Capítulo 6 - Gerenciamento de pacotes (instalação e remoção de programas)

6.1. Objetivos:

Entender como gerenciar a instalação e remoção de pacotes no Linux

6.2. Gerenciador de pacote dpkg

O dpkg é um programa de computador que é a base do sistema de gerenciamento de pacotes para as distribuições GNU/Linux baseadas em Debian. O dpkg é uma ferramenta em linguagem de baixo nível. Front-ends de alto nível são exigidos para buscar pacotes em lugares remotos ou ajudar no solucionamento de conflitos nas dependências dos pacotes. Para essa finalidade o Debian fornece o aptitude. O apt, largamente conhecido, não deve ser utilizado com este propósito uma vez que é um protótipo e não realiza o gerenciamento de pacotes tão bem quanto o aptitude.

Outro conceito importante é o de dependência que nada mais é que outro programa, ou seja, outro pacote que precisa já estar instalado na máquina para você instalar o pacote que precisa. O dpkg trabalha com pacotes com extensão .deb. Antes de instalar um pacote no GNU/Linux, tenho que verificar se o mesmo já está instalado. E para isso, eu uso o comando:

```
# dpkg -l
```

Onde esse comando mostra todos os pacotes instalados no sistema. Mas, se estou procurando um programa específico e não quero ver toda lista de programas instalados, melhoro o comando fazendo um filtro na pesquisa:

```
# dpkg -l | grep eject
ii eject 2.0.3-1 ejects CDs and operates CD-Changers under Li
```

Aparecendo o ii significa que o mesmo está instalado.

Caso não retorne nada no comando “dpkg -l | grep pacote” tenho que instalar. Vamos supor que esse pacote não está instalado.

Nosso próximo passo então é instalar esse pacote, e para isso preciso obtê-lo, muitos pacotes se encontram no próprio cd/dvd de instalação do Debian, para isso vamos montar o CD:

```
# mount /dev/hdb /media/cdrom
```

Obs: Nesse exemplo minha leitora de cd é representada por /dev/hdb, isto é, está ligada na minha IDE1 como slave.

Depois que eu monto o CD, verifico se o mesmo foi montado:

```
# df -h
```

ou

```
# mount
```

Depois de montado o cd, acesso o diretório de pacotes do CD:

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

Dê um ls nesse diretório:

```
# ls /media/cdrom/pool/main/
```

Nesse diretório vocês podem ver que o Debian separa os pacotes em ordem alfabética! Como o pacote que eu quero instalar é o eject, entro no diretório “e”. Lá terá todos os pacotes com a letra e que estão nesse primeiro CD de instalação.

```
# cd e
# ls
```

Agora hora tão esperada! Vamos instalar o pacote:

```
# dpkg -i eject*.deb
```

Onde * é o restante do nome, que você deve colocar exato ou deixar com o * se não quiser digitar tudo. E o -i seria o install, ou seja, instalar!

Apenas preciso montar o CD, acessar o diretório onde está o pacote .deb e usar a opção -i. Você pode obter o pacote .deb pela internet também. Mas veremos como fazer com o aptitude que é automatizado, isto é, mais fácil e prático.

Para remover o pacote, basta usar a opção -r no lugar do -i:

```
# dpkg -r eject
```

Uma coisa interessante para se falar nessa parte de remover pacotes: É interessante usar a opção “--purge” ao invés de “-r”. Pois a opção “--purge” além de remover o pacote, remove os arquivos de configuração desse pacote caso o mesmo possua! Ou seja, ela remove limpando tudo:

```
# dpkg --purge eject
```

Atenção: *Tanto para instalar, quanto para remover um pacote se o mesmo tiver uma dependência você terá que primeiro resolver essa dependência para depois instalar ou remover o programa em si.*

Se quero verificar onde ficarão os arquivos de um novo pacote a ser instalado no sistema e lembrando que darei esse comando para um pacote .deb, pois ele não foi instalado ainda, o parâmetro a ser usado é o “-L”:

```
# dpkg -L nome_pacote.deb
```

