

Fundamentos básicos de Linux

Marcos Flávio Araújo Assunção
Fundamentos de Ethical Hacking

1. SURGIMENTO DO LINUX

O Linux foi originalmente desenvolvido como um passatempo de Linus Torvalds. Ele queria um sistema operacional que fosse semelhante a um Unix, com todas as suas funcionalidades e, ainda, que pudesse utilizá-lo num PC. A partir dessa idéia, Linus começou a trabalhar nesse que seria o futuro kernel do sistema operacional que hoje é chamado de *Linux*.

1.2 KERNEL

O kernel é o núcleo do sistema operacional, a parte que controla o hardware da máquina. Quando falamos de Linux, estamos nos referindo apenas ao kernel do sistema. Tudo que existe *ao redor* do kernel são aplicativos que compõem uma distribuição do Linux.

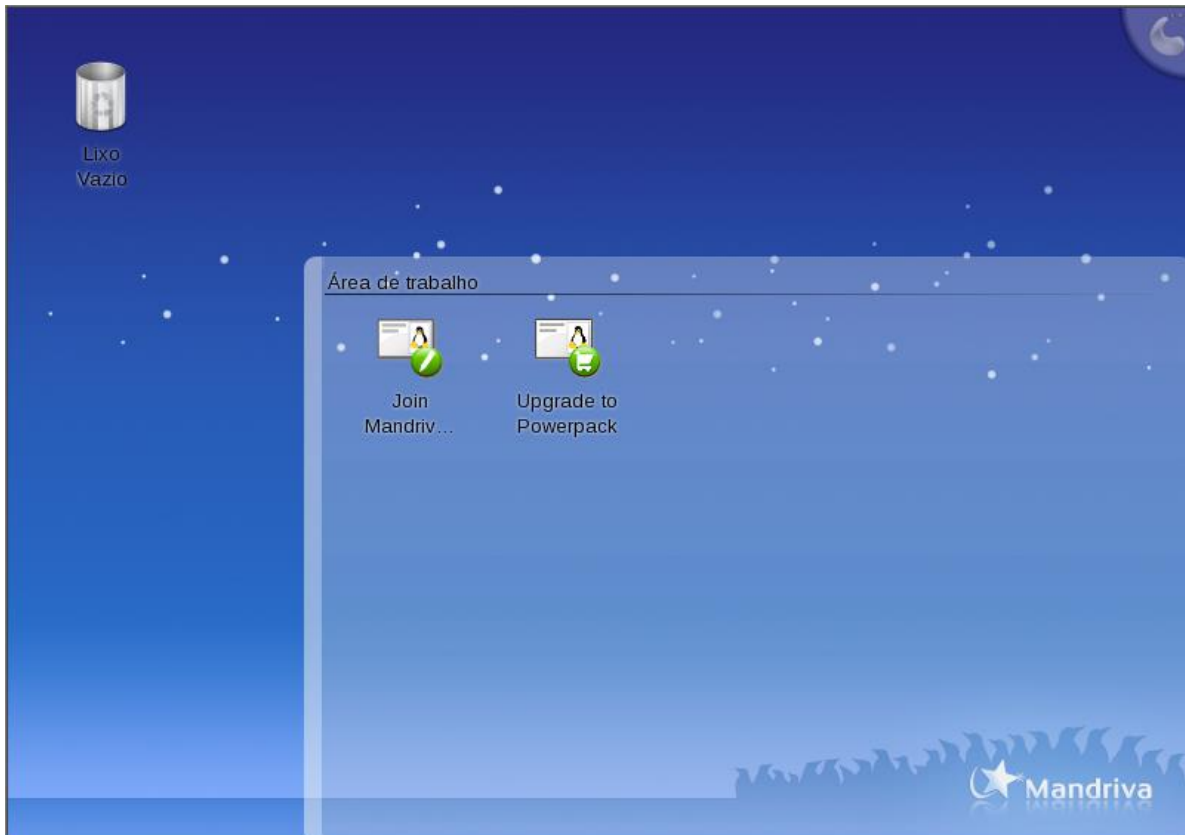
2. DISTRIBUIÇÕES LINUX

Pelo fato de o Linux ser um software livre, muitas empresas ou mesmo pessoas físicas se empenham em organizar pacotes que contém o kernel e mais uma série de aplicativos e manuais para que o sistema atenda a demandas específicas..

A esse conjunto de aplicativos mais o kernel dá-se o nome de distribuição Linux. Algumas distribuições Linux são maiores que outras, dependendo da quantidade de aplicativos e a finalidade a que se propõem.

Entre as "grandes" distribuições pode-se citar: *Debian, Red Hat, Fedora, Suse, CentOS, Ubuntu, etc....* O que diferencia uma distribuição de outra é a maneira como são organizados e pré-configurados os aplicativos que cada uma contém

3. MODO TEXTO X MODO GRÁFICO



No Linux a responsabilidade pelo ambiente gráfico não é do kernel e sim de um programa especial, o XFree86. No entanto, este programa provê apenas as funções de desenho de elementos gráficos e interação com a placa de vídeo. A interação final do usuário com a interface gráfica se dá através de programas gerenciadores de janelas, como o KDE, o WindowMaker e o GNOME, e são eles os responsáveis pela "aparência" do seu Linux.

A separação do ambiente gráfico do resto do sistema apresenta muitas vantagens. Como o ambiente gráfico consome recursos do sistema, é possível desativá-lo, principalmente em servidores, resultando assim em um melhor desempenho de outras aplicações, uma vez que a quantidade de processamento da CPU que seria utilizado para o XFree86, poderá ser utilizada para essas aplicações. Além do mais, o desenvolvimento do ambiente gráfico pode ocorrer de maneira independente ao do kernel.

O Linux também pode funcionar em modo texto. Nesse caso a interação com o usuário se dá por meio de um *shell*, como o Bash, que é capaz de interpretar e executar comandos digitados pelo usuário.

```
Mandriva Linux release 2009.0 (Official) for i586
Kernel 2.6.27-desktop-0.rc8.2mnb on an i686 / tty1
localhost login: root
Password:
Last login: Mon Jun  8 12:59:49 on tty1
[root@localhost ~]#
```

Nome de Usuário
Senha do Usuário
símbolo que identifica o usuário ROOT
símbolo que identifica a pasta do usuário

4. COMANDOS DE NAVEGAÇÃO.

Assim como no Prompt de Comando do Windows, podemos utilizar o shell do Linux para navegar no sistema, criando diretórios, arquivos, configurando permissões, dentre várias outras possibilidades.

É necessário entender o significado de simbologias comuns no Shell:

/ : Representa a Raiz do Sistema

~ : É usado para representar a pasta pessoal do usuário.

\$: Identifica o usuário comum logado no sistema.

: Identifica o usuário root logado no sistema.

```
[root@localhost ~]#
[root@localhost ~]#
[linux@localhost ~]$
[linux@localhost ~]$
```

No primeiro exemplo, estamos localizados na raiz do Linux, utilizando o usuário root. Já no segundo exemplo, estamos na pasta home, utilizando o usuário root.

Vamos explicar sobre os diretórios do Linux:

/ : Raiz do sistema

/BIN : Comandos e utilitários usados por todos os usuários .

/SBIN: Comandos e utilitários que só podem ser utilizados pelo root

/BOOT: Contem os arquivos necessários para a inicialização do sistema.

/LIB: Contem as bibliotecas essenciais que o sistema necessita.

/PROC :Contem informações sobre hardware e o S.O da máquina.

/VAR: Contém arquivos com conteúdo muito variado:

/ROOT: Arquivos ou diretórios pessoais do usuário root (ou super usuário)

/HOME: Arquivos e diretórios pessoais dos usuários (com exceção do root)

/USR: Contém todas as configurações e programas instalados pelos usuários.

/ETC: Contém os arquivos de configuração de quase tudo no Linux.

/DEV: Arquivos de dispositivos de hardware:

Ex.: FD0 →disquete

HDC → cd-rom

/MNT : Contém o ponto de montagem dos dispositivos que estão em DEV (É onde você acessa os arquivos dos dispositivos)

Alguns comandos necessários:

- **startx**: inicia o Linux no modo gráfico;
- **logout**: sai do usuário atual e volta à tela de login;
- **clear**: limpa a tela.

4.1 – Comando cd

O comando **cd**, torna corrente o diretório especificado. Vejamos o exemplo:

```
Mandriva Linux release 2009.0 (Official) for i586
Kernel 2.6.27-desktop-0.rc8.2mnb on an i686 / tty1
localhost login: root
Password:
Last login: Tue Jun  9 11:50:12 on tty1
[root@localhost ~]# cd /
[root@localhost /]#
```

Logamos com o usuário root, e caímos no diretório home do usuário (no caso do root, a pasta /root). Nós queremos nos transferir para a raiz (o C: no Windows), então usamos o comando cd para entrar na raiz. Note que logo abaixo a barra que representa a raiz entra dentro do colchetes, significando que o diretório está corrente.

