

THE ONE PAGE FOR... SHELL!

Autores: Rudson Ribeiro Alves e Maycon Maia Vitali

Última atualização: 24/09/2008

Shell Commands

O *Bash* é uma ótimo interpretador de comandos para o *Shell*, mas não seria nada sem os muitos programas e aplicativos disponíveis no *Shell GNU/Linux*. Este texto apresenta algumas destas ferramentas.

Algumas abreviações para este texto:

OC: opções mais comuns

[xxx] : parâmetros opcionais

<xxx> : parâmetros obrigatórios

opção mais comuns com `-i` → modo interativo e `-v` → modo *verbose* são apresentados apenas como `-v` e `-i`.

Manipulando arquivos e diretórios

cd [dest] – mudar de diretório. Diretórios especiais: `~` → diretório *\$HOME*, o mesmo que apenas `cd`; `-` → último diretório anterior entrado; `.` → diretório atual; `..` → diretório anterior.

pwd – apresenta o diretório corrente

mv [options] <source> <dest> – move um arquivo ou diretório da fonte (*source*) para um destino (*dest*). OC: `-u` → *update*; `-v`; `-i`.

cp [options] <source> <dest> – copia arquivo/diretório para um destino. OC: `-a` → modo *arquivo*. Faz *backup* recursivo, com *update* de arquivos; `-r`, `-R` → recursivo; `-v`; `-i`.

rm [options] <source> – remove arquivos e diretórios (opção `-rf`). OC: `-r` → recursivo; `-f` → forçar a remoção; `-v`; `-i`.

mkdir [options] <dir_name> – cria um diretório. OC: `-p` → cria diretórios recursivamente; `-v`.

rmdir [options] <dir_name> – remove diretório vazio. OC: `-p` → remove os diretórios recursivamente; `-v`.

less, more, most <file> – são paginadores de arquivos.

cat [options] [file1] [file2] ... – concatena arquivos, lançando o resultado na saída padrão (geralmente tela).

split [options] [file] – quebra um arquivos em várias partes

ls [options] [dir/file] – listar arquivos/diretórios: OC: `-a` → apresenta arquivos escondidos (iniciados por ponto); `-l` → lista informações dos arquivos; `-h` → apresenta o tamanho em kB, MB, GB, ...

vi, vim, gvim, pico, nano, emacs [filename] – editores txt

Pesquisa em arquivos

grep [options] <pattern> [file] – filtra as linhas que casam com um padrão (*pattern*) passado. OC: `-e` → mesmo que usar o *egrep*; `-v` → filtra as linhas que não casam com o padrão passado; `-l` → apresenta o nome dos arquivos que possuem linhas que casam com um determinado padrão; `-R` → recursivamente; `-i` → *case sens*. Ignorado.

```
$ grep 'spool' /etc/passwd
```

filtra as linhas que possuem a palavra *'spool'* em */etc/passwd*.

```
$ grep -lr 'hald' /etc/*
```

busca, recursivamente, por todos as palavra *'hald'*, em todos os arquivos e pastas em */etc/*

sed [options] <commands> [file] – o *sed* é um editor de texto de linha de comando. Veja alguns exemplos abaixo:

```
$ sed '/spool!/! d' /etc/passwd
```

faz o mesmo que o *grep* acima. O */spool/*, filtra a linha que possui a palavra *spool*. O *!* inverte a seleção. Por fim, o *d* deleta todas as linhas selecionadas.

```
$ sed 's/Alfredo/Solange/g' texto.txt
```

substitui todas as ocorrências de *'Alfredo'* por *'Solange'*, que encontrar no arquivo *texto.txt*. Sem *'g'*, apenas a primeira ocorrência em cada linha será substituída.

awk [options] <commands> [file] – O *awk* é um poderoso editor de texto por linha de comando. O *Slackzine* 16 traz um bom resumo do *Piter Punk*, do que este comando pode fazer. Veja alguns exemplos abaixo:

```
$ awk '/spool/' /etc/passwd
```

faz o mesmo que o *grep* anterior.

```
$ ls -la | awk '{ print $6 " " $8 }'
```

imprime o proprietário e o nome dos arquivos/diretórios no diretório corrente.

```
$ ls -la | awk '{ SOMA+=$5 } END { print SOMA}'
```

imprime a soma do tamanho em bytes de todos arquivos/diretórios no diretório corrente.

```
awk '{ WORDS+=NF} END { print WORDS }' /etc/fstab
```

conta o número de palavras em */etc/fstab*

```
awk 'END { print NR }' /etc/fstab
```

conta o número de linhas em */etc/fstab*.