Então, eu consigo ver onde serão gravados todos os arquivos que esse pacote precisa na hora da instalação! Para conhecer mais sobre o dpkg, man:

```
# man dpkg
```

6.3. APT-GET (aptitude) - Advanced Packaging Tool

Aptitude (apt-get)

O apt é sistema de gerenciamento de pacotes de programas que possui resolução automática de dependências entre pacotes, método fácil de instalação de pacotes e permite atualizar facilmente sua distribuição.O apt-get ou o aptitude funciona através de linha de comando mas mesmo assim sua operação é muito fácil! O aptitude é um front-end do apt-get, isto é, possui uma interface em modo texto de uma forma amigável, mas funciona no mesmo esquema do apt-get. Vamos usar o aptitude pois o Debian na versão 4.0 foi baseado nele, além de que, o aptitude além de resolver uma dependência, ele também remove a dependência de um pacote caso você o remova, exceto se essa dependência for dependência de outro pacote. Eu tenho 2 opções para instalar pacotes pelo aptitude: Internet e - CD/DVD de instalação. E quando falo Internet, pode ser por FTP ou HTTP.

A primeira coisa que temos que fazer para usar o aptitude é verificar se o seu arquivo de mirror (espelho) está configurado corretamente. Inicialmente, vamos configurar nosso APT para instalar a partir do cdrom/dvd.

Mirrors são endereços que armazenamos em um arquivo, para que o apt-get e o aptitude possam saber onde procurar pelos pacotes. O arquivo que armazena os mirrors se chama sources.list e fica em /etc/apt.Ou seja é nesse arquivo sources.list que ele vai buscar a fonte para instalar os pacotes que queremos! Então para apagar o conteúdo desse arquivo:

 # echo "" > /etc/apt/sources.list

Aqui, vamos começar do zero que seria o arquivo vazio, e vou indicar como vou instalar pacotes aqui nesse primeiro momento pelo CD! Agora temos que adicionar um mirror, pois o arquivo agora está vazio. Como decidimos que iremos instalar pelo CD de instalação ou DVD, colocamos o CD 1 no drive e executamos o comando:

 # apt-cdrom add

Ou seja, adicione o CD 1 como um mirror.

E podemos repetir esse comando para os vários cds ou dvds que o Debian tem.

Isso irá criar as linhas do CD (caso você tiver os 3 primeiros cds iniciais):

```
Binary 1--> equivale ao CD 1
Binary 2--> equivale ao CD 2
Binary 3--> equivale ao CD 3
```

E ele vai pedir exatamente assim, quando for instalar algum pacote diferenciando os cds. Para para verificar o conteúdo do arquivo sources.list, faça:

```
# cat /etc/apt/sources.list
```

Para atualizar a base de dados do apt:

```
# aptitude update
```

Com esse comando, ele irá criar a base com a lista de todos os pacotes que tem disponível no seu source (fonte, mirror), seja ele pela NET ou pelo CD. Sendo assim, sempre que mexer no arquivo sources.list será necessário atualizar a base do aptitude! Depois que atualizamos a base do aptitude, podemos instalar o pacote com o comando:

```
# aptitude install nome_do_pacote
```

Então, só para fixar: Depois que configurei o aptitude para instalar pelo cdrom e atualizei a sua base, posso instalar os pacotes com a opção install como mostrei acima. Vamos a um exemplo:

```
# aptitude install vim
```

Para remover:

```
# aptitude remove vim
```

Ou seja, basta trocar o install por remove.

O comando para fazer uma atualização dos pacotes que estão no sistema para uma versão mais nova que tenha no seu mirror é:

```
# aptitude upgrade
```

Mas a atualização vai depender muito de quais versões tem no seu mirror. No caso do mirror ser o cdrom, não vou atualizar nada, afinal de contas sempre vão ser os mesmos pacotes! Para atualizar o sistema é interessante obter os pacotes de um mirror da Internet. Para atualizarmos o sistema tempos também o seguinte comando:

```
# aptitude dist-upgrade
```

A diferença desse último comando para o anterior é que o último atualiza todos os pacotes sem exceção enquanto que o “aptitude upgrade” só atualiza pacotes que não precisam atualizar as dependências.

