

MANUAL DE COMANDOS:

O que é Shell?

Shell é um programa que permite ao usuário interagir com o sistema operacional através de comandos digitados do teclado. No DOS o shell era o command.com, que permitia executar alguns comandos como: cd, dir, ...

O shell mais famoso de Linux é o Bash, pois o mesmo oferece vários recursos que facilitam a vida do usuário. O mais básico é o sh. Em todos estes é possível criar scripts (mini-programas) que executam sequências de comandos, como se estivessem sido digitados pelo usuário.

Lembremos que para um usuário normal, o shell aparece com o símbolo \$. Já para o root, o símbolo é o #.

Comandos Básicos

Qualquer dúvida em um dos comandos abaixo, rode o man. Por exemplo, se estiver com dúvida no comando **ls** :

```
$ man ls
```

LS

O **ls** é o comando mais básico de um shell. Ele serve para listar o conteúdo de um diretório, mostrando os arquivos que estão no mesmo.

Exemplos:

Listar o diretório atual:

```
$ ls
```

Listar o diretório /etc:

```
$ ls /etc
```

Listar o diretório atual mostrando todos os detalhes dos arquivos:

```
$ ls -l
```

Mostrar arquivos ocultos (que começam com . (ponto final)):

```
$ ls -a
```

Combinando os dois últimos comandos acima:

```
$ ls -la
```

PWD

O **pwd** mostra o diretório atual:

```
$ pwd
```

CD

CD vem de **C**hange **D**irectory (mudar de diretório) e serve justamente para mudar o diretório atual.

Exemplo:

```
$ pwd
/home/luke
$ cd /
$ pwd
/
```

MKDIR

O **mkdir** cria diretórios:

```
$ mkdir tmp
```

RMDIR

O **rmdir** apaga diretórios vazios (como o tmp acima):

```
$ rmdir tmp
```

RM

O **rm** serve para apagar arquivos:

```
$ rm imagem.jpg
```

Para apagar diretórios e seu conteúdo:

```
$ rm -rf tmp
```

DU

O comando **du** verifica o tamanho de diretórios e seus subdiretórios:

```
$ du /etc
```

DF

O **df** verifica o quanto você tem disponível nas suas partições e o quanto foi gasto:

```
$ df
```

Para visualizar melhor (em MegaBytes):

```
$ df -h
```

FREE

O **free** mostra quanto você tem de memória RAM e swap, gastos e livres:

```
$ free
```

FIND

O **find** procura por arquivos em um diretório e seus subdiretórios. Neste exemplo vou procurar arquivos JPG a partir do diretório atual:

```
$ find . -name *jpg
```

Agora vou procurar arquivos MPG no diretório do CDROM:

```
$ find /cdrom -name *mpg
```

WHOAMI

O **whoami** retorna o usuário logado no momento (que executou este comando):

```
$ whoami
```

WHO

O **who** mostra os usuários logados no sistema:

```
$ who
```

HOSTNAME

O **hostname** retorna o nome do computador (na rede, ou não) que se está usando:

```
$ hostname
```

SU

O **su** muda para o root ou para outro usuário:

```
$ su
Passwd:
```

ou

```
$ su baptista
Passwd:
$
```

ECHO

O **echo** escreve um conjunto de caracteres na tela:

```
$ echo "OLinux é um bom site!"
OLinux é um bom site!
```

MOUNT

O **mount** monta (ativa) devices:

```
$ mount /floppy
```

UMOUNT

O **umount** desmonta (desativa) devices:

```
$ umount /floppy
```

CAT

O **cat** imprime arquivos na tela:
\$ cat README

MORE

O **more** imprime arquivos a tela, porém para esperando um retorno do teclado a cada tela cheia:
\$ more README

GREP

O **grep** é uma ferramenta muito poderosa, principalmente para programadores. GREP = Generalized Regular Expression Parser. Facilitando para entender, ele procura por um texto dentro de uma arquivo. Neste exemplo procuro por "autor" dentro do "README":
\$ grep autor README

PS

O **ps** mostra todos os processos (programas) que estão rodando na memória.
Vendo os processos do usuário:
\$ ps
Vendo todos os processos do sistema (de todos os usuários):
\$ ps aux

TOP

O **top** mostra todos os processos que estão rodando com várias outras informações do sistema:
\$ top