Busca por arquivo/diretórios

find [options] <dir_name> – o *find* permite fazer busca por arquivos que atendem a diversos padrões. Ele pode ser combinado com outros comandos para fazer alterações nos arquivos encontrados.

```
$ find .
```

lista todos os arquivos/diretórios, recursivamente, do diretório atual (*.*)

```
$ find . -type d
```

lista todos os diretórios, recursivamente, do diretório atual (*.*). Para listar apenas os arquivos use *'f'* no lugar do *'d'*.

```
$ find /etc/ -name xinitrc
```

procura pelo arquivo *xinitrc* em */etc/*. Pode-se usar curingas (***, *?*) para filtra a busca.

```
$ find . -name '*~' -exec rm {} \;
```

procura por todos os arquivos terminados por um *~* e executa (*-exec*) o comando *rm* neles. O *{}* é uma referência a lista retornada pelo *find*. O *\;* no final é obrigatório.

updatedb – cria um banco de dados (geralmente em */var/lib/slocate/slocate.db*) de todos os arquivos em seu disco. Este banco é usado pelo *locate* a seguir e deve ser criado como *root*.

locate [file] – procura por um arquivo no disco rígido.

whereis <options> [programname] – procura por um executável, fontes e páginas de manual. OC: `-b` → binários; `-s` → fontes; `-m` → manuais

which <options> [programname] – procura por um executável. Esta busca é feita no *PATH*. Mesmo que *'whereis -b [programname]'*

Contas e grupos

adduser [username] – cria uma conta para o usuário *username*. *adduser* é um *frontend* para o *useradd*, usado para criar uma conta interativamente.

useradd [option] <username> – cria uma conta no sistema.

OC: -c → *string* de comentários; -d <home_dir> → diretório da conta; -g <group> → grupo principal; -G <group1, group2,...> → grupos que o usuário participará; -s <shell> → o interpretador *shell* padrão; -u <id> → a id do usuário.

As strings devem ser protegidas por aspas simples ou duplas quando necessário.

userdel [option] <username> – remove uma conta do sistema. OC: -r → remove recursivamente os arquivos e diretórios em */home/username*.

groupadd [option] <groupname> – cria um grupo no sistema. OC: -g → gid do grupo.

groupdel <groupname> – remove um grupo do sistema.

passwd [username] – troca/define uma *password* para o usuário *username*.

su [option] [-] [username] – mudar de usuário. O padrão é se tornar *root*. Para se tornar outro usuário deve usar o “*username*”

```
$ su - alex
```

se torna o usuário *alex*.

```
$ su -c halt
```

executa o comando *halt* como *root*.

sudo [option] [comand] – executa um comando como super-usuário (*root*).

exit – sai de uma conta. Em um console gráfico, fecha a *shell* corrente.

Gerenciamento de Processo

bg/fg [pid/%jobs] – manda um processo interrompido (CNTRL+Z) para *background/foreground*.

ps <options> – imprime informações sobre os processos em execução no sistema. Um conjunto de flags comumente usadas com o *ps* é “*aux*” → (*all*) mostra todos os processos, de todos os usuários em modo estendido para o usuário.

A coluna *STAT (state)* apresenta os seguintes códigos:

D dormindo sem interrupção (geralmente IO)

R Rodando

S dormindo sem interrupção (aguardando um evento)

T Parado, (por um CNTRL+Z ou *trace*)

Z Processo “zumbi”.