Portanto, a **sintaxe** do comando faz-se:

cd <diretório>

4.1.1 – Comando cd..

O comando cd.. volta ao diretório anterior. Note:

```
[root@localhost /]# cd /etc/ggi
[root@localhost ggi]# cd..
[root@localhost etc]# _
```

Estando na raiz, entraremos na pasta /etc/ggi. Observamos, que na linha abaixo, o diretório ggi, tornou-se corrente. Porém, ao digitar cd.., eu volto ao diretório anterior ao ggi, nesse caso o etc.

4.2 – Comando whoami

Comando utilizado para verificar o usuário logado na máquina local.

```
[root@localhost ~]# whoami
root
[root@localhost ~]# _
```

4.3 – Comando pwd

Comando pwd, exibe na tela a localização exata do diretório corrente. Veja:

```
[root@localhost run]# pwd
/var/run
[root@localhost run]# _
```

Estamos dentro do diretório run, mais não sabemos exatamente de onde é essa pasta, ou subpasta. Entrando com o comando pwd, exibirá-se a localização exata do diretório logo abaixo, que nesse caso, pertence ao diretório /var.

4.4 – Comando ls

O comando ls, lista diretórios e arquivos dentro do diretório. Veja:

```
[root@localhost bin]# ls
arch*      date*      fbtruetype.static*  kill*      mv*         rvi@      umount*
ash@        dd*        fgrep*             ksh*       netstat*    rview@    umount.cifs@
awk@        df*        find*              link*       nice*       rvim@     umount.cifs3*
basename*   dmesg*     gawk*              ln*         nisdomainname@ sed*       uname*
bash*       dnsdomainname@ gawk-3.1.4*       login*      pidof@      sh@       unicode_start*
bash3@      doexec*    gettext*           ls*         ping*       sleep*    unlink*
bsh@        domainname@ grep*            mail*       procp3-kill* sort*      usleep*
cat*        echo*      gtar@              nkndir*     progress*   stat*     vi@
chgrp*      ed*        gunzip*            mknod*      ps*         stty*     view@
chmod*      egrep*     gzip*              nktemp*     pwd*        su*       vim@
chown*      env*       hostname*          more*       rbash@      sync*     vim-minimal*
cp*         ex@        id*                mount*      red@        tar*      ypdomainname@
cpio*       expr*      igawk*             mount.cifs@ rm*         tcsh*     zcat*
csh@        false*     ipcalc*            mount.cifs3* rmdir*      touch*
cut*        fbresolution* ipv6calc*          mountpoint* rpm*        true*
```

Dica: Para executar o comando, não é necessário tê-lo corrente. Basta usar a sintaxe ls <diretório>.

4.4.1 – Comando ls -l

A opção -l do comando ls, permite listar os arquivos de forma que se possa ver as permissões, grupo, data de criação. Veja.

```
lrwxrwxrwx 1 root root 11 Out 6 2006 bsh -> ../sbin/bsh*
-rwxr-xr-x 1 root root 16124 Ago 18 2005 cat*
-rwxr-xr-x 1 root root 34528 Ago 18 2005 chgrp*
-rwxr-xr-x 1 root root 34724 Ago 18 2005 chmod*
-rwxr-xr-x 1 root root 36780 Ago 18 2005 chown*
-rwxr-xr-x 1 root root 55552 Ago 18 2005 cp*
-rwxr-xr-x 1 root root 98060 Jul 19 2005 cpio*
lrwxrwxrwx 1 root root 4 Out 6 2006 csh -> tcsh*
-rwxr-xr-x 1 root root 26660 Ago 18 2005 cut*
-rwxr-xr-x 1 root root 45520 Ago 18 2005 date*
-rwxr-xr-x 1 root root 28196 Ago 18 2005 dd*
-rwxr-xr-x 1 root root 34556 Ago 18 2005 df*
-rwxr-xr-x 1 root root 4480 Set 20 2005 dmesg*
lrwxrwxrwx 1 root root 8 Out 6 2006 dnsdomainname -> hostname*
-rwxr-xr-x 1 root root 2668 Set 19 2005 doexec*
lrwxrwxrwx 1 root root 8 Out 6 2006 domainname -> hostname*
-rwxr-xr-x 1 root root 13884 Ago 18 2005 echo*
-rwxr-xr-x 1 root root 43608 Jun 7 2004 ed*
-rwxr-xr-x 1 root root 33 Dez 22 2004 egrep*
-rwxr-xr-x 1 root root 12908 Ago 18 2005 env*
lrwxrwxrwx 1 root root 20 Out 6 2006 ex -> /etc/alternatives/ex*
-rwxr-xr-x 1 root root 22148 Ago 18 2005 expr*
-rwxr-xr-x 1 root root 11388 Ago 18 2005 false*
-rwxr-xr-x 1 root root 4088 Set 16 2005 fbresolution*
-rwxr-xr-x 1 root root 892752 Set 16 2005 fbtruetype.static*
-rwxr-xr-x 1 root root 33 Dez 22 2004 fgrep*
```



4.4.2 – Comando ls -la

A opção -a, permite a visualização de arquivos ocultos.

```
[root@localhost ~]# ls -la
total 92
drwx----- 13 root root 4096 2009-04-30 13:08 ./
drwxr-xr-x 32 root root 4096 2009-06-09 11:49 ../
drwxr-xr-x 2 root root 4096 2009-04-30 08:23 aif-mount/
-rw-r--r-- 1 root root 145 2006-08-04 20:34 .bash_completion
-rw----- 1 root root 1937 2009-06-09 12:10 .bash_history
-rw-r--r-- 1 root root 24 2006-08-04 20:34 .bash_logout
-rw-r--r-- 1 root root 106 2006-08-04 20:34 .bash_profile
-rw-r--r-- 1 root root 226 2006-08-04 20:34 .bashrc
-rw-r--r-- 1 root root 233 2006-08-04 20:34 .cshrc
drwx----- 2 root root 4096 2009-04-30 08:23 drakx/
drwxr-xr-x 2 root root 4096 2009-04-30 08:20 DrakX-screenshots/
drwx----- 3 root root 4096 2009-04-30 11:15 .gconf/
drwx----- 2 root root 4096 2009-04-30 11:16 .gconfd/
drwx----- 3 root root 4096 2009-04-30 11:15 .gnome2/
drwx----- 2 root root 4096 2009-04-30 11:15 .gnome2_private/
drwx----- 2 root root 4096 2009-04-30 11:14 .kde4/
drwxr-xr-x 3 root root 4096 2009-04-30 13:08 .mc/
-rw-r--r-- 1 root root 189 2006-08-04 20:34 .tcshrc
drwx----- 3 root root 4096 2009-06-09 11:49 tmp/
-rw----- 1 root root 1657 2009-04-30 13:08 .viminfo
-rw-r--r-- 1 root root 927 2006-08-04 20:34 .vimrc
drwxr-xr-x 2 root root 4096 2009-04-30 08:10 .xauth/
-rw-r--r-- 1 root root 1479 2006-08-04 20:34 .Xdefaults
[root@localhost ~]#
```

Note que todos os arquivos **.nomedoarquivo** significa que é oculto.

4.5 Comando man

Esse comando permite visualizar o 'manual' do comando.

A sintaxe é:

man <comando>

```
[root@localhost ~]# man logout
BASH_BUILTINS(1)                                BASH_BUILTINS(1)

NAME
    bash, :, ., [, alias, bg, bind, break, builtin, cd, command, compgen, complete,
    continue, declare, dirs, disown, echo, enable, eval, exec, exit, export, fc, fg,
    getopts, hash, help, history, jobs, kill, let, local, logout, popd, printf, pushd,
    pwd, read, readonly, return, set, shift, shopt, source, suspend, test, times, trap,
    type, typeset, ulimit, umask, unalias, unset, wait - bash built-in commands, see
    bash(1)

SEE ALSO
    bash(1), sh(1)

GNU Bash-3.0                                2004 Apr 20                                BASH_BUILTINS(1)
lines 1-18/18 (END)
```

Para sair do ‘manual’, aperta-se a tecla ‘q’.