TAR

O **tar** server para compactar e descompactar arquivos no formato .tar.gz (tar e gzip).
Compactando:
\$ tar cvfz arquivo.tar.gz [arquivos|diretório]
Descompactando:
\$ tar xvzf arquivo.tar.gz

Pipe

Podemos usar o símbolo especial | para fazer o que chamamos de pipe, que nada mais é do que a conexão da saída de dois programas.
Neste exemplo, eu vou procurar em todos os processos do sistema pelo que contém o texto "init":
\$ ps aux | grep init
Ou seja, a saída de **ps aux** funciona como o arquivo para o **grep init**. É como fazer **ps aux** e gravar sua saída num arquivo chamado **teste**, por exemplo. E depois fazer **grep init teste**. Só que o pipe faz tudo automático, conectando a

saída do **ps** com a entrada do **grep** neste exemplo (sem a necessidade de um arquivo intermediário).

Outro exemplo:

\$ ls -l | more
Neste exemplo a saída do **ls** é usada pelo **more**. Ou seja, o **ls -l** é impresso na tela pausadamente, como se fosse um arquivo usado pelo **more**.

Redirecionamento

Além do pipe, podemos usar o redirecionamento. Para isso, usamos os símbolos < e >, significando entrada e saída respectivamente.

Vamos a um exemplo. Imagine que eu queira guardar o resultado do **ls** na tela. Então faria assim:

\$ ls > teste
O arquivo **teste** foi criado com o que o **ls** imprimiria na tela.

Usando o Bash

Agora vamos ver algumas facilidades do bash.

O bash possui história, ou seja, cada vez que apertamos a tecla que representa uma seta para cima, temos um comando já executado anteriormente. Se apertamos uma vez, temos o último comando executado. Se apertamos duas vezes, temos o penúltimo comando executado. E assim por diante.

Quando estamos digitando um diretório ou arquivo, ao apertarmos a tecla TAB, nos aparece o nome do arquivo todo. Ou seja, ele é completado automaticamente para você. Caso não seja, o bash emite um som via speaker. Se você apertar o TAB e sair este som, aperte de novo o TAB que ele irá lhe mostrar as opções de complemento de nome.

Por exemplo, imagine que no diretório atual eu tenho dois arquivos: **teste1.txt** e **teste2.txt**. Agora digito (sem apertar ENTER):

\$ cat t
Agora apertado TAB e ele completa o nome e emite o som:
\$ cat teste
Então apertado TAB novamente e me aparecem as opções (pois ele não sabe qual eu quero):
\$ cat teste
teste1.txt teste2.txt

Fonte: www.olinux.com.br - Autor: [André Souza](#)

Comandos de rede mais utilizados:

Winipcfg – Windows 95/98/ME

No geral, mostra várias informações sobre uma rede, englobando placas de rede, configurações de ips, servidores dns, nome de host, MAC, etc. Versão gráfica do ipconfig.

Ipconfig/ifconfig

Fornecer informações completas sobre os números ips fornecidos a(s) placas de rede, por Dial-Up e por placa de comunicação. Mostra também configurações do protocolo pppoa. Versão em modo texto do winipcfg.

Sintaxe mais usada:
ipconfig /All - Windows
ifconfig -a - Linux

Netstat

Mostra conexões de rede, tabela de roteamento, estatísticas de interfaces, conexões masquerade, e mensagens.

netstat [opções]

Onde:

opções

-i [interface]

Mostra estatísticas da interface [interface].

-M, --masquerade

Se especificado, também lista conexões masquerade.

-n, --numeric

Usa endereços numéricos ao invés de tentar resolver nomes de hosts, usuários e portas.

-c, --continuous

Mostra a listagem a cada segundo até que a CTRL+C seja pressionado.

Se não for especificada nenhuma opção, os detalhes das conexões atuais serão mostrados.

Ping

É utilizado para testar uma conexão, sendo que este utilitário de diagnóstico utiliza-se das mensagens Echo Request e Echo Reply do protocolo ICMP para determinar se uma máquina está ligada e funcional. Ele opera enviando um ICMP (Control Message Protocol), se o software de IP da máquina destino recebe-o ele emite uma resposta de echo imediatamente.

sintaxe :

```
ping 192.168.0.1 -t
ping www.terra.com.br
ping phr34k3r
```

ping [-t] [-a] [-n x] [-l tamanho] [-f] [-i ttl] [-v tos] [-r count] [-s count] [-j host_list] | [-k host_list]] [-w timeout] destination_list

Por default o ping envia quatro pacotes de 64 bytes ICMP com uma pausa entre cada pacote .