Segundo caractere de status:

< alta-prioridade

N baixa-prioridade

L possui páginas trancadas na memória

s líder de uma seção

l é um *multi-threaded*

+ está em *foreground*.

top [options] – apresenta os processos em um modo interativo, com consumo de CPU, memória, ...

kill [signal] <pid1 pid2 ...> – mata um processo enviando o sinal *signal*.

```
$ kill -9 23445
```

mata o processo 23445 enviando o sinal 9, *SIGKILL*. Ver “man 7 *signal*” para conhecer outros sinais.

killall [signal] <command> – faz o mesmo que o *kill* acima, mas mata o processo pelo nome do comando.

nice <prior> [command] – roda um processo com uma prioridade ajustada (*n*). O valor padrão de um processo é 10. A prioridade pode variar de -20 (maior prioridade) até 19 (menor prioridade)

```
$ nice 12 firefox
```

inicia o *firefox* com baixa prioridade (12).

Permissões de arquivos e diretórios

chmod [options] MODE <file/dir> – muda as permissões de um arquivo/diretório. OC: -R → recursivamente.

MODE tem a forma ‘[*ugo*a]*([-+=]([*rw*Xst]*|[*ugo*]))+’.

*ugo*a → (*u*) usuário, (*g*) grupo, (*o*) outros e (*a*) todos

+ = → adicionar, remover ou manter a permissão

rwXst → (*r*) leitura, (*w*) escrita, (*x*) execução/pesquisa (para diretórios), (*X*) execução/pesquisa somente se for um diretório ou já possui permissão de execução para algum usuário, (*s*) seta o id do grupo em execução.

ugo → (*u*) usuário, (*g*) grupo, (*o*) outros

```
$ chmod +xo-lo loop
```

faz o programa *loop* executável por outros, mas não lido.

chown [options] [oner][:<gourp>] <file/dir> – muda o proprietário e grupo de um arquivo/diretório.

```
# chown root:burning /dev/cdrw
```

passa o dispositivo */dev/cdrw* para o usuário *root* e grupo *burning*.

Outros comandos

zip, bzip2, gzip, compress, ... – alguns compressores de arquivos. Os descompressores geralmente são chamados adicionando o prefixo “*un*” ao comando, ou pela *flag* “-d”.

date – imprime e seta a data do sistema

dmesg – imprime informações de *log* gerados pelo *kernel*

dirname – imprime o diretório base de um *path*

basename – imprime o nome base de um *path*

echo – imprime na saída padrão

```
$ echo -e "\033[31;1mGNU/LINUX\033[m”
```

imprime **GNU/LINUX**

free – mostra o uso da memória, *swap*, *buffers*, ..., do sistema

df – mostra o uso dos discos rígidos e outros dispositivos de armazenagem montados

du <options> [dir] – soma recursivamente os tamanhos dos arquivos em um dado diretório. OC: -h → apresenta em kB, MB, ...; -s → apresenta apenas um resumo da soma.

mount <device> <mountpoint> – monta um dispositivo no sistema

```
$ mount /dev/cdrom /media/cdrom
```

monta o dispositivo */dev/cdrom* em */media/cdrom*

umount <mountpoint> – desmonta um dispositivo.

seq [last]/[frist last]/[frist increm last] – gera uma seqüência de números

```
$ seq 10 2 20
```

imprime 10 12 14 16 18 e 20 na saída padrão.

sleep [time] – dorme por *time* segundos.

sort [options] [file] – ordena as linhas de um texto. OC: -r → ordem reversa, -u → faz um *uniq*.

tar [options] [dir] – empacota/desempacota arquivos. Uso comum:

```
$ tar cvjf backup.tar.bz2 /etc
```

empacota o conteúdo do diretório */etc* em *backup.tar.bz2*. O comando a seguir faz o desempacotamento,

```
$ tar xvfj backup.tar.bz2
```

uniq – remove as linhas repetidas de uma lista

wc [options] [file] – conta palavras e caracteres de um arquivo.

tr [options] – substitui ou deleta caracteres

```
$ echo 'Translate' | tr '[a-z]' '[A-Z]'
```

aumenta a caixa da *string* ‘Translate’ (TRANSLATE)