4.6 Comando file

Identifica o tipo de arquivo.

```
[root@localhost ~]# file /etc/passwd
/etc/passwd: ASCII text
[root@localhost ~]#
```

6. COMANDO DE MANIPULAÇÃO

6.1 Comando cat e tac

O comando cat permite visualizar o arquivo, e o comando tac permite visualizar ao contrário. Veja no exemplo:

```
[root@localhost etc]# cat shells
/bin/bash
/bin/csh
/bin/ksh
/bin/sh
/bin/tcsh
[root@localhost etc]# tac shells
/bin/tcsh
/bin/sh
/bin/ksh
/bin/csh
/bin/bash
[root@localhost etc]# _
```



A sintaxe é:

cat <arquivo>

cat /etc/passwd

tac <arquivo>

tac /etc/passwd

6.3 Direcionadores e Separadores

Usamos ambos para auxiliar os comandos principais

6.3.1 – Maior (>)

É um direcionador. Ele envia a saída do comando para um arquivo ou dispositivo. Se houver algo armazenado, ele sobrescreve o conteúdo. Exemplo:

```
[root@localhost ~]# tree var > resultado
[root@localhost ~]# ls
backup/  initrd/  ooobuildtime.log  pasta4/  predio/  sys/  texto5*  tmp/
bin/     lib/     opt/              pasta5/  proc/    texto1*  texto6  usr/
boot/    lost+found/  pasta1/          pasta6/  provedores  texto10  texto7*  usuarios
dev/     mandriva2009/  pasta10/        pasta7/  resultado  texto2*  texto8*  var/
etc/     media/       pasta2/          pasta8/  root/      texto3*  texto9
home/    mnt/        pasta3/          pasta9/  sbin/      texto4*  textos/
[root@localhost ~]#
```

Nesse caso estamos jogando o resultado do comando tree para um arquivo chamado 'resultado'. Logo abaixo, usando o comando ls, podemos ver que ele foi criado.

6.3.2 – Maior Maior (>>)

Também um direcionador. Envia a saída do comando para um arquivo ou dispositivo. Se houver algo armazenado, essa opção adiciona o conteúdo.


```

[root@localhost ~]# cat provedores >> usuarios
[root@localhost ~]# cat usuarios
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
rpm
messagebus
polkituser
vcsa
haldaemon
avahi
avahi-autoipd
htdig
mysql
ftw
linux
terra.com.br
uol.com.br
vivaolinux.com.br
microsoft.com
uai.com
ig.com
mg.senac.com.br

```

6.3.3 – Dois pontos (;)

É um separador. Usado para entrar outro comando na mesma linha. Veja o exemplo:

```

[root@localhost ~]# cp texto1 pasta1 ; cp texto2 pasta2
[root@localhost ~]# tree pasta
pasta [error opening dir]

0 directories, 0 files
[root@localhost ~]# tree pasta1
pasta1
|-- texto1

0 directories, 1 file
[root@localhost ~]# tree pasta2
pasta2
|-- texto2

0 directories, 1 file
[root@localhost ~]# _

```

Na primeira linha, estamos copiando o texto1 para a pasta1 e texto2 na pasta2, ambos criados na raiz, numa única linha.

Abaixo, utilizando o comando tree (esse comando visualiza os diretórios e arquivos em formato árvore), vemos os devidos arquivos em suas devidas pastas.

6.4 Comando head e tail

O comando head (que significa cabeça) permite ver as 'n' primeiras linhas do arquivo. Já o comando tail (significa cauda) permite ver as 'n' últimas linhas do arquivo.

A sintaxe dá-se:

head <nº de linhas> <arquivo>

tail <nº de linhas> <arquivo>

Vejamos o exemplo:

```
[root@localhost ~]# head -5 /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

[root@localhost ~]# tail -5 /etc/crontab
# run-parts
01 * * * * root nice -n 19 run-parts --report /etc/cron.hourly
02 4 * * * root nice -n 19 run-parts --report /etc/cron.daily
22 4 * * 0 root nice -n 19 run-parts --report /etc/cron.weekly
42 4 1 * * root nice -n 19 run-parts --report /etc/cron.monthly
[root@localhost ~]# _
```

No primeiro comando, vemos as primeiras 5 linhas do arquivo. No segundo, vemos as 5 últimas linhas.

6.5 - Comando mkdir

Permite criar pastas no linux. Usando a opção -p, criamos toda a estrutura hierárquica de pastas de uma só vez.

Sintaxe utilizada:

Mkdir <opção> <nomedapasta>

Exemplo:

```
[root@localhost ~]# mkdir carol
[root@localhost ~]# ls
bin/      home/      media/     pasta10/  pasta6/   proc/     texto1*   texto5*   tmp/
boot/     initrd/    mnt/       pasta2/   pasta7/   root/     texto10   texto6    usr/
carol/    lib/       ooobuildtime.log  pasta3/   pasta8/   sbin/     texto2*   texto7*   var/
dev/      lost+found/ opt/        pasta4/   pasta9/   sys/      texto3*   texto8*
etc/      mandriva2009/ pasta1/     pasta5/   predio/   texto    texto4*   texto9
[root@localhost ~]#
```

Usando o comando `mkdir`, criamos a pasta denominada 'carol' localizada no diretório corrente.

Vejam os outros exemplos:

```
[root@localhost ~]# mkdir -p aluno/trabalhos aluno/pessoal
[root@localhost ~]# ls
aluno/  etc/      mandriva2009/  pasta1/  pasta5/  predio/  texto  texto4*  texto9
bin/    home/     media/         pasta10/ pasta6/  proc/    texto1* texto5*  tmp/
boot/   initrd/   mnt/          pasta2/  pasta7/  root/    texto10 texto6    usr/
carol/  lib/      ooobuildtime.log pasta3/  pasta8/  sbin/    texto2* texto7*  var/
dev/    lost+found/ opt/         pasta4/  pasta9/  sys/     texto3* texto8*
[root@localhost ~]# tree aluno
aluno
|-- pessoal
'-- trabalhos

2 directories, 0 files
[root@localhost ~]#
```

Nesse caso estamos criando a estrutura hierárquica toda de uma vez. Com o comando `ls`, podemos notar a criação da pasta `aluno`, e com o comando `tree`, podemos exibir as pastas `trabalhos` e `pessoal` dentro da pasta `aluno`.

6.6 Comando `rmdir`

Esse comando é usado para apagar apenas diretórios **vazios**. A opção `-p` permite remover toda a estrutura. Veja um exemplo:

```
[root@localhost ~]# rmdir carol
[root@localhost ~]# ls
aluno/  home/     media/         pasta10/ pasta6/  proc/    texto1* texto5*  tmp/
bin/    initrd/   mnt/          pasta2/  pasta7/  root/    texto10 texto6    usr/
boot/   lib/      ooobuildtime.log pasta3/  pasta8/  sbin/    texto2* texto7*  var/
dev/    lost+found/ opt/         pasta4/  pasta9/  sys/     texto3* texto8*
etc/    mandriva2009/ pasta1/      pasta5/  predio/  texto    texto4*  texto9
[root@localhost ~]#
```

Ainda utilizando a pasta criada no tópico 6.5, vamos apagá-la, utilizando `rmdir`. Logo na próxima linha podemos verificar que a pasta foi realmente removida.

Dica: Esse é um comando obsoleto e restrito. Para apagar geralmente utilizamos o comando `rm`.

6.7 Comando `rm`

O comando `rm` apaga diretórios cheios ou não.

Opções:

- f : formação a remoção;
- r : exclusão recursiva (apaga subdiretórios)

```
[root@localhost ~]# ls
aluno/  home/      media/      pasta10/  pasta6/  proc/  texto1*  texto5*  tmp/
bin/    initrd/    mnt/        pasta2/   pasta7/  root/  texto10  texto6   usr/
boot/   lib/       ooobuildtime.log  pasta3/   pasta8/  sbin/  texto2*  texto7*  var/
dev/    lost+found/ opt/        pasta4/   pasta9/  sys/   texto3*  texto8*
etc/    mandriva2009/ pasta1/     pasta5/   predio/  texto  texto4*  texto9
[root@localhost ~]# rm -rf aluno
[root@localhost ~]# ls
bin/    initrd/    mnt/        pasta2/   pasta7/  root/  texto10  texto6   usr/
boot/   lib/       ooobuildtime.log  pasta3/   pasta8/  sbin/  texto2*  texto7*  var/
dev/    lost+found/ opt/        pasta4/   pasta9/  sys/   texto3*  texto8*
etc/    mandriva2009/ pasta1/     pasta5/   predio/  texto  texto4*  texto9
home/   media/     pasta10/    pasta6/   proc/    texto1*  texto5*  tmp/
[root@localhost ~]# _
```

Podemos notar no exemplo acima, a exclusão da pasta /aluno e todos os subdiretórios.

