Comandos no Linux

Hugo Cisneiros, hugo_arroba_devin_ponto_com_ponto_br
Última atualização em 04/02/2003

Comandos para manipulação de arquivos

A primeira coisa que sempre vem em mente no uso de um sistema operacional é como lidar com os arquivos dentro dele... Nesta seção eu vou mostrar alguns comandos básicos para mexer com os arquivos.

- **cd** - Navegando entre diretórios
- **ls** - Listar arquivos
- **mkdir** - Cria um diretório
- **rmdir** - Remove um diretório vazio
- **cp** - Cópia de arquivos e diretórios
- **mv** - Move arquivos e diretórios
- **rm** - Deleta arquivos e diretórios
- **ln** - Linkando arquivos
- **cat** - Exibe o conteúdo de um arquivo ou direciona-o para outro
- **file** - Indicando tipo de arquivo

cd - Navegando entre diretórios

```
cd [nome_do_diretório]
```

Este comando acima mudará o diretório atual de onde o usuário está. Há também algumas abreviações de diretórios no Linux para a facilitação, estes são:

Abreviação	*Significado*
. (ponto)	Diretório atual
.. (dois pontos)	Diretório anterior
~ (til)	Diretório HOME do usuário
/ (barra)	Diretório Raiz
- (hífen)	Último diretório

Por exemplo, se eu quero ir para o meu diretório home, faço o seguinte:

```
$ pwd
/usr/games
$ cd ~
```

```
$ pwd
/home/hugo
```

Ou seja, eu estava no diretório /usr/games, e com um simples cd para o diretório ~, fui para o meu diretório home (/home/hugo). Quando você deseja saber o caminho completo do diretório em que você está, utilize o comando pwd. Se você deseja ir para um diretório que está na raiz diretamente, você usa o / antes, exemplo:

```
$ pwd
/usr/local/RealPlayer7/Codecs
$ cd /etc/rc.d
$ pwd
/etc/rc.d
$ cd -
$ pwd
/usr/local/RealPlayer7/Codecs
```

Eu estava no diretório /usr/local/RealPlayer7/Codecs e quis ir para o diretório etc/rc.d que está na raiz. Note depois que eu usei o hífen e fui de volta para o último diretório em que eu estava.

ls - Listar arquivos

```
ls [opções] [arquivo/diretório]
```

Este comando lista os arquivos, nada mais que isso. Se você executar apenas o ls sozinho, ele vai mostrar todos os arquivos existentes no diretório atual. Há também as opções extras:

Parâmetro	*Significado*
-l	Lista os arquivos em formato detalhado.
-a	Lista os arquivos ocultos (que começam com um .)
-h	Exibe o tamanho num formato legível (combine com -l)
-R	Lista também os subdiretórios encontrados

Exemplo de uma listagem detalhada:

```
$ ls -l
total 9916
drwxrwxr-x   5 hugo hugo    1302 Aug 16 10:15 CursoC_UFMG
-rw-r--r--   1 hugo hugo  122631 Jul 12 08:20 Database.pdf
-rw-r--r--   1 hugo hugo 2172065 Jul 12 08:20 MySQL.pdf
-rw-r--r--   1 hugo hugo 2023315 Jul 12 08:20 PHP3.pdf
```

Podemos também usar no ls os wildcards, ou seja, caracteres que substituem outros. Exemplo: eu quero listar todos os arquivos que têm a extensão .txt, faço o seguinte:

```
$ ls *.txt
debian-install.txt  manualito.txt  named.txt          plip.txt
seguranca.txt
ipfw.txt            mouse.txt      placa_de_video.txt  rede.txt  sis.txt
```

O wildcard é o "*", que representa "tudo".txt. Existem outros wildcards, exemplo disso é o ponto de interrogação (?), que substitui apenas 1 caractere, exemplo:

```
$ ls manual?.txt
manual1.txt  manual2.txt  manual3.txt  manualx.txt  manualP.txt
```

Existe outro wildcard, que envolve os colchetes. Por exemplo:

```
$ ls manual[3-7].txt
manual3.txt  manual4.txt  manual6.txt  manual7.txt
```

Lista todos os arquivos que tiverem como manual?.txt, onde o ? pode ser substituído por 3, 4, 5, 6 e 7.

mkdir - Cria um diretório

```
mkdir <nome_do_diretório>
```

Cria um diretório. Exemplo:

```
$ mkdir ~/paginas
```

Este comando criará o diretório paginas no seu diretório home.

rmdir - Remove um diretório vazio

```
rmdir <nome_do_diretorio>
```

Apaga um diretório que esteja vazio. Exemplo:

```
$ rmdir /tmp/lixo
```

Isto apagará o diretório /tmp/lixo apenas se ele estiver vazio. Para apagar um diretório com seu conteúdo, refira-se ao comando rm.

