

AULA 4

1 - Utilizando pela primeira vez: Comandos básicos

Após instalado, você vai utilizar seu sistema Linux aos poucos em modo de console (*Command Line Interface*) para poder se familiarizar com os mecanismos e interface de funcionamento.

Há saber, os carregadores de sistema mais utilizados são o *grub* e o *lilo*. Estes são executados após a inicialização do computador. Neles, é possível escolher uma imagem de Linux armazenada no disco rígido e então deixá-la controlar a máquina como sendo o kernel do sistema operacional.

Esse sistema de inicialização e de *boot* será explicado com mais detalhes nas aulas mais a frente.

Para o momento, o que interessa, é saber, como é possível utilizar o sistema de duas maneiras: modo gráfico ou modo texto. Em geral indica-se, como já citado, utilizá-lo em modo texto para poder treinar as habilidades e acostumar-se com o ambiente dos sistemas POSIX.

Inicialmente, é possível que você faça o *login*, ou conecte-se ao sistema através de terminais, que dependendo da configuração do sistema podem variar entre um até sete ou mais. Esses terminais podem se acessados através do acionamento das teclas *CTRL+ALT+Fn*, onde *n* é o número do terminal desejado.

Escolhendo um terminal (*tty*), deve-se utilizar um usuário credenciado no sistema para que se possa fazer o *login*. Existem dois tipos de usuários: o administrador e os usuários convencionais.

Utilizando o usuário administrador (*root*) você terá acesso a todo sistema, por isso, é sempre recomendada a utilização dos usuários convencionais. Somente nos momentos que se deseja utilizar um comando privilegiado, deve-se realizá-lo temporariamente como usuário *root*.

Após o *login* ou acesso ao sistema, você estará executando um interpretador de comandos (*Shell*). Este *Shell* tem a capacidade de receber instruções e executá-las. Em geral, são passados comandos para que esse interpretador, nesse caso a interface com o sistema, possa devolver as respostas para a saída padrão.

Existem vários tipos de interpretadores. Atualmente, o mais utilizado e com mais recursos é o interpretador *bash*. Por isso, recomenda-se seu uso enquanto principiante em ambientes de console.

2 - Operando o sistema

Existem comandos considerados internos ou aqueles que são externos. Comandos internos são aqueles que o próprio interpretador possui embutido em seu código. Com isso, ao se carregar o interpretador em memória, estes comandos já estarão residentes e poderão ser diretamente processados. O resultado disso é a rapidez de execução e o tempo de resposta desses comandos. Conjuntos desses comandos devem variar conforme a versão do interpretador, entretanto os mais clássicos são *cd*, *alias*, *kill*, *logout*, etc.

Por outro lado, existem comandos externos. Estes são a maioria dos comandos existentes

e ficam em disco rígido e quando executados são trazidos para a memória. Isto é, o interpretador, em geral, faz uma chamada de sistema, invocando o comando selecionado para que seja executado. Exemplos destes são `ls`, `cp`, `rm`, `mkdir`, etc.

Em geral, os interpretadores de comando apresentam um sinal de *prompt* com um símbolo (caractere). Pode-se notar que ao utilizar a sessão como usuário convencional, o *prompt* tem o símbolo de `$`, já quando o usuário é o *root*, o *prompt* deverá aparecer com o símbolo `#`.

3 - Principais comandos

login: comando que serve para que seja iniciada uma nova sessão. É requisitado um usuário e senha.

logout: comando para realizar o encerramento de uma sessão. Em geral pode-se fazê-lo também pelo atalho `CTRL+d`.

exit: mesmo resultado de `logout`, onde a sessão é encerrada.

reboot: equivalente ao reinício do sistema. Desliga todos os serviços e reinicia o computador.

halt: comando para desligar o computador. Invoca todos desligamentos de serviços

shutdown: similar ao comando `halt`. Pode-se passar parâmetros de desligamento.
Ex: `shutdown -h now`

man: comando que exibe a página de manual de um comando, aplicação ou função. Podem existir vários números para um mesmo manual.
Ex: `$> man 5 shutdown`

info: exibe uma descrição mais detalhada da ajuda de um comando. Uma versão mais aprofundada do comando `man`.

whatis: consulta manual online para um comando ou palavra-chave. Esse banco deve ser sempre atualizado com o comando `makewhatis`.

apropos: utiliza o mesmo conceito do `whatis`, exibindo informações relevantes a um determinado assunto ou palavra-chave.

cd: comando extremamente utilizado. Serve para navegação em diretório, alterando o diretório corrente.