Portanto a **sintaxe** é:

rm <opções> <nomedoarquivo>

6.8. Comando cp

Copia arquivos e pastas.

Opções:

- r : copia subdiretórios e seus arquivos
- f: força a substituição dos arquivos.

```
[root@localhost ~]# cp /textos/teste /BACKUP
[root@localhost ~]# ls /BACKUP
teste
[root@localhost ~]# _
```

Nesse exemplo estamos copiando da pasta texto, um documento chamado teste para a pasta BACKUP.

A sintaxe é:

cp <opção> <origem> <destino>

6.9 Comando mv

É usado para copiar e renomear arquivos e diretórios.

A **sintaxe** é:

mv <origem> <destino> - para mover

mv <nomevelho> <nomenovo> - para renomear

Exemplo:

```
[root@localhost ~]# mv texto /BACKUP
[root@localhost ~]# ls /BACKUP
teste  texto
[root@localhost ~]# ls
BACKUP/  home/      media/      pasta10/  pasta6/  proc/      texto10  texto6  tmp/
bin/     initrd/    mnt/        pasta2/   pasta7/  root/      texto2*  texto7*  usr/
boot/    lib/       ooobuildtime.log  pasta3/   pasta8/  sbin/      texto3*  texto8*  var/
dev/     lost+found/ opt/        pasta4/   pasta9/  sys/       texto4*  texto9
etc/     mandriva2009/ pasta1/     pasta5/   predio/  texto1*    texto5*  textos/
[root@localhost ~]#
```

Agora estamos movendo um arquivo da raiz, chamado texto para a pasta BACKUP. Observe que na linha seguinte, não vemos esse arquivo na raiz.

Outro exemplo:

```
[root@localhost ~]# mv BACKUP backup
[root@localhost ~]# ls
backup/  home/      media/      pasta10/  pasta6/  proc/      texto10  texto6  tmp/
bin/     initrd/    mnt/        pasta2/   pasta7/  root/      texto2*  texto7*  usr/
boot/    lib/       ooobuildtime.log  pasta3/   pasta8/  sbin/      texto3*  texto8*  var/
dev/     lost+found/ opt/        pasta4/   pasta9/  sys/       texto4*  texto9
etc/     mandriva2009/ pasta1/     pasta5/   predio/  texto1*    texto5*  textos/
[root@localhost ~]# _
```

Agora estamos renomeando a pasta BACKUP para backup.

Dica: Perceba que o linux diferencia letras minúsculas de MAIÚSCULAS.

6.10 – Comando cut

```
[root@localhost ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/sh
daemon:x:2:2:daemon:/sbin:/bin/sh
adm:x:3:4:adm:/var/adm:/bin/sh
lp:x:4:7:lp:/var/spool/lpd:/bin/sh
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/bin/sh
news:x:9:13:news:/var/spool/news:/bin/sh
uucp:x:10:14:uucp:/var/spool/uucp:/bin/sh
operator:x:11:0:operator:/var:/bin/sh
games:x:12:100:games:/usr/games:/bin/sh
nobody:x:65534:65534:Nobody:/:/bin/sh
rpm:x:13:101:system user for rpm:/var/lib/rpm:/bin/false
messagebus:x:14:105:system user for dbus:/:/sbin/nologin
polkituser:x:15:106:system user for policykit:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
haldaemon:x:70:70:system user for hal:/:/sbin/nologin
avahi:x:71:71:system user for avahi:/var/avahi:/bin/false
avahi-autoipd:x:72:72:system user for avahi:/var/avahi:/bin/false
htdig:x:73:107:/:/var/lib/htdig:
mysql:x:74:74:system user for mysql:/var/lib/mysql:/bin/bash
ftw:x:75:75:system user for drakfirsttime:/home/ftw:/bin/false
linux:x:501:501:/:/home/linux:/bin/bash
[root@localhost ~]# _
```

6.11 Comando VI

O vi é o editor de texto do shell.

```
[root@localhost ~]# vi teste.txt
```

```
"teste.txt" [Novo arquivo]                                0,0-1      Tudo
```

Podemos dividir o vi em dois:

Modo de INSERÇÃO ou SUBSTITUIÇÃO

```
-- INSERÇÃO --                                           0,1      Tudo
-- SUBSTITUIÇÃO --                                       0,1      Tudo
```

Modo **MENU** (pode ser acessado ao se pressionar a tecla **ESC** para sair do modo de inserção.

Comandos mais utilizados

:**e** Nome – Abre novo arquivo

:**w** _Salvar

:**q** _sair

:**q!** _sai sem salvar

Há um terceiro modo chamado de “**VISUAL**” que pode ser acessado através da tecla **V** (A partir do modo menu)

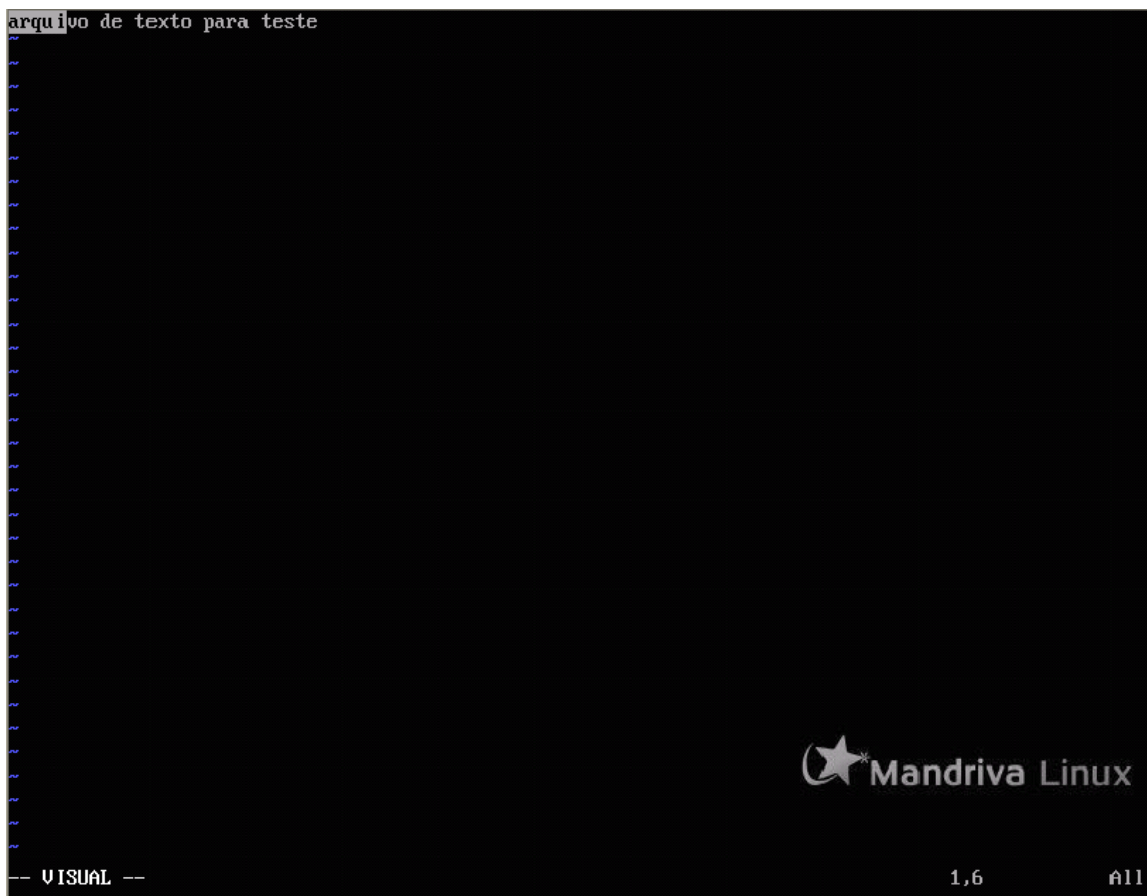
v: Marca texto

y : Cópia

d : recorta

p : cola

u : desfazer



7. DESLIGANDO O SISTEMA

7.1 Comandos Utilizados pelo usuário comum:

- **Ctrl+Alt+Del**

- **Halt** : Delisga a máquina

- **REBOOT** : Reboota a máquina

7.2 Comando utilizado pelo ROOT

Para desligar o sistema, o ROOT pode utilizar todos os comandos citados acima, e dispõe de um comando que é usado mais para fins administrativos.