T : Indica que o ping deve enviar pacotes continuamente até que se aperte Ctrl+c ;

A : É usado para solucionar o endereço IP para o hostname da DNS ;

N x : Indica que o ping deve enviar x pacotes ICMP . O Default é 4 ;

L tamanho : Indica a duração do pacote de ICMP . O conteúdo do pacote de ICMP é uma sucessão periódica de caracteres alfabéticos . O tamanho de pacote por default é 64 bytes e o máximo é 8192 , porém , redes Ethernet têm como máximo o tamanho de 1512 bytes ;

F : Fixa o flag de não-fragmentado no pacote . Se você usar o -l e indicar um pacote , este será devolvido com um erro . Você pode usar esta opção , junto com -l , para descobrir o maior pacote que você pode enviar de seu computador a um host remoto sem que seja fragmentado ;

I ttl : Fixa o time live (ttl) do campo dos pacotes . Ottl é o número máximo de "saltos" que pacote pode ter antes de ser descartado . Este número varia de 1 a 255 , sendo 30 o default ;

V tos : Fixa o campo do Serviço para o valor especificado por tos ;

R count : Registros da rota dos pacotes ICMP . Um mínimo de um e máximo de nove hosts devem ser especificados através de count .

S count : Especifica o time stamp para o número de "saltos" especificado por count .

J Host_list : Envia pacotes por meio da lista de hosts especificada por host_list . Hosts sucessivos podem ser separados através de gateways intermediários . Nove é o número máximo de hosts permitido ;

K Host_list : Envia pacotes por meio da lista de hosts especificada por host_list . Hosts sucessivos não podem ser separados através de gateways intermediários . O número máximo de hosts permitidos é 9 ;

W timeout (Intervalo) : Especifica im intervalo de tempo em milissegundos ;

Destination_list : Especifica os hosts distantes para ping .

Tracert/Traceroute

Mostra o caminho percorrido por um pacote para chegar ao seu destino. Este comando mostra na tela o caminho percorrido entre os Gateways da rede e o tempo gasto de retransmissão. Este comando é útil para encontrar computadores defeituosos na rede caso o pacote não esteja chegando ao seu destino .

tracert [opções] [host/IP de destino] - para sistemas operacionais *nix

Onde:

host/IP destino

É o endereço para onde o pacote será enviado (por exemplo, <http://www.facens.br>). Caso o tamanho do pacote não seja especificado, é enviado um pacote de 38 bytes.

opções:

-l

Mostra o tempo de vida do pacote (ttl)

-m [num]

Ajusta a quantidade máximas de ttl dos pacotes. O padrão é 30.

-n

Mostra os endereços numericamente ao invés de usar resolução DNS.

-p [porta]

Ajusta a porta que será usada para o teste. A porta padrão é 33434.

-r

Pula as tabelas de roteamento e envia o pacote diretamente ao computador conectado a rede.

-s [end]

Usa o endereço IP/DNS [end] como endereço de origem para computadores com múltiplos endereços IPs ou nomes.

-v

Mostra mais detalhes sobre o resultado do traceroute.

-w [num]

Configura o tempo máximo que aguardará por uma resposta. O padrão é 3 segundos.

tracert - Windows

traceroute - Linux

[opções] [host/IP de destino]

host/IP destino

É o endereço para onde o pacote será enviado (por exemplo, <http://www.facens.br>). Caso o tamanho do pacote não seja especificado, é enviado um pacote de 38 bytes.

opções:

-d

Não resolver endereços para nomes hosts.

-h nmax_saltos

Número máximo de saltos para a procura do destino.

-j lst_hosts

Rota ampliada de origens usada com a lista lst_hosts.

-w tempo_limite

Tempo limite de espera em milissegundos para cada resposta.

Nbtstat

Mostra estatísticas de protocolos e conexões de TCP/IP correntes usando NBT (NetBIOS) sobre TCP/IP.