Ex: `cd /etc`
`cd /var/log`
`cd ~batman`
`cd ../../..`
`cd /`
`cd -`

cd +

pwd: exibe o caminho completo em relação à raiz do diretório corrente.

tree: exibe a árvore de diretórios especificada.

Ex: tree /

ls: outro comando muito utilizado. Lista arquivos e diretórios conforme a especificação em seus parâmetros.

Ex: ls -la

ls -lart

find: procura por arquivos no disco.

Ex: find . -name "**texto**"

find / -ctime +5

grep: filtra caracteres a partir de um texto uma variável de entrada.

Ex: grep root /etc/passwd

grep inet /etc/*

strings: exibe caracteres de arquivos (em geral binários).

Ex: strings /bin/cat

whereis: localiza o caminho para um arquivo ou executável.

Ex: whereis ls

locate: localiza um arquivo dentro do sistema de arquivos. Essa busca é realizada em uma base de dados que deve ser atualizada com o comando *updatedb*.

which: localiza um arquivo (em geral executável) dentro do caminho padrão (PATH) da sessão atual.

touch: cria um arquivo vazio.

rm: remove arquivos e diretórios.

Ex: rm /var/log/*

rm -r /tmp/*

rm ~aluno/.Xsession

rm -rf /tmp

mkdir: cria diretórios

Ex: mkdir ~megadeth/musicas

mkdir /home/novousuario

rmdir: remove diretórios

Ex: rmdir /lixo

mv: move o arquivo origem para o destino. Usado também para renomear arquivos.

Ex: mv velhonome novonome

mv /var/log .

cp: copia arquivos da origem para o destino especificado.

Ex: cp /tmp/*.txt ./aluno/

cp *.html /tmp/

ln: cria links (atalhos). O parâmetro da esquerda ou origem é o atalho e o da direita é o arquivo original.

Ex: ln -s /usr/src/Linux/Map Linux-2.6.12-map

clear: limpa o terminal.

reset: reinicia o terminal sem precisar refazer o *login*.

echo: envia dados para a saída padrão. Em geral texto para saída no terminal.

Ex: echo 'Ola Mundo'

write: envia mensagem a um determinado terminal

Ex: Write lemmy <ENTER>

Estou esperando na saída <ENTER>

talk: inicia conversa o com outro usu rio

Ex: talk jose

wall: envia mensagem a todos usu rios conectados

Ex: wall < /tmp/mensagem_de_boot.txt

date: exibe ou manipula a data do sistema

cal: exibe na tela o calend rio

uname: exibe informa es sobre o sistema operacional e hardware

cat: exibe conte do na saida padr o de arquivos sem utilizar quebra de pagina o

Ex: cat /etc/passwd

more: exibe o conte do de arquivo na sa da padr o utilizando a quebra e navega o por p ginas.

Ex: more /etc/passwd

less: exibe o conte do de arquivo na sa da padr o utilizando a quebra e navega o por linhas e colunas.

Ex: less /etc/passwd

head: exibe na sa da padr o as linhas iniciais de arquivo (cabe a).

Ex: head /var/log/messages

tail: exibe na saída padrão as linhas finais de um arquivo (rabo).

Ex: tail /var/log/messages

wc: conta o número de palavras/linhas e exibe o resultado na saída padrão.

Ex: wc -l /var/log/syslog

wc /etc/passwd

tr: substitui caracteres da entrada (esquerda) pelos caracteres da saída (direita).

Ex: tr r p < /etc/passwd

sort: ordena os dados recebidos na entrada e os exibe na saída padrão.

Ex: sort /etc/hosts

cut: Seleciona campos de um arquivo recebido conforme o filtro estabelecido e os exibe na tela

Ex: cut -f 1,3 -d: /etc/passwd

paste: Exibe na saída padrão lado a lado as linhas de dois arquivos

Ex: paste -d, /etc/group /etc/passwd

diff: compara e exibe na tela as diferenças entre dois arquivos texto.

Ex: diff arquivo1 arquivo2

tar: concatena arquivos de entrada em um único arquivo de saída.

Ex: tar -cvf arquivo_final.tar arquivo1 arquivo2 arquivo3

split: fraciona um arquivo de entrada em vários arquivos de saída conforme a especificação de tamanho.

Ex: split -b1500k arquivo.log arquivo.

gzip: compacta uma ou mais arquivos.

Ex: gzip backup.tar

gunzip: descompacta arquivos compactados com o gzip.

Ex: gunzip backup.tar.gz

bzip2: compacta uma ou mais arquivos.

Ex: bzip2 backup.tar

4 - Bibliografia:

Este documento é resenha retirada do capítulo 3 de: Ferreira, E. Rubem. Guia do Administrador Linux. 1 edição, 2003, Novatec Editora, São Paulo.