O aptitude sempre irá lhe falar a ação que ele vai tomar e você decide se aceita ou não o que ele vai fazer. Por exemplo: ele pode falar que para instalar o xyz, ele precise instalar a biblioteca libxxx, mas também remover o pacote yyy pois ele conflita. Então, ele vai perguntar o que pode ser feito ou não e você decide.

Uma coisa muito importante é entender como o Debian funciona nessa questão de versões.Sempre temos: versões stable, testing e unstable. Mas qual a diferença entre elas?

stable *- Versão ATUAL TOTALMENTE HOMOLOGADA E TESTADA PELA EQUIPE. Em contra partida, com o tempo acaba se tornando um pouco desatualizada na questão de versões de aplicativos e recursos.*

testing *- A próxima VERSÃO do Debian que está sendo TESTADA para se tornar STABLE. Que tem versões mais atuais, mas não sempre as últimas (é relativo)*

unstable *- Versões mais NOVAS! Porém quase ninguém testou no Debian ainda, isso não significa que não funcione ou tenha BUG, mas sim que a Debian ainda não deu um OK.*

Sendo que a escolha de stable, testing e unstable, sempre vamos optar no MIRROR no arquivo sources.list!

Exemplo com o mesmo Mirror Oficial da Debian:

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

# Testing:
deb http://ftp.debian.org/debian testing main contrib non-free

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

O Debian também utiliza codinomes para versões, muitos baseados no filme Toy Story. A versão atual do Debian estável é conhecida também como lenny. A versão testing é conhecida como squeeze. Isso significa que você pode usar o mirror da seguinte maneira também:

```
Stable:
deb http://ftp.debian.org/debian lenny main contrib non-free

Testing:
deb http://ftp.debian.org/debian squeeze main contrib non-free
```

Por questões ideológicas, o Debian só instala pacotes que são software livre por padrão. Mas podemos instalar pacotes não livres no Debian se quisermos? Sim. Quando você usa a opção non-free no mirror você ativa para que ele busque pacotes não livres também. Caso você não encontre um pacote não livre no mirror oficial do Debian, você pode usar o site: http://www.apt-get.org Lá tem todos os mirrors não oficiais que o aptitude tem!

E quando falo não oficiais, é que terá praticamente todos os pacotes do GNU/Linux, porém alguns o Debian não traz por padrão por questões ideológicas como dito anteriormente. Mas aí, basta fazer uma pesquisa pelo pacote nesse site e pegar esse endereço (mirror) e colocar no arquivo e atualizar a base de dados do aptitude. Devo copiar o mirror do jeito que está lá! Ou seja, tenho que copiar a linha exatamente igual que aparece no site!

Dica: Se você escolher instalar pela internet, é melhor comentar as linhas no arquivo que diz respeito ao cdrom, pois ele sempre terá preferência!

Mas vamos pensar juntos. Quando você usa o apt para instalar pacotes, esses pacotes precisam ser baixados para a máquina local certo? E onde ficam esses pacotes? Quando você pede para o aptitude instalar pacotes ele guarda o mesmo no diretório **/var/cache/apt/archives**. E os pacotes .deb apenas ocupam espaço em HD, então para tirar esses pacotes:

```
# aptitude clean
```

O comando acima limpa esse diretório e libera mais espaço no seu HD! A vantagem de ele guardar os .deb é para que caso no futuro você precise reinstalar o pacote ele não precisará baixá-lo novamente, pois estará nesse diretório. Também é importante falar, que a idéia de se colocar um mirror da NET é tentar usar as versões mais novas dos aplicativos. Vamos imaginar que quero instalar um cliente IRC no sistema, mas não sei o nome do pacote, então eu uso a opção search do aptitude para localizar nomes de arquivos:

```
# aptitude search irc
```

•p cgiirc - web based irc client

•p jabber-irc - IRC transport for jabber

•i ksirc - IRC client for KDE

•p pidgin - IRC client

Esse “p” antes do nome do pacote significa que o pacote ainda não está instalado. O “i” significa que o pacote já está instalado.