Sintaxe:

SHUTDOWN <opção> <horário> <mensagem opcional>

Opções

-h : desliga a maquina


```
[root@localhost ~]# shutdown -h 18:00 "o sistema sera desligado"
```

-r : reinicia a maquina

Horário

Now : agora

00:00 : 23:59

Exemplo

Shutdown -r now

8.0 COMANDOS DE BUSCA

8.1 Comando FIND

Encontra arquivos no Linux. Sua sintaxe é:

FIND <onde><opções><o que>

“onde” → diretório a ser pesquisado

“opção” As mais comuns são:

-name → nome

-user → usuário

Ex:

Find / -name *.conf

```
[root@localhost ~]# find / *txt
```

8.2 Comando GREP

Filtra 'strings' e expressões em arquivos e exibe na tela. Permite filtrar um conteúdo em texto com base em uma expressão. É muito utilizado com o pipe (|). Sua sintaxe é:

grep 'expressão' <entrada>

Exemplos:

grep 'linux' /etc/passwd

cat /etc/passwd | grep [a-z]in

```
[root@localhost ~]# cat /etc/passwd | grep [a-z]lin
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/sh
daemon:x:2:2:daemon:/sbin:/bin/sh
adm:x:3:4:adm:/var/adm:/bin/sh
lp:x:4:7:lp:/var/spool/lpd:/bin/sh
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/bin/sh
news:x:9:13:news:/var/spool/news:/bin/sh
uucp:x:10:14:uucp:/var/spool/uucp:/bin/sh
operator:x:11:0:operator:/var:/bin/sh
games:x:12:100:games:/usr/games:/bin/sh
nobody:x:65534:65534:Nobody:/:/bin/sh
rpm:x:13:101:system user for rpm:/var/lib/rpm:/bin/false
messagebus:x:14:105:system user for dbus:/:/sbin/nologin
polkituser:x:15:106:system user for policykit:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
haldaemon:x:70:70:system user for hal:/:/sbin/nologin
avahi:x:71:71:system user for avahi:/var/avahi:/bin/false
avahi-autoipd:x:72:72:system user for avahi:/var/avahi:/bin/false
mysql:x:74:74:system user for mysql:/var/lib/mysql:/bin/bash
ftw:x:75:75:system user for drakfirsttime:/home/ftw:/bin/false
linux:x:501:501:/home/linux:/bin/bash
aluno:x:502:502:/home/aluno:/bin/bash
named:x:76:76:system user for bind:/var/lib/named:/bin/false
[root@localhost ~]#
```

9. LINKS

Apesar de o princípio ser o mesmo, a criação dos atalhos no Linux, são totalmente diferentes do Windows. Temos dois tipos de links:

9.1 Links Simbólicos

Funciona como um atalho é apenas uma referência ao outro arquivo. Se o arquivo for apagado, o link ficará inútil. Um atalho do arquivo origem.

```
[root@localhost ~]# ls
Desktop/ Documentos/ drakx/ Música/ proved.txt tmp/ Vídeo/
diversos/ Download/ Fotos/ provedores.txt texto.txt usuarios.txt
[root@localhost ~]# ln -s provedores.txt provedor.txt
[root@localhost ~]# ls
Desktop/ Documentos/ drakx/ Música/ provedor.txt texto.txt usuarios.txt
diversos/ Download/ Fotos/ provedores.txt proved.txt tmp/ Vídeo/
[root@localhost ~]# _
```

9.2 Links Físicos

É a “criação” de outro arquivo independente do original, mas que possui uma ligação

simbiótica com o primeiro. É como se um arquivo tivesse dois nomes.

```
[root@localhost ~]# ls
Desktop/ Documentos/ drakx/ Música/ texto.txt usuarios.txt
diversos/ Download/ Fotos/ provedores.txt tmp/ Vídeo/
[root@localhost ~]# ln provedores.txt proved.txt
[root@localhost ~]# ls
Desktop/ Documentos/ drakx/ Música/ proved.txt tmp/ Vídeo/
diversos/ Download/ Fotos/ provedores.txt texto.txt usuarios.txt
[root@localhost ~]# _
```

9.3 Comando ln

Ln <opção><arquivo><link>

Opção:

-s → cria um link simbólico.

**quando omitido o “-s” cria-se um link físico*

Arquivo: localização do arquivo que queremos linkar.

Link: O nome do link que será criado

10. DISPOSITIVOS NO LINUX

A maioria dos dispositivos no Windows são reconhecidos por letras (por exemplo: **A:** , **C:** , **D:** , etc). Já no Linux é diferente.

Veja abaixo:

HD → Disco IDE

SD → Disco SCSI / SATA

A → Primeira porta máster

B → Primeira porta slave

C → Segunda porta máster

1 → Primeira partição

2 → Segunda partição

3 → Terceira partição

4 → Quarta partição

5 → Primeira partição lógica

6 → Segunda partição lógica

7 → Terceira partição lógica

Outros

FD0 / FD1 →Referem-se ao disquete (floppy disk) em duas portas diferentes.
CDROM→ Refere-se ao drive de CD, independente de que porta ele esta instalado.
(Mesmo que /DEV/HDC)

10.1 Montando dispositivos

No Linux, para acessar o conteúdo dos dispositivos antes é obrigatório sua montagem.
Normalmente, a montagem é feita no diretório dele.

Comando:

Mount <opção><dispositivo> <pasta>

Opção:

-t <sistema de arquivos>

Exemplos

Mount -t vfat /dev/hdb1 /mnt/x

Mount -t iso 9660 /dev/hdc /mnt/cdrom

Mount /dev/hdc /mnt/cdrom

Mount→ mostra os dispositivos já montados

10.2 Desmontando dispositivos

Umount <pasta ou dispositivo>

Exemplos:

Umount /dev/hdc

Umount /mnt/cdrom

Umount -a →Desmonta todos os dispositivos montados.

10.3 Formatando disquete

10.3.1 Criando sistema de arquivos

**obs: para formatar o disquete, este deve estar desmontado.*

FDFORMAT<dispositivo>

Fdformat /dev/fd0

Formata o disquete destruindo todos os dados. Não cria o sistema de arquivos.

A opção -n permite formatação mais rápida não fazendo verificação nas trilhas.

```
[root@localhost ~]# fdformat -n /dev/fd0
Dupla faces, 80 trilhas, 14 setores/trilha. Capacidade total de 1120 Kb.
Formatando...
```

MKE2FS <dispositivo>

Exemplo→ MKE2FS /dev/fd0

Cria o sistema de arquivo EXT2 no dispositivo.

11. PERMISSÕES NO LINUX

O sistema operacional Linux trabalha com permissões relacionadas a três “sujeitos”. São eles:

u→Dono

g→Grupo(do dono)

o→Outros

Há uma forma de trabalhar com os três sujeitos ao mesmo tempo:

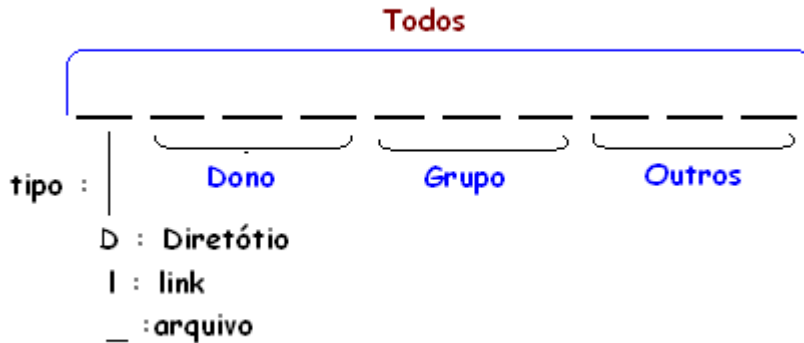
a→ Todos

As permissões existentes no Linux são:

r→leitura

w→gravação

x→execução



11.1 Mudando Permissões

CHMOD <permissão><arquivo>

Permissão → A nova permissão que será adicionada ou removida.

Arquivo → A arquivo/pasta que terá suas permissões alteradas.

12. Chown → Modifica usuário (e grupo) de um arquivo ou pasta.

Sintaxe:

CHOWN<usuário:grupo><arquivo>

Exemplo:

Chown linux /root

13. Usuários e Grupos

Comandos:

Adduser: Adiciona usuário no Linux
(o comando useradd também pode ser usado)

```
Mandriva 2009 - Microsoft Virtual PC 2007
Action Edit CD Floppy Help
[root@localhost etc]# adduser carolina
[root@localhost etc]# tail -3 passwd
ftw:x:75:75:system user for drakfirsttime:/home/ftw:/bin/false
linux:x:501:501::/home/linux:/bin/bash
carolina:x:502:502::/home/carolina:/bin/bash
[root@localhost etc]#
```

Groupadd: Adiciona grupos no Linux

```
Mandriva 2009 - Microsoft Virtual PC 2007
Action Edit CD Floppy Help
[root@localhost etc]# groupadd isis
[root@localhost etc]# tail -3 group
linux:x:501:
carolina:x:502:
isis:x:503:
[root@localhost etc]# _
```

Passwd: troca a senha de um usuário

```
Mandriva 2009 - Microsoft Virtual PC 2007
Action Edit CD Floppy Help
[root@localhost etc]# passwd carolina
Changing password for user carolina.
Nova UNIX senha:
Redigite a nova UNIX senha:
passwd: all authentication tokens updated successfully.
[root@localhost etc]#
```

Userdel: Apaga um usuário.

