente procurar por respostas nas listas de discussão Debian. Você pode encontrar mais informações sobre listas específicas de usuários no website do Debian: <http://www.debian.org>.

Lembre-se que essas listas e recursos devem ser usados somente por usuários Debian; usuários de outros sistemas irão encontrar melhor suporte nas comunidades de suas distribuições.

## Capítulo 10

# Quais distribuições suportam APT?

Aqui estão os nomes de algumas das distribuições que usam APT:

Debian GNU/Linux (<http://www.debian.org>) - foi para esta distribuição que o APT foi desenvolvido

Conectiva (<http://www.conectiva.com.br>) - esta foi a primeira distribuição a portar o APT para ser usado com rpm

Libranet (<http://www.libranet.com>)

Mandrake (<http://www.mandrake.com>)

PLD (<http://www.pld.org.pl>)

Vine (<http://www.vinelinux.org>)

APT4RPM (<http://apt4rpm.sf.net>)

Alt Linux (<http://www.altlinux.ru/>)

Red Hat (<http://www.redhat.com/>)

Sun Solaris (<http://www.sun.com/>)

SuSE (<http://www.suse.de/>)

Yellow Dog Linux (<http://www.yellowdoglinux.com/>)



## Capítulo 11

# Agradecimentos

Um muito obrigado a vocês, meus grandes amigos do projeto Debian-BR, e ao Debian, que são uma ajuda constante para mim e me dão sempre a força para continuar trabalhando para o benefício da humanidade, como também me ajudando no meu objetivo de salvar o mundo. :)

Eu também quero agradecer ao CIPSGA pela enorme ajuda que tem dado ao nosso projeto e para todos os projetos livres que afloram de grandes idéias.

E um obrigado especial a:

Yooseong Yang <yooseong@debian.org>

Michael Bramer <grisu@debian.org>

Bryan Stillwell <bryan@bokeoa.com>

Pawel Tecza <pawel.tecza@poczta.fm>

Hugo Mora <h.mora@melix.com.mx>

Luca Monducci <luca.mo@tiscali.it>

Tomohiro KUBOTA <kubota@debian.org>

Pablo Lorenzoni <spectra@debian.org>

Steve Langasek <vorlon@netexpress.net>

Arnaldo Carvalho de Melo <acme@conectiva.com.br>

Erik Rossen <rossen@freesurf.ch>

Ross Boylan <RossBoylan@stanfordalumni.org>

Matt Kraai <kraai@debian.org>

Aaron M. Ucko <ucko@debian.org>

Jon Åslund <d98-jas@nada.kth.se>



## Capítulo 12

# Novas versões deste tutorial

Este manual foi criado pelo projeto Debian-BR (<http://www.debian-br.org>), com o objetivo de auxiliar no uso diário do Debian.

Novas versões deste documento estarão disponíveis na página de Projetos de Documentação Debian, em <http://www.debian.org/doc/ddp>.

Comentários e críticas podem ser enviadas diretamente para mim através do email <kov@debian.org>.