

Instalação de pacotes



Introdução

Qualquer distribuição Linux sempre irá possuir um gerenciador de pacotes, que nada mais é do que um programa que irá instalar (adicionar) programas, listar, remover, etc.

No nosso material iremos separar os exemplos de uso do gerenciamento de pacotes no **Debian** e **Red Hat**, pois elas utilizam gerenciadores de pacotes diferentes, o **dpkg/apt-get** e o **rpm**, respectivamente.





Debian



Consultando os programas (Debian)

O **dpkg** deve sempre checar o sistema para saber todo pacote que está nele. Isso seria feito no sistema através do seguinte comando:

```
# dpkg -l | more
```

Onde:

-l – Listar todos os pacotes.

more – Paginando a cada tela cheia.

A saída desse comando irá informar quais programas estão instalados.



Consultando os programas

Para verificar se um pacote está instalado, usamos o **grep** para filtrar. Vamos verificar se o programa **BZIP2** está instalado:

```
# dpkg -l | grep bzip2
```

Não retornando resposta nenhuma, o pacote não está instalado, então partimos para a instalação do mesmo.



Buscando os programas

Antes de adicionar o pacote no sistema, precisamos dizer para o dpkg obter o mesmo através de uma fonte, por exemplo o CD-ROM de instalação da distro. Por exemplo, vamos instalar pacotes através do CD de Instalação.

Colocamos o CD 1 do Debian:

```
# mount /cdrom
```

O diretório onde ficam armazenados todos os pacotes disponíveis no CDROM é:

```
# cd /cdrom/pool/main
```

```
# ls
```

O Debian trabalha de forma organizada, onde no CD dentro do diretório **main** você terá diretórios em ordem alfabética.



Buscando os programas

Vamos exemplificar essa parte da aula instalando o programa **bzip2**. Então teremos que entrar no diretório **b** para verificar se o mesmo está disponível nesse CD, caso contrário teremos que acessar o segundo CD da Distro, e assim sucessivamente.

```
# pwd
```

```
/cdrom/pool/main/
```

```
# cd b/
```

```
# ls
```

```
base-passwd bibtool binutils bonobo bzip2
```

```
# cd bzip2/
```

```
# ls
```

```
bzip2_1.0.2-1_i386.deb libbz2-1.0_1.0.21_i386.deb
```



Informações sobre o programa

OBS: Para instalar pacotes no sistema é importante lembrar que temos que estar com o usuário root.

Para adicionar um programa (instalá-lo), é sempre bom dar uma lida na informações dele. Fazemos isso no nosso exemplo, com o bzip2:

```
# dpkg -I bzip2_1.0.2-1_i386.deb
```

Package: bzip2

Version: 1.0.2-1

Architecture: i386

Depends: libbz2-1.0, libc6 (>= 2.2.4-4)

Replaces: libbz2(<<0.9.5d-3)

Installed-Size: 405

Maintainer: Philippe Troin <phil@fifi.org>

Description: A high-quality block-sorting file compressor –
utilities bzip2 is a freely available....



Dependências dos programas

Você viu que no slide anterior que há uma linha mais importante, que está destacada em vermelho, que seriam as dependências desse pacote...

Uma dependência nada mais é do que outro programa que já deveria estar instalado para que, no nosso caso, o bzip2 possa ser instalado. Se baseando no bzip2, temos que verificar antes de instalá-lo se suas dependências já estão instaladas no Sistema.

```
# dpkg -l | grep libbz2-1.0
```

```
# dpkg -l | grep libc6 ii libc6 2.3.2.ds1-20 GNU/C-Library:Shared  
libraries and Timezone
```

OBS: Caso ainda não esteja instalada, você deverá instalar!!!



Instalando um programa

Depois de consultar as informações sobre o programa e suas possíveis dependências, agora podemos instalar via dpkg o programa **bzip2**:

```
# dpkg -i bzip2_1.0.2-1_i386.deb
```

```
Selecting previously deselected package bzip2.
```

```
(Reading database ... 40878 files and directories currently installed.)
```

```
Unpacking bzip2 (from bzip2_1.0.2-1_i386.deb) ...
```

```
Setting up bzip2 (1.0.2-1) ...
```

Onde:

i – install ou instalar!!!



Outras opções do dpkg

Uma opção interessante do dpkg é consultar a qual pacote pertence um determinado arquivo (binário, arquivo de configuração) do sistema:

```
# dpkg -S /etc/issue
```

OBS: Não são todos os arquivos que ele faz referência ao pacote, apenas os arquivos que estão em sua base de dados, então não se assuste caso ele não consiga encontrar o pacote de um determinado arquivo.

Para verificar o status de um pacote (através do nome dele, verificado no comando **dpkg -l**) no sistema:

```
# dpkg -s passwd
```



Outras opções do dpkg

Outra opção interessante do dpkg é mostrar todos os arquivos que o programa trará quando for instalado no sistema:

```
# dpkg -L base-files
```

Também posso saber os arquivos existentes em um programa, mesmo sem tê-lo instalado no sistema:

```
# dpkg -c xterm_4.2.1-16_i386.deb
```

OBS: Nessa opção precisamos passar como parâmetro o pacote .deb, pois o mesmo ainda não foi instalado em nosso sistema.