Opção:

-r: Apaga o diretório Home do usuário

```
Mandriva 2009 - Microsoft Virtual PC 2007
Action Edit CD Floppy Help
[root@localhost etc]# userdel -r carolina
[root@localhost etc]# tail -3 passwd
mysql:x:74:74:system user for mysql:/var/lib/mysql:/bin/bash
:x:75:75:system user for drakfirsttime:/home/ftw:/bin/false
linux:x:501:501::/home/linux:/bin/bash
[root@localhost etc]#
```

Groupdel: Apaga um grupo

```
Mandriva 2009 - Microsoft Virtual PC 2007
Action Edit CD Floppy Help
[root@localhost etc]# groupdel isis
[root@localhost etc]# tail -3 group
ftw:x:75:
linux:x:501:
carolina:x:502:
[root@localhost etc]# _
```

Groupmod: Modifica nome do grupo

Opção:

-n: Modifica o grupo especificado

```
Mandriva 2009 - Microsoft Virtual PC 2007
Action Edit CD Floppy Help
[root@localhost etc]# groupmod -n carol isis
[root@localhost etc]# tail -3 group
linux:x:501:
carolina:x:502:
carol:x:503:
[root@localhost etc]# _
```

Usermod: Adiciona usuário ao grupo

Opção:

-G: Indica o nome do grupo que o usuário será adicionado

```
Mandriva 2009 - Microsoft Virtual PC 2007
Action Edit CD Floppy Help
[root@localhost etc]# usermod -G carol carolina
[root@localhost etc]# tail -3 passwd
ftw:x:75:75:system user for drakfirsttime:/home/ftw:/bin/false
linux:x:501:501:::/home/linux:/bin/bash
carolina:x:502:502:::/home/carolina:/bin/bash
[root@localhost etc]#
```

16. Processos

Bg: força um processo suspenso a ser executado em segundo plano

Jobs: Permite visualizar jobs em execução

Split: Fragmenta arquivos em vários segmentos

Join: Junta os arquivos cortados linha por linha

Dv : Fornece o tamanho do sub diretório

Ps : Lista processos em andamento

Kill: Encerra processos em andamento

Top: exibe a lista dos processos
Who: Mostra quem esta usando o sistema
Mkfs: constrói um sistema de arquivo

Trabalhando com processos

Trabalhar com processos no linux envolve alguns aspectos. É necessário saber

- Iniciar e parar processos
- Rodar processos
- visualizar e encerrar processos

Executando um processo em segundo plano

Comando **&**

Ex: vi teste.txt &

```
[root@localhost ~]# vi teste.txt &  
[4] 2795  
[root@localhost ~]# _
```

Passar um processo de primeiro plano (foreground) para segundo plano (background)

Basta digitar CTRL + Z

```

"texto.txt" [sem fim de linha] 2L, 63C                               1,1          Tudo
[5]+  Stopped                  vi texto.txt
[root@localhost ~]#

```

Como listar os processos em background

Comando **JOBS**

```
[root@localhost ~]# jobs
[1]  Stopped                  cat /etc/group | less
[2]  Stopped                  cat /etc/group | less
[3]  Stopped                  cat /etc/group | less
[4]- Stopped                  vi teste.txt
[5]+ Stopped                  vi texto.txt
[root@localhost ~]#
```

Como trazer o processo para o primeiro plano

FG<número do JOBS>

```
[root@localhost ~]# fg 4
```



Listando os processos em execução na memória

PS [-A]

PS

```
[root@localhost ~]# ps
  PID TTY          TIME CMD
 2644 tty1        00:00:00 bash
 2785 tty1        00:00:00 less
 2788 tty1        00:00:00 less
 2793 tty1        00:00:00 less
 2795 tty1        00:00:00 vi
 2798 tty1        00:00:00 vi
 2812 tty1        00:00:00 ps
[root@localhost ~]# _
```

Opção

-A : Exibe todos os processos, inclusive aqueles iniciados pelo sistema.

```
[root@localhost ~]# ps -a
  PID TTY          TIME CMD
 2785 tty1      00:00:00 less
 2788 tty1      00:00:00 less
 2793 tty1      00:00:00 less
 2795 tty1      00:00:00 vi
 2798 tty1      00:00:00 vi
 2810 tty1      00:00:00 ps
[root@localhost ~]#
```

TOP : Mostra os processos com atualizações em tempo real ; digite CTRL+C para sair.

```
top - 19:12:12 up 7 min,  1 user,  load average: 0.14, 0.42, 0.29
Tasks: 42 total,  1 running, 40 sleeping,  1 stopped,  0 zombie
Cpu(s):  0.4% us,  2.7% sy,  0.0% ni, 87.9% id,  0.0% wa,  0.0% hi,  9.1% si
Mem:   233584k total,   81036k used,  152548k free,   31772k buffers
Swap:  1124508k total,    0k used,  1124508k free,   26112k cached
```

PID	USER	PR	NI	VRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3652	root	16	0	2004	1008	804	R	1.6	0.4	0:04.12	top
2897	root	16	0	4008	2600	1656	S	0.7	1.1	0:06.38	hald
3	root	10	-5	0	0	0	S	0.3	0.0	0:01.03	events/0
3081	root	16	0	2800	684	584	S	0.3	0.3	0:02.62	nifd
1	root	16	0	1564	532	472	S	0.0	0.2	0:04.10	init
2	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
4	root	10	-5	0	0	0	S	0.0	0.0	0:00.25	khelper
5	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kthread
7	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kacpid
60	root	10	-5	0	0	0	S	0.0	0.0	0:00.26	kblockd/0
90	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pdf lush
91	root	15	0	0	0	0	S	0.0	0.0	0:00.35	pdf lush
93	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	aio/0
92	root	22	0	0	0	0	S	0.0	0.0	0:00.00	kswapd0
684	root	16	0	0	0	0	S	0.0	0.0	0:00.07	kseriod
765	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	ata/0
772	root	15	0	0	0	0	S	0.0	0.0	0:00.73	kjournald
914	root	11	-4	1564	484	420	S	0.0	0.2	0:01.10	udevd
1594	root	21	0	0	0	0	S	0.0	0.0	0:00.00	kjournald
2488	root	16	0	1584	524	452	S	0.0	0.2	0:00.46	ifplugd
2571	rpc	15	0	1692	592	492	S	0.0	0.3	0:00.03	portmap
2594	root	16	0	1608	640	524	S	0.0	0.3	0:00.51	syslogd
2602	root	15	0	2336	1348	464	S	0.0	0.6	0:00.31	klogd
2684	root	22	0	1548	592	520	S	0.0	0.3	0:00.01	acpid
2729	root	18	0	1692	732	636	S	0.0	0.3	0:00.08	rpc.statd
2849	xf	18	0	3920	2104	932	S	0.0	0.9	0:00.18	xf
2873	messageb	16	0	2220	1144	988	S	0.0	0.5	0:00.58	dbus-daemon-1
3142	nobody	16	0	11420	1140	688	S	0.0	0.5	0:00.18	mDNSResponder
3167	daemon	22	0	1688	616	520	S	0.0	0.3	0:00.01	atd
3248	root	20	0	4272	1656	1364	S	0.0	0.7	0:00.05	sshd



```
top - 19:12:25 up 7 min, 1 user, load average: 0.11, 0.40, 0.28
Tasks: 43 total, 1 running, 41 sleeping, 1 stopped, 0 zombie
Cpu(s): 0.4% us, 3.5% sy, 0.0% ni, 83.4% id, 0.0% wa, 0.0% hi, 12.7% si
Mem: 233584k total, 81540k used, 152044k free, 31772k buffers
Swap: 1124508k total, 0k used, 1124508k free, 26112k cached
```