- "nbtstat -a" para listar as máquinas por nome.
- "nbtstat -A" para listar as máquinas por IP.
- "nbtstat -c" para listar o nome do cache remoto incluindo os endereços IP.
- "nbtstat -n" para listar os nomes de NETBIOS Local.
- "nbtstat -r" para listar nomes resolvidos por Broadcast e por WINS.
- "nbtstat -R" para recarregar a tabela de cache remoto.
- "nbtstat -S" para listar a tabela de sessões com os IPs de destino.
- "nbtstat -s" para listar tabela de sessões convertendo IP de destino para nomes de Hosts pelo arquivo de Hosts.

ftp

Permite a transferência de arquivos do computador remoto/local e vice versa. O file transfer protocol é o sistema de transmissão de arquivos mais usado na Internet. É requerida a autenticação do usuário para que seja permitida a conexão. Muitos servidores ftp disponibilizam acesso anônimo aos usuários, com acesso restrito.

Uma vez conectado a um servidor ftp, você pode usar a maioria dos comandos do GNU/Linux para operá-lo.

ftp [ip/dns]

Abaixo alguns dos comandos mais usados no FTP:

binary

Ou simplesmente **bin**. Estabelece como binário o tipo de representação dos arquivos a serem manipulados. Use este comando sempre que for lidar com arquivos de imagem, documentos formatados, executáveis e arquivos compactados.

cd diretório_remoto

Muda o diretório de trabalho na máquina remota.

cdup

Muda o diretório de trabalho para o diretório "pai" (superior) do diretório atual.

lcd [diretório]

Muda o diretório de trabalho na máquina local. Se nenhum diretório for especificado, o diretório "home" do usuário é utilizado.

get arq_remoto [arq_local]

Recupera o arquivo_remoto e o armazena na máquina local. Se um nome de arquivo_local não for especificado, é dado o mesmo nome do arquivo na máquina remota.

mget arquivos_remotos

Faz um **get** para cada arquivo

put arq_local [arq_remoto]

Armazena um arquivo local na máquina remota. Se não for especificado um nome arquivo remoto, ele terá o mesmo nome do arquivo local.

mput arquivos_locais

Faz um **put** para cada arquivo local cujo nome esteja na lista de arquivos_locais

help [comando]

Ou apenas **?**, escreve uma mensagem explicativa sobre o significado do comando.

Quando nenhum comando é especificado, apresenta uma lista de comandos.

ls [dir_remoto] [arq_local]

Dá uma listagem do conteúdo de um diretório da máquina remota. Se não for especificado um diretório remoto, é mostrada a listagem do diretório de trabalho usado.

Se nenhum arquivo local for especificado, a listagem é mostrada na tela.

dir [dir_remoto] [arq_local]

Lista o conteúdo do diretório da máquina remota, colocando o resultado na máquina local.

Se nenhum diretório remoto for especificado, o diretório de trabalho atual na máquina remota será utilizado.

Se nenhum arquivo na máquina local for especificado para receber a lista do diretório remoto, o resultado é enviado para o terminal.

ldir [dir_local] [arq_local]

Em alguns sistemas ftp, este comando lista o conteúdo de um diretório da máquina local.

pwd

Retorna o nome do diretório atual na máquina remota.

quit

Termina uma sessão ftp

Telnet

Permite acesso a um computador remoto. É mostrada uma tela de acesso correspondente ao computador local onde deve ser feita a autenticação do usuário para entrar no sistema. Muito útil, mas deve ser tomado cuidados ao disponibilizar este serviço para evitar riscos de segurança.

telnet [opções] [ip/dns] [porta]

onde:

ip/dns

Endereço IP do computador de destino ou nome DNS.

porta

Porta onde será feita a conexão. Por padrão, a conexão é feita na porta 23.

opções

-8

Requisita uma operação binária de 8 bits. Isto força a operação em modo binário para envio e recebimento. Por padrão, telnet não usa 8 bits.

-a

Tenta um login automático, enviando o nome do usuário lido da variável de ambiente USER.

-d

Ativa o modo de debug.

-r

Ativa a emulação de rlogin.

-l [usuário]

Faz a conexão usando [usuário] como nome de usuário.

Exemplo: telnet 192.168.1.1, telnet 192.168.1.1 23.