Outras opções do dpkg

Para remover um pacote do sistema:

```
# dpkg -r dhcp-client
```

OBS: Caso esse pacote que está sendo removido for dependente de outro pacote, você deverá primeiro remover a sua dependência!

OU:

```
# dpkg --purge dhcp-client
```

OBS: Recomendamos sempre que você for desinstalar um pacote sempre use a opção **--purge** para que ele possa remover também os arquivos de configuração desse pacote, não deixando “lixo” no sistema.



APT-GET

O apt-get é uma ferramenta poderosa e prática de usar. Ele está disponível também no **Ubuntu**, **Kurumin** e em **outras distribuições baseadas no Debian**,

Essa ferramenta usa um conceito de fontes de atualização, podendo obter pacotes de qualquer lugar, como **CD-ROMs** do Debian, **unidades de rede** etc.

O meio mais usado é baixar os pacotes via internet, via **http** ou **ftp**, o que permite obter sempre as versões mais novas dos programas.

O apt-get se baseia em um arquivo chamado **/etc/apt/sources.list** onde ficam os repositórios disponíveis. Adicionamos ou removemos fontes nesse arquivo.

Exemplo de /etc/apt/sources.list:

```
#deb http://ftp.debian.org/debian etch main contrib non-free
```

```
# Debian Updates de segurança
```

```
deb http://security.debian.org/ etch/updates main contrib
```

```
deb-src http://security.debian.org/ etch/updates main contrib
```



APT-GET

Depois do comando **apt-get update** fazer seu trabalho, você pode começar a instalar os programas, usando o comando **apt-get install**, seguido do nome do pacote que você quer instalar.

Para instalar o **Firefox**, por exemplo, use o comando **apt-cache search** mais a palavra chave do pacote para obter o nome correto do pacote, nem sempre o nome do pacote é só o nome do programa:

```
# apt-cache search firefox
```

```
firefox - lightweight web browser based on Mozilla
```

```
firefox-locale-pt-br - Portuguese (Brazil) language package for Firefox
```

Irá aparecer tudo relacionado ao **Firefox**, para instalar:

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

```
# apt-get install firefox-locale-pt-br (Para ficar em português do Brasil)
```



APT-GET

O **apt-get** cuida de toda a parte chata, que é resolver dependências.

Dependências são programas ou bibliotecas necessárias para que um outro programa funcione corretamente. E isso facilita a vida do administrador!

É possível também remover pacotes instalados, neste caso usando o parâmetro "**remove**", como em:

```
# apt-get remove firefox
```

O cache do apt-get

O apt-get salva todos os pacotes baixados em **/var/cache/apt/archives**, para um possível uso posterior.



APT-GET

No Debian, é muito comum fazer **backup** desse diretório para manter o **cache** do apt-get depois de reinstalar o sistema, ou mesmo instalar os pacotes já baixados em outras máquinas. Mas isso ocupa muito espaço no HD depois de um certo tempo.

Para limpar o diretório **/var/cache/apt/archives/**, use o comando:

```
# apt-get clean
```

Aprofundaremos esse assunto em chat :-)





Instalando um programa

Para instalar um programa no Red Hat via o utilitário **rpm**, podemos usar o seguinte comando:

```
# rpm -ivh bzip2*
```

Onde:

i – install ou instalar

v – verbose (Mostra em detalhes a instalação)

h – hash (Mostra a barra de progresso “#####”)



Outras opções do rpm

Listando todos os pacotes instalados no sistema:

```
# rpm -qa | more
```

Onde:

q – query, ou seja, procurar, pesquisar

a – all , todos os pacotes

Listando um pacote específico:

```
# rpm -qa | grep dhcp
```

Onde:

grep dhcp – filtrando apenas os pacotes que possuam a palavra dhcp.



Outras opções do rpm

Verificando os arquivos de um determinado pacote instalado no sistema:

```
# rpm -qi passwd
```

OBS: Aqui nessa opção o nome do pacote deverá ser igual ao retornado pelo comando **rpm -qa**.

Verificando os arquivos de um determinado pacote que não está instalado no sistema:

```
# rpm -qi xterm_4.2.1-16_i386.rpm
```

OBS: Nessa opção precisamos passar como parâmetro o pacote .rpm, pois o mesmo ainda não foi instalado em nosso sistema. Ou seja, é um pacote do CD de instalação.



Outras opções do rpm

Removendo um programa do sistema:

```
# rpm -e dhcp-client
```

OBS: Caso esse pacote que está sendo removido for dependente de outro pacote, você deverá primeiro remover a sua dependência.

Simulando a Remoção de um pacote do sistema:

```
# rpm -e dhcp-client --test
```

OBS: Essa opção **--test** também pode ser usada para simular a instalação de um pacote no sistema. Usamos essa opção para chegar dependência de um determinado pacote.



BIBLIOGRAFIA

Linux – Guia do Administrador do Sistema

Autor: Rubem E. Pereira

Editora: Novatec

Manual Completo do Linux (Guia do Administrador)

Autor: Evi Nemeth, Garth Snyder, Trent R. Hein

Editora: Pearson Books

Guia Foca GNU/Linux

<http://focalinux.cipsga.org.br/>