PID	USER	PR	NI	VRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3652	root	16	0	2004	1008	804	R	1.6	0.4	0:04.28	top
3081	root	16	0	2800	684	584	S	1.0	0.3	0:02.72	nifd
3	root	10	-5	0	0	0	S	0.3	0.0	0:01.05	events/0
772	root	16	0	0	0	0	S	0.3	0.0	0:00.74	kjournald
2897	root	16	0	4008	2600	1656	S	0.3	1.1	0:06.42	hald
1	root	16	0	1564	532	472	S	0.0	0.2	0:04.11	init
2	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
4	root	10	-5	0	0	0	S	0.0	0.0	0:00.25	khelper
5	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kthread
7	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kacpid
60	root	10	-5	0	0	0	S	0.0	0.0	0:00.26	kblockd/0
90	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pdf lush
91	root	15	0	0	0	0	S	0.0	0.0	0:00.35	pdf lush
93	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	aio/0
92	root	22	0	0	0	0	S	0.0	0.0	0:00.00	kswapd0
684	root	16	0	0	0	0	S	0.0	0.0	0:00.07	kseriod
765	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	ata/0
914	root	11	-4	1564	484	420	S	0.0	0.2	0:01.10	udevd
1594	root	21	0	0	0	0	S	0.0	0.0	0:00.00	kjournald
2488	root	16	0	1584	524	452	S	0.0	0.2	0:00.47	ifplugd
2571	rpc	15	0	1692	592	492	S	0.0	0.3	0:00.03	portmap
2594	root	16	0	1608	640	524	S	0.0	0.3	0:00.51	syslogd
2602	root	15	0	2336	1348	464	S	0.0	0.6	0:00.31	klogd
2684	root	22	0	1548	592	520	S	0.0	0.3	0:00.01	acpid
2729	root	18	0	1692	732	636	S	0.0	0.3	0:00.08	rpc.statd
2849	xfs	18	0	3920	2104	932	S	0.0	0.9	0:00.18	xfs
2873	messageb	16	0	2220	1144	988	S	0.0	0.5	0:00.58	dbus-daemon-1
3142	nobody	16	0	11420	1140	688	S	0.0	0.5	0:00.18	mDNSResponder
3167	daemon	22	0	1688	616	520	S	0.0	0.3	0:00.01	atd
3248	root	20	0	4272	1656	1364	S	0.0	0.7	0:00.05	sshd



Kill :” Mata” um processo

Modo de usar:

KILL<numero do pid>

Caso o programa não seja encerrado pode-se forçar o encerramento deste

Opção:

-9 :

Ex: KILL -9

```

[root@localhost ~]# ps
  PID TTY          TIME CMD
 3366 tty1      00:00:00 bash
 3408 tty1      00:00:00 vi
 3419 tty1      00:00:00 vi
 3422 tty1      00:00:00 ps
[root@localhost ~]# kill 3408
[root@localhost ~]# ps
  PID TTY          TIME CMD
 3366 tty1      00:00:00 bash
 3408 tty1      00:00:00 vi
 3419 tty1      00:00:00 vi
 3423 tty1      00:00:00 ps
[root@localhost ~]# kill -9 3408
[root@localhost ~]# ps
  PID TTY          TIME CMD
 3366 tty1      00:00:00 bash
 3419 tty1      00:00:00 vi
 3424 tty1      00:00:00 ps
11- Morto                                vi teste.txt
[root@localhost ~]# _

```

Informações sobre memória

DF : Mostra informações sobre as partições

```

[root@localhost ~]# df
Sist. Arq.          Tam  Usad Disp  Uso% Montado em
/dev/hda1           5,8G  2,6G  3,0G   46% /
/dev/hda6           9,0G  129M  8,9G    2% /home
[root@localhost ~]# _

```

Free: Mostra informações sobre o uso da memória RAM

```

[root@localhost ~]# free
              total        used        free       shared    buffers     cached
Mem:          233584        81160       152424           0        31776       26232
-/+ buffers/cache:        23152       210432
Swap:         1124508           0       1124508
[root@localhost ~]# _

```

DU [-s] <pasta>

Exibe o tamanho de todos os arquivos de uma pasta

```

[root@localhost ~]# du /etc

```

```

28K      /etc/rpm
8,0K     /etc/sysconfig/console/consolefonts
12K      /etc/sysconfig/console/consoletrans
44K      /etc/sysconfig/console
8,0K     /etc/sysconfig/network-scripts/hostname.d
4,0K     /etc/sysconfig/network-scripts/ifdown.d
4,0K     /etc/sysconfig/network-scripts/ifup.d
4,0K     /etc/sysconfig/network-scripts/wireless.d
192K     /etc/sysconfig/network-scripts
4,0K     /etc/sysconfig/networking/devices
4,0K     /etc/sysconfig/networking/profiles/default
8,0K     /etc/sysconfig/networking/profiles
4,0K     /etc/sysconfig/networking/tmp
24K      /etc/sysconfig/networking
12K      /etc/sysconfig/harddrake2
44K      /etc/sysconfig/suspend-scripts/suspend.d
52K      /etc/sysconfig/suspend-scripts
412K     /etc/sysconfig
16K      /etc/ld.so.conf.d
4,0K     /etc/skel/tmp
20K      /etc/skel
8,0K     /etc/default
24K      /etc/security/console.apps
16K      /etc/security/msec
72K      /etc/security
8,0K     /etc/ssl/webmin
12K      /etc/ssl
92K      /etc/pam.d
228K     /etc/rc.d/init.d
4,0K     /etc/rc.d/rc0.d
4,0K     /etc/rc.d/rc1.d

```

-s : sumário

```

[root@localhost ~]# du -s /etc
23M      /etc
[root@localhost ~]# _

```

Visualizando os usuários conectados

Who (sem argumentos)

```

[root@localhost ~]# who
root      tty1          Jun 19 19:08
[root@localhost ~]# _

```

Alterar a data de criação e modificação do arquivo ou pasta para a atual:

TOUCH<ARQUIVO>

```

[root@localhost ~]# ls
Desktop/ Documentos/ Download/ drakx/ Fotos/ Música/ tmp/ Vídeo/
[root@localhost ~]# ls -l
total 32
drwx----- 2 root root 4096 Abr 2 17:23 Desktop/
drwxr-xr-x 2 root root 4096 Abr 2 16:54 Documentos/
drwxr-xr-x 2 root root 4096 Abr 2 16:54 Download/
drwx----- 2 root root 4096 Out 6 2006 drakx/
drwxr-xr-x 2 root root 4096 Abr 2 16:54 Fotos/
drwxr-xr-x 2 root root 4096 Abr 2 16:54 Música/
drwx----- 5 root root 4096 Abr 2 16:55 tmp/
drwxr-xr-x 2 root root 4096 Abr 2 16:54 Vídeo/
[root@localhost ~]# touch tmp
[root@localhost ~]# ls -l
total 32
drwx----- 2 root root 4096 Abr 2 17:23 Desktop/
drwxr-xr-x 2 root root 4096 Abr 2 16:54 Documentos/
drwxr-xr-x 2 root root 4096 Abr 2 16:54 Download/
drwx----- 2 root root 4096 Out 6 2006 drakx/
drwxr-xr-x 2 root root 4096 Abr 2 16:54 Fotos/
drwxr-xr-x 2 root root 4096 Abr 2 16:54 Música/
drwx----- 5 root root 4096 Jun 19 19:47 tmp/
drwxr-xr-x 2 root root 4096 Abr 2 16:54 Vídeo/
[root@localhost ~]# _

```

Exibindo calendário:

CAL → Exibe o mês atual

```

[root@localhost ~]# cal
      junho 2009
Su Mo Tu We Th Fr Sa
      1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30

[root@localhost ~]# _