cp - Cópia de arquivos e diretórios

```
cp [opções] <arquivo_origem> <arquivo_destino>
```

O comando cp copia arquivos e diretórios. Como opções dele, podemos ver:

Parâmetro	*Significado*
-i	Modo interativo
-v	Mostra o que está sendo copiado
-R	Copia recursivamente (diretórios e subdiretórios)

Exemplos:

Quero copiar brasil.txt para livro.txt, com a opção de modo interativo.

```
$ cp -i brasil.txt livro.txt
cp: sobrescrever `livro.txt'?
```

Como o arquivo livro.txt já existia, ele pergunta se quer sobrescrever, responda y(sim) ou n(não). Agora eu quero copiar o diretório /home/ftp com tudo dentro (até seus subdiretórios) para /home/ftp2, faço o seguinte:

```
$ cp -R /home/ftp /home/ftp2
```

mv - Move arquivos e diretórios

```
mv <arquivo_origem> <arquivo_destino>
```

Este comando simplesmente move algum arquivo para outro lugar. Ele também é usado para renomear um arquivo. Por exemplo, se eu quero renomear o industria.txt para fabrica.txt, eu faço o seguinte:

```
$ mv industria.txt fabrica.txt
```

Se eu quiser mover o industria.txt para /home/usuario com o mesmo nome, faço:

```
$ mv industria.txt /home/usuario
```

rm - Deleta arquivos e diretórios

```
rm [opções] <arquivo>
```

Este comando apaga definitivamente o arquivo ou diretório. Exemplo:

```
$ rm arquivo.bin
```

Para apagar um diretório com todo seu conteúdo, usa-se a opção -r, assim:

```
$ rm -r /tmp/lixo
```

ln - Linkando arquivos

```
ln -s <arquivo_origem> <link simbólico>
```

Este comando é usado para gerar links simbólicos, ou seja, que se comportam como um arquivo ou diretório, mas são apenas redirecionadores que mandam seu comando para outro arquivo ou diretório, por exemplo:

```
$ ln -s /manual /home/linux-manual
```

Este comando criará o link /home/linux-manual, se você der um ls -l você verá que o diretório /home/linux-manual está apontando para /manual. Se você ir para o /home/linux-manual, você na verdade estará no /manual, mas como é um link, não há diferença.

cat - Exibe o conteúdo de um arquivo ou direciona-o para outro

```
cat <arquivo>
```

Este comando existe para mostrar o conteúdo de um arquivo, ou para fazer a cópia deste arquivo, ou uma junção. Vejamos um exemplo, se eu quiser mostrar o conteúdo de /home/usuario/contato, eu digito:

```
$ cat /home/hugo/contato
```

Aparecerá o conteúdo do arquivo contato:

```
Hugo Cisneiros
hugo_arroba_devin_ponto_com_ponto_br
http://tlm.conectiva.com.br
```

Este comando pode também servir de direcionador para outro arquivo. Indicadores são usados para isso:

```
Indicador ">" - faz uma cópia, exemplo:
$ cat contato1 > contato2
```

```
Indicador ">>" - Acrescenta um arquivo ao outro, exemplo:
cat contato1 >> contato2
```

O cat pode fazer coisas que nem você imagina, como tocar sons. Para fazer isso é simples, ele direciona o arquivo som para o dispositivo de áudio (que no linux é representado por um arquivo), exemplo:

```
cat som-dumau.au > /dev/audio
```

file - Indicando tipo de arquivo

```
file <arquivo>
```

Este comando identifica o tipo de arquivo ou diretório indicado pelo usuário conforme os padrões do sistema operacional. Há varios tipos de retorno, vamos aqui ver alguns mais importantes:

```
ASCII text      C Program source
directory       ELF-Executable
data            Bourn-again shell-script
```

Apenas um exemplo deste comando:

```
$ file linux.txt
ASCII Text
```

Comandos sobre processos do sistema

- **ps** - Listando processos
- **kill** - Matando um processo
- **killall** - Matando processos pelo nome
- **w** - Lista os usuários logados

ps - Listando processos

```
ps [opções]
```

Quando um programa é executado no sistema, ele recebe um número de identificação, o chamado PID. Este comando lista esses processos executados, e apresenta o PID. Além do PID, ele também mostra o comando executado (CMD) e também o STAT (status atual do processo executado, veja nota abaixo), além de outros.