6.4. Gerenciamento de pacotes no Red Hat

rpm

O RedHat Package Manager (RPM) é um sistema de gerenciamento de pacotes para sistemas GNU/Linux baseados em RedHat. Ele instala, atualiza, desinstala e verifica softwares. Originalmente desenvolvido pela Red Hat Linux, é agora usado por muitas distribuições como Novell Suse que possui sua própria versão de RPM, Fedora, CentOS etc.

Assim como o dpkg no Debian, o rpm não resolve dependências automaticamente. Para resolver as dependências automaticamente você pode usar o gerenciador de pacotes yum, que é o aptitude do RedHat e distribuições derivadas.

Para verificar quais pacotes estão instalados no sistema:

 # rpm -qa

E como eu faço para verificar se o pacote xyz está instalado?

 # rpm -q xyz

Para verificar o que vai ser instalado junto com o pacote xyz:

 # rpm -qp xyz<versao>.rpm

Para verificar informações do pacote, não instalado, xyz:

 # rpm -qpi xyz<versao>.rpm

Para verificar se a instalação irá ocorrer corretamente:

```
# rpm -ih –test –percent xyz<versao>.rpm
```

As opções “-h' e “--percent” servem para mostrar uma barra de progressos e a porcentagem concluída. Quais parâmetros eu uso para realizar instalação o programa vlock?

```
# rpm -qi vlock<versao>.rpm
```

Para instalar o pacote xyz sem suas dependências:

```
# rpm -ih –-percent –-nodeps xyz<versao>.rpm
```

Verifique o que será efetuado ao removermos o pacote xyz:

```
# rpm -e –test xyz
```

Agora remova o vlock:

```
# rpm -e vlock
```

Para realizar uma atualização de versão de algum programa podemos utilizar o comando:

```
# rpm -Uh <pacote>.rpm
```

Verifique a integridade de todos os pacotes instalados no sistema:

```
# rpm -Va
```

6.5. Gerenciamento de pacotes com o yum

yum

Yum significa “Yellow dog Update, Modified) e é o gerenciador de pacotes que faz instalação e remoção de pacotes em distribuições derivadas de Red Hat através de um mirror na internet como o aptitude para o Debian.

O Yum trabalha com pacotes com a extensão .rpm. Assim como o aptitude do Debian, o yum tem seu arquivo de configuração que permitite especificar os repositórios que contém os pacotes para serem instalados e resolve dependências automaticamente.

O arquivo de configuração do yum é o /etc/yum.conf. Além disso, os repositórios são configurados através de cada arquivo de extensão .repo, localizados no diretório /etc/yum.repos.d.

Por exemplo, o padrão para o repositório do Fedora é o arquivo /etc/yum.repos.d/fedora.repo, para o CentOS são os arquivos “CentOS-Base.repo" e "CentOS-Media.repo", no qual o primeiro inclui os repositórios oficiais da distribuição e o segundo permite que você instale pacotes contidos nos CDs (ou no DVD) de instalação.

Exemplo de um arquivo .repo:

```
[fedora]
name=Fedora $releasever - $basearch
baseurl=http://fedora.c3sl.ufpr.br/linux/releases/
$releasever/Everything/$basearch/os/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora file:///etc/pki/rpm-
gpg/RPM-GPG-KEY
```

Explicando os parâmetros do arquivo:

•**name**: Descrição: Fedora e versão da distribuição - (arquitetura da distribuição, como, por exemplo: i386);

•**baseurl**: O endereço que contém a lista dos programas e os pacotes;

•**enabled**: Se o repositório está habilitado ou não (1 significa sim, 0 significa não);

•**gpgcheck**: Se todos os pacotes devem ter sua autenticidade verificada (1 significa sim, 0 significa não);

•**gpgkey**: Especifica qual chave criptográfica utilizar para a verificação dos pacotes.