```

CAL 2010 → Exibe o calendário de todo o ano de 2010.


```
[root@localhost ~]# cal 2010
2010
janeiro      fevereiro      março
Do Se Te Qu Qu Se Sá Do Se Te Qu Qu Se Sá Do Se Te Qu Qu Se Sá
    1 2          1 2 3 4 5 6    1 2 3 4 5 6
 3 4 5 6 7 8 9    7 8 9 10 11 12 13    7 8 9 10 11 12 13
10 11 12 13 14 15 16 14 15 16 17 18 19 20    14 15 16 17 18 19 20
17 18 19 20 21 22 23 21 22 23 24 25 26 27    21 22 23 24 25 26 27
24 25 26 27 28 29 30 28    28 29 30 31
31

abril        maio        junho
Do Se Te Qu Qu Se Sá Do Se Te Qu Qu Se Sá Do Se Te Qu Qu Se Sá
    1 2 3          1          1 2 3 4 5
 4 5 6 7 8 9 10    2 3 4 5 6 7 8    6 7 8 9 10 11 12
11 12 13 14 15 16 17 9 10 11 12 13 14 15    13 14 15 16 17 18 19
18 19 20 21 22 23 24 16 17 18 19 20 21 22    20 21 22 23 24 25 26
25 26 27 28 29 30    23 24 25 26 27 28 29 27 28 29 30
30 31

julho        agosto        setembro
Do Se Te Qu Qu Se Sá Do Se Te Qu Qu Se Sá Do Se Te Qu Qu Se Sá
    1 2 3          1 2 3 4 5 6 7    1 2 3 4
 4 5 6 7 8 9 10    8 9 10 11 12 13 14    5 6 7 8 9 10 11
11 12 13 14 15 16 17 15 16 17 18 19 20 21    12 13 14 15 16 17 18
18 19 20 21 22 23 24 22 23 24 25 26 27 28    19 20 21 22 23 24 25
25 26 27 28 29 30 31 29 30 31    26 27 28 29 30

outubro      novembro      dezembro
Do Se Te Qu Qu Se Sá Do Se Te Qu Qu Se Sá Do Se Te Qu Qu Se Sá
    1 2          1 2 3 4 5 6    1 2 3 4
 3 4 5 6 7 8 9    7 8 9 10 11 12 13    5 6 7 8 9 10 11
10 11 12 13 14 15 16 14 15 16 17 18 19 20    12 13 14 15 16 17 18
17 18 19 20 21 22 23 21 22 23 24 25 26 27    19 20 21 22 23 24 25
24 25 26 27 28 29 30 28 29 30    26 27 28 29 30 31
31
```



CAL 04 2014→Exibe os mês de abril de 2014.

```
[root@localhost ~]# cal 04 2014
abril 2014
Do Se Te Qu Qu Se Sá
    1 2 3 4 5
 6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

[root@localhost ~]#
```

Ver e alterar a data e hora do sistema.

Opções

-s→Muda a hora

-d→Muda a data

Contando arquivos de texto:

WC<argumento><arquivo>

Opções

-m→Conta caracteres

```
[root@localhost ~]# wc -m /etc/group
467 /etc/group
[root@localhost ~]#
```

-l→conta linhas

```
[root@localhost ~]# wc -l /etc/group
39 /etc/group
[root@localhost ~]# _
```

-w→conta palavras

```
[root@localhost ~]# wc -w /etc/group
39 /etc/group
[root@localhost ~]# _
```

Instalando e Removendo Pacotes (Debian-like)

No Kali (e distribuições baseadas em Debian em geral), a forma mais simples de se instalar um programa ou remove-lo é através do conjunto de comandos APT.

Instalar Pacote

apt-get install <pacote>

Instala um novo pacote:

Ex: apt-get install metasploit

Remover Pacote

apt-get remove <pacote>

Remove pacote instalado

Ex: apt-get remove nmap

A opção **-purge** permite apagar arquivos adicionais instalados.

Ex: apt-get -purge remove nmap

Pesquisar nome de Pacote

apt-cache search <pacote> → Pesquisa os pacotes com o nome

Ex: apt-cache search ssl

Atualizando a lista de pacotes:

apt-get update

Atualizando os pacotes/sistema:

apt-get upgrade

apt-get dist-upgrade

Vendo Informações no Sistema

Uname<opção>

Opção

-a→Mostra todos os dados

```
[root@localhost ~]# uname -a
Linux localhost 2.6.12-12mdk #1 Fri Sep 9 18:15:22 CEST 2005 i686 Intel(R) Pentium(R) 4 CPU 2.40GHz
unknown GNU/Linux
[root@localhost ~]#
```

-r→Mostra o número do
Kernel

```
[root@localhost ~]# uname -r
2.6.12-12mdk
[root@localhost ~]# _
```

-o→Mostra o S.O

```
[root@localhost ~]# uname -o
GNU/Linux
[root@localhost ~]# _
```

-m→Mostra a arquitetura da máquina

```
[root@localhost ~]# uname -m
i686
[root@localhost ~]# _
```

-p→Mostra o processador

```
[root@localhost ~]# uname -p
Intel(R) Pentium(R) 4 CPU 2.40GHz
[root@localhost ~]#
```

Vendo o histórico de comandos digitados

HISTORY<número>

```
[root@localhost ~]# history _
```

```
1  kde
2  logout
3  clear
4  vi teste.txt &
5  clear
6  top
7  clear
8  jobs
9  kill 1
10 jobs
11 pid
12 clear
13 ps
14 kill 3649
15 ps
16 kill -g 3649
17 kill -g3649
18 kill -G 3649
19 kill 3649
20 ps
21 kill -g 3649
22 kill _g 3649
23 kill -f 3649
24 kill 3649 -g
25 kill 3649
26 ps
27 clear
28 df
29 clear
30 free
31 clear
```

Numero→Quantos comandos digitados você quer ver.

```
[root@localhost ~]# history 5
68 history lless
69 clear
70 history lless
71 clear
72 history 5
[root@localhost ~]#
```

Compactando e descompactando arquivos

TAR - <opção><arquivo compactado><dados>

Opções:

VF→Sempre utilizado."Força" a substituição de arquivos

C→compacta

X→descompacta

Z→compacta

Exemplos:

COMPACTANDO :

Tar -cvf temporarios.tar /tmp/gconfd-linux

```
[linux@localhost ~]$ ls -l /tmp
total 8
drwx----- 3 linux linux 4096 Mai 29 06:38 gconfd-linux/
drwx----- 2 root root 4096 Mai 29 06:37 gconfd-root/
[linux@localhost ~]$ tar -cvf temporarios.tar /tmp/gconfd-linux
tar: Removing leading '/' from member names
/tmp/gconfd-linux/
/tmp/gconfd-linux/lock/
/tmp/gconfd-linux/lock/ior
[linux@localhost ~]$ ls -l
total 40
drwx----- 2 linux linux 4096 Mai 29 06:39 Desktop/
drwxrwxr-x 2 linux linux 4096 Mai 29 06:37 Documentos/
drwxrwxr-x 2 linux linux 4096 Mai 29 06:38 Download/
drwxrwxr-x 2 linux linux 4096 Mai 29 06:38 Fotos/
drwxrwxr-x 2 linux linux 4096 Mai 29 06:38 Música/
-rw-rw-r-- 1 linux linux 10240 Jun 22 05:31 temporarios.tar
drwx----- 5 linux linux 4096 Mai 29 06:38 tmp/
drwxrwxr-x 2 linux linux 4096 Mai 29 06:38 Vídeo/
[linux@localhost ~]$
```

Pode-se também compactar em formato "gz" o que fará com que o arquivo fique ainda mais pequeno.

Tar -zcvf temporarios.tar.gz /tmp/gconfd-linux

```
[linux@localhost ~]$ tar -zcvf temporarios1.tar.gz /tmp/gconfd-linux
tar: Removing leading '/' from member names
/tmp/gconfd-linux/
/tmp/gconfd-linux/lock/
/tmp/gconfd-linux/lock/ior
[linux@localhost ~]$ ls -l
total 44
drwx----- 2 linux linux 4096 Mai 29 06:39 Desktop/
drwxrwxr-x 2 linux linux 4096 Mai 29 06:37 Documentos/
drwxrwxr-x 2 linux linux 4096 Mai 29 06:38 Download/
drwxrwxr-x 2 linux linux 4096 Mai 29 06:38 Fotos/
drwxrwxr-x 2 linux linux 4096 Mai 29 06:38 Música/
-rw-rw-r-- 1 linux linux 379 Jun 22 05:32 temporarios1.tar.gz
-rw-rw-r-- 1 linux linux 10240 Jun 22 05:31 temporarios.tar
drwx----- 5 linux linux 4096 Mai 29 06:38 tmp/
drwxrwxr-x 2 linux linux 4096 Mai 29 06:38 Vídeo/
[linux@localhost ~]$
```

DESCOMPACTANDO :

Tar -xvf textos.tar

Tar -zxvf textos.tar.gz

```
[root@localhost ~]# ls
aif-mount/      Download/      Imagens/      Modelos/      textos.tar      trabalho/
Área de Trabalho/ drakx/        marcos/      Música/      textos.tar.gz  Vídeos/
Documentos/     DrakX-screenshots/ meusemails@ processos*  tmp/
[root@localhost ~]# tar -xvf textos.tar
emails.txt
interfacegrafica.txt
linkhistoria.txt
provedores.txt
texto.txt
usuarios.txt
[root@localhost ~]# ls
aif-mount/      DrakX-screenshots/ marcos/      provedores.txt  trabalho/
Área de Trabalho/ emails.txt          meusemails@ textos.tar      usuarios.txt
Documentos/     Imagens/          Modelos/     textos.tar.gz   Vídeos/
Download/       interfacegrafica.txt Música/      texto.txt
drakx/          linkhistoria.txt   processos*    tmp/
[root@localhost ~]#
```