O status do processo é identificado por letras, aqui segue uma tabela com as definições de cada letra:

Letra	*Definição*
0	Não existente
S	Descansando, fora de funcionamento (Sleeping)
R	Rodando (Running)
I	Intermediando (Intermediate)
Z	Terminando (Zumbi)
T	Parado (Stopped)
W	Esperando (Waiting)

Agora um exemplo para este comando:

```
$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   1120    52 ?        S    Dec25   0:05 init
root         2  0.0  0.0      0     0 ?        SW   Dec25   0:00 [kflushd]
root         3  0.0  0.0      0     0 ?        SW   Dec25   0:00 [kupdate]
root         4  0.0  0.0      0     0 ?        SW   Dec25   0:00 [kpiod]
root      1004  0.0  0.0  10820    48 ?        SN   Dec25   0:00 [mysqld]
root      1007  0.0  0.0   2852     0 ?        SW   Dec25   0:00 [smbd]
hugo      1074  0.0  0.0   1736     0 tty1     SW   Dec25   0:00 [bash]
hugo      1263  0.0  0.0   1632     0 tty1     SW   Dec25   0:00 [startx]
hugo      1271  0.0  0.0   2304     0 tty1     SW   Dec25   0:00 [xinit]
hugo      1275  0.0  2.4   4312  1360 tty1     S    Dec25   0:16 wmaker
hugo      2461  0.0  0.0   1636     0 tty1     SW   07:09   0:00
[netscape]
hugo      9618  0.9  4.9   5024  2688 pts/1    S    09:56   0:06 vim
d03.html
hugo     12819  6.7  6.9   5580  3796 ?        S    10:03   0:13 mpg123
King Diamond - Help.mp3
```

Este parâmetro (aux) fez o ps listar todas as informações sobre todos os processos executados.

kill - Matando um processo

```
kill [-SINAL] <PID>
```

O comando kill é muito conhecido (principalmente pelos usuários do Netscape :)), ele serve para matar um processo que está rodando. Matar? Terminar este processo, finalizar natoralmente! Para matar um processo, temos de saber o PID dele (veja o comando ps), e então executar o kill neste PID. Vamos killar o Netscape:

```
$ ps aux | grep netscape
hugo      2461  0.0  0.0  1636    0 tty1     SW   07:09   0:00
[netscape]
$ kill -9 2461
```

E o processo do Netscape foi morto! Vivas! O sinal -9 significa para forçar e matar natoralmente mesmo. Uma lista de sinais pode ser encontrada com o comando:

```
man 7 signal
```

killall - Matando processos pelo nome

```
killall [-SINAL] <comando>
```

Faz a mesma coisa que o kill, só que a vantagem aqui é que você não precisa saber o PID do processo, e sim o nome. A desvantagem é que se tiver dois processos com o mesmo nome, os dois são finalizados. Seguindo o exemplo do comando kill:

```
$ ps aux | grep netscape
hugo      2461  0.0  0.0  1636    0 tty1     SW   07:09   0:00
[netscape]
$ killall -9 netscape
```

w - Listas os usuários logados

```
w
```

Com este comando, é possível você ver quais usuários estão atualmente logados no seu sistema, além de informações como "O que ele está fazendo", "aonde está fazendo", "desde quando está logado", etc. Vejamos um exemplo aqui da minha máquina:

```
[hugo@songoku hugo]$ w
 10:37am  up 13:45,  4 users,  load average: 0.85, 0.70, 0.71
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU  WHAT
hugo     tty1     -                Mon 8pm 25.00s 34:16   0.09s  -bash
root     tty2     -                10:37am  5.00s  0.27s  0.14s  top
hugo     tty3     -                10:37am 11.00s  0.25s  0.13s  vi
cmpci.c
jim      tty4     -                10:37am 22.00s  0.23s  0.12s  BitchX
```

Comandos de pacotes (instalação/desinstalação/consulta)

O que são pacotes?

No Linux, geralmente os aplicativos vêem em forma de código-fonte, então o usuário tem de baixar e compilar. Os pacotes servem justamente para facilitar o trabalho do usuário, dando a ele um arquivo empacotado com o código já compilado. Existem diversos gerenciadores de pacotes que iremos aprender a usar o básico deles aqui. Tem o RPM (RedHat Package Manager), que é usado por várias distribuições como o Conectiva Linux, Red Hat, SuSE e Mandrake. Também tem o DEB (Debian Packages), muito bom também e usado pela distribuição Debian e Corel Linux (que é baseada no Debian por isso). E temos também o pacoteamento do Slackware (TGZ), que não é tão poderoso como os anteriores, mas quebra galhos também.

Além de empacotar o código-fonte compilado, os gerenciadores de pacotes também armazenam as informações de instalação em um banco de dados, para depois o usuário ter informações sobre a instalação, e para desinstalar o pacote do sistema. E não há apenas pacotes com código-fonte compilado, também há pacotes que contém o código-fonte sem compilar, mas empacotado.