Para instalar um pacote:

```
# yum install nome_do_pacote
```

Para removê-lo, use:

```
# yum remove nome_do_pacote
```

O yum possui um recurso de busca, que é útil quando você está procurando por um pacote, mas não sabe o nome exato:

```
# yum search nome_do_pacote
```

Um relatório será apresentado mostrando todos os pacotes relacionados, incluindo o texto de descrição de cada um. Para fazer uma busca mais restrita, procurando apenas nos nomes dos pacotes, use o parâmetro "list", como em:

```
# yum list httpd
```

Ele retornará apenas os pacotes que possuem "httpd" no nome. Para atualizar um pacote já instalado, use o comando "yum update", como em:

```
# yum update nome_do_pacote
```

Observação: Após atualizar um serviço, é necessário recarregar o serviço (como o comando service) para que a nova versão passe a ser utilizada.

Para atualizar todo o sistema, comece usando o parâmetro "check-update", que lista as atualizações disponíveis:

```
# yum check-update
```

Se usado sem especificar um pacote, o "update" vai atualizar de uma vez só todos os pacotes do sistema, de forma similar ao "aptitude dist-upgrade" do Debian:

```
# yum update
```

6.6. Compilando um programa disponibilizado como tar.gz ou tar.bz2.

A instalação a partir do código fonte é feita de forma padronizada, conforme as convenções da comunidade. Vamos usar como exemplo um programa fictício chamado 4linux-1.0.tar.gz.

Geralmente pode-se obter um código fonte pelo site do seu desenvolvedor. O código fonte geralmente é fornecido compactado ou por .gz ou bz2.

```
# tar zxvf 4linux.1-0.tar.gz -C /usr/src
```

ou

```
# tar jxvf 4linux-1-0.tar.bz2 -C /usr/src
```

Descompactado o arquivo vai ser criado um diretório, usualmente com o mesmo nome do arquivo, no nosso caso "4linux-1.0" dentro do diretório /usr/src.

```
# cd /usr/src/4linux-1.0
```

Provavelmente existem dois arquivos - README e INSTALL - que trazem informações genéricas sobre o pacote e informações sobre a instalação. É altamente recomendável que se leia esses arquivos para saber o funcionamento do software pois essa é a principal vantagem do software livre.

Qual é o primeiro passo para realizar a compilação deste pacote?

O primeiro passo da compilação geralmente é rodar um script chamado configure.

Este script, que é gerado pelo desenvolvedor do programa com o autoconf, examina o seu sistema na busca por bibliotecas e arquivos de configuração e executáveis necessários para a compilação do programa.

Se tudo estiver OK ele gera um arquivo chamado Makefile, que será usado posteriormente pelo make. Se alguma dependência não for encontrada ele pára e mostra uma pequena mensagem de erro, indicando o que ocorreu ou qual arquivo estava faltando. Na maioria das vezes o configure executa sem problemas.

```
# ./configure
```

O próximo passo é a compilação em si. A compilação é coordenada pelo make, que segue um roteiro definido no Makefile, compilando e gerando os arquivos binários usando o gcc.

Qual comando usamos para que essa compilação seja feita?

```
# make
```

Terminada a compilação - sem nenhum erro - é hora de instalarmos o novo programa. Como realizamos agora a instalação do programa?

```
# make install
```

Dica: Vamos supor que você esteja usando o Debian, mas encontrou um pacote para Red Hat e gostaria de converter o rpm para deb, você pode fazer isso e vice-versa com um pacote chamado ***alien****.*

```
# aptitude install alien
```

Convertando um pacote rpm para deb:

```
# alien -d pacote.rpm
```

Convertendo um pacote deb para rpm:

```
# alien -r pacote.deb
```

Atenção:
O uso do alien só faz sentido com pacotes binários;
O pacote que o alien gera pode não seguir as regras de colocação de arquivos e diretórios da sua distro (lembre-se que algumas vezes os locais de arquivos mudam de uma distro para outra);
Só use o alien quando não houver outra alternativa, tente primeiro achar o programa para a sua distribuição.