Utilizando o PKGTOOL (Slackware)

Nas distribuições Slackware, é bem simples o gerenciamento de pacotes dele. Os pacotes têm extensão .tgz (diferente de .tar.gz), e além de conter os arquivos, contém scripts de pós-descompactação também. Existe uma interface muito amigável para o gerenciamento dos pacotes .tgz, e se chama pkgtool. Tente executar o pkgtool no console e ver no que dá.

Mas também existem os comandos individuais:

Comando	*O que faz*
installpkg X.tgz	Instala o pacote X.tgz
removepkg X	Desinstala o pacote X
makepkg	Cria um pacote

Utilizando o RPM

Para instalar um pacote, usa-se a opção -i:

```
# rpm -i pacote.rpm
```

Você também pode utilizar as opções -v e -h combinadas com a -i para

atualizar para uma versão mais recente da mesma, você utiliza a opção -U ao invés da -i, exemplo:

```
# rpm -Uvh pacote-atualizacao.rpm
```

Isso irá atualizar os arquivos do pacote. Se você quer retirar o pacote do seu sistema, você utiliza a opção -e, assim:

```
# rpm -e pacote
```

Caso este pacote gere dependências com outros pacotes, e mesmo assim você queira removê-lo, você pode utilizar a opção --force, que como o nome diz, força a remoção do mesmo:

```
# rpm -e pacote --force
```

Agora uma característica muito importante também para o usuário é a capacidade de consulta que o RPM traz. Por exemplo, se você quer listar todos os pacotes instalados no sistema, você utiliza o comando:

```
$ rpm -qa
```

Isto irá gerar a listagem dos pacotes. Veja que a opção -q (query) é a opção de consulta, e seguida de outra letra ela faz tarefas. Combinando o comando anterior com o comando grep, podemos ver se um certo pacote está instalado no sistema:

```
$ rpm -qa | grep BitchX
BitchX-75p3-8cl
```

E se você quer saber informações sobre um pacote? Então usa-se a opção -i. Vejamos um exemplo:

```
$ rpm -qi BitchX
Name        : BitchX                       Relocations: (not
relocateable)
Version     : 75p3                             Vendor: conectiva
Release     : 8cl                           Build Date: qua 16 fev 2000
01:28:59 BRST
Install date: dom 10 set 2000 19:33:23 BRT      Build Host:
mapinguari.conectiva.com.br
Group       : Aplicações/Internet           Source RPM: BitchX-75p3-
8cl.src.rpm
Size        : 2812352                          License: GPL
URL         : http://www.bitchx.org
Summary     : Cliente IRC para o console do Linux
Description :
```

```
O BitchX é um cliente de IRC com suporte a cores para o console
do Linux. Ele incorpora várias características que normalmente
requereriam um script, e a sua interface é mais colorida, e simples
de trabalhar que a do ircII :)
```

Se quisermos ver quais pacotes fazem dependência com um certo pacote, utilizamos a opção -R:

```
$ rpm -qR pacote
```

E para verificar a qual pacote um certo arquivo pertence, utilize a opção -f, assim:

```
$ rpm -qf /diretorio/arquivo
```

Ou o contrário, se você quiser listar todos os arquivos pertencentes à um pacote, faça assim:

```
$ rpm -ql pacote
```

Outros tipos de comandos

Descompactar arquivos

Extensão .tar.gz	tar zxpvf arquivo.tar.gz
Extensão .tar	tar xpvf arquivo.tar
Extensão .gz	gunzip arquivo.gz
Extensão .tar.bz2	bunzip2 arquivo.tar.bz2 ; tar xpvf arquivo.tar
Extensão .bz2	bunzip2 arquivo.bz2
Extensão .zip	unzip arquivo.zip

Compactar arquivos

Empacotar um diretório em .tar	tar cvf diretorio/
Empacotar um diretório em .tar.gz	tar zcvf diretorio/
Compacta um arquivo para .gz	gzip arquivo
Compacta um arquivo para .bz2	bzip2 arquivo

Espaço em disco

df -h	Mostra o espaço em disco das partições montadas

du -hs	Mostra o espaço ocupado pelo diretório atual

Informações do sistema

date	Mostra a data e hora atual
cal	Mostra um calendário
uptime	Mostra quanto tempo seu sistema está rodando
free	Exibe a memória livre, a usada, e os buffers da memória RAM
top	Mostra os processos que mais gastam memória
uname -a	Mostra informações de versão do kernel

Programas (console)

vi	Editor de texto
pico	Editor de texto
pine	Leitor de E-Mail
mutt	Leitor de E-Mail
lynx	Navegador Web
links	Navegador Web