

GUIA RÁPIDO DE COMANDOS LINUX

GERAL

Shell de usuário \$
Shell de root #

login
inicia uma nova sessão

su
troca de usuário, ex
su root

logout, exit
encerra a sessão

reboot, init 6
reinicia o micro

halt, init 0
desliga o micro

shutdown
desliga ou reinicia, ex
shutdown -r now (reinicia agora)
shutdown -h 10 (desliga daqui a 10 min)

help
informações sobre comandos, ex
comando --help

man
consulta manual do comando, ex
man comando

whatis
consulta o que é o comando, ex
whatis comando

pwd
exibe o diretorio corrente

tree
exibe a árvore

ls
lista diretorios, ex
ls -lha (a - ocultos, l - detalhes, h - tamanho em kbytes)

clear
limpa tela

find
localiza arquivos, ex
find / -iname arquivo.txt (procura arq.txt em / pelo nome Maius ou menusc)

string
procura strings em arquivos binários, ex
string programa.exe

which
mostra o caminho de um programa/comando, ex
which firefox

touch
cria arquivos, ex
touch arquivo.txt

rm
remove arquivos, ex
rm arquivo.txt
rm -rf (recursivo e force, remove diretorios não vazios)

mkdir
cria diretórios, ex
mkdir pasta

rmdir
apaga diretorios vazios, ex
rmdir pasta

cp
copia, ex
cp -r (copia recursivamente)

ln
cria links simbólicos, ex
ln -s pasta link

mv
move ou renomeia arquivos e pastas, ex
mv pasta pasta2

cd
muda de diretorio, ex
cd ../ (vai para o diretorio anterior)

split
divide arquivos, ex
split -b 1440k pasta_grande.tar.gz (divide em tamanhos de disquetes)

cat
concatena arquivos e imprimir na saída padrão, ex
cat arquivo.txt (função mais usada)
cat xaa xab > pasta_restaurada.tar.gz (joiner)

file
exibe o tipo do arquivo, ex
file arquivo.txt

date
exibe e altera a data, ex
date 072117152012 (mes,dia,hora,min,ano)
Sab jul 21 17:15:00 BRT 2012

cal
exibe o calendário

uptime
exibe o tempo de execução

cat /etc/issue
exibe informações sobre a versão da distribuição

ARQUIVOS

cat, more, less
exibe arquivos

echo
escreve

| (pipe) - une comandos, ex
comando1 |comando2 |comando3 ...

grep
busca, ex
ls |grep arquivo

> (recirciona) - redireciona a saída, ex
cat > arquivo.txt (cria um arquivo para escrita, use CTRL+D para salvar)
echo "fim" >> arquivo.txt (>> escreve no fim do arquivo)

head
exibe n primeiras linhas de um arquivo, ex
head -n arquivo.txt (n - linha, b - blocos, c - caracteres)

tail
exibe n ultimas linhas de um arquivo, ex
tail -n arquivo.txt (n - linha, b - blocos, c - caracteres)

tr

substitui ou traduz caracteres, ex
ll /etc |tr " " ". " (troca espaços por traços)
cat arquivo.txt |tr -d " " (tira os espaços)

wc

conta, ex
wc -l arquivo.txt (l-linhas, w-palavras, c-caracteres)

nl

visualiza a numeração das linhas, ex
nl arquivo.txt

sort

ordena, ex
sort arquivo.txt

rev

inverte linhas de texto, ex
rev arquivo.txt

tac

inverte um arquivo, contrario do cat, ex
tac arquivo.txt

cut

seleciona trechos de uma arquivo, ex
cut -f 1 -d " " arquivo.txt
(seleciona o 1o campo com delimitador espaço)
cut -c 1-3 arquivo.txt
(seleciona os caracteres de 1 a 3)

diff

exibe a diferença de arquivos, ex
diff arquivo1 arquivo2

tee

grava a saída de um comando em um arquivo, ex
echo "fim" |tee -a arquivo (equivale >>)

sed, awk

veja folha anexa

EDITORES DE TEXTO

vim
pico, nano
mcedit

COMPACTAÇÃO

tar

compacta ou extrai, ex
tar -czvf arquivo.tar.gz PASTA (compacta)
tar -xvf arquivo.tar.gz (extrai)

zip, unzip

compacta, extrai, ex
zip arquivo.zip PASTA
unzip arquivo.zip

HARDWARE

arch

exibe a arquitetura do SO (32 ou 64)

lspci, lsdev, lsusb

lista dispositivos de hardware

discover

utilitário para detectar hardware

cat /proc/cpuinfo, meminfo, partitions, devices

informações sobre os dispositivos

dmesg

mensagens do kernel

INSTALAÇÃO

rpm -ivh pacote.rpm (instala)
dpkg -i pacote.deb (instala)
rpm -e pacote (desinstala)
dpkg -r pacote (desinstala)

yum install pacote
apt-get install pacote

yum remove pacote
apt-get remove pacote

yum update
apt-get update

VARIÁVEIS DE AMBIENTE

\$SHELL (/etc/shells), **\$HOME**, **\$LOGNAME**, **\$USER** ...

alias

cria apelidos, ex
alias lista "ls -lha"

history

histórico de comandos, ex
history -c (apaga histórico)

unalias

apaga apelidos
unalias lista

SERVIÇOS

Tabela: 0-halt , 1-monouser, 2-multiuser sem nfs, 3-multiuser com nfs, 4-reservado, 5-X11, 6-reboot ; Localização: /etc/rc?.d

RedHat

chkconfig --add serviço (adiciona)
chkconfig --del serviço (remove)
chkconfig --list serviço (lista)

ntsysv (gerencia)
service serviço start,stop,restart

Debian

update-rc.d nome-serviço defaults (adiciona)
update-rc.d -f nome-serviço remove (remove)

rcconf (gerencia)

cron
atumatiza, edite /etc/crontab

PROCESSOS

jobs -l

exibi os processos executados no terminal

& (executa em background)

CTRL+C - finaliza

CTRL+Z (interrompe, use bg,fg para alternar de background e foreground)

ps

lista processos ativos, ex
ps -aux |grep firefox

kill

mata processo, ex
kill -9 numero-processo

top - visualiza processos mais usados

free - memoria livre

USUÁRIOS

w - mostra quem está logado

white

envia uma mensagem para o usuário, ex
white everton tty1
ola aluno

wall

envia uma mensagem de texto a todos os usuários, ex
wall mensagem.txt

/etc/passwd

(usuario:senha:id-usuario:id-grupo:descrição:diretorio:shell)

/etc/group

(grupo:senha:id-grupo:descrição:lista-usuarios)

Ob:(para bloquear um usuario use #, para deletar apague a linha)

/etc/defaults/useradd

valores padrão para criação de usuarios

adduser, useradd

adiciona usuarios, ex:
useradd aluno -d /home/pasta -g grupo -s /bin/bash

usermod

modifica características do usuario, ex
usermod -c "comentario" -d /diretorio -s /shell -g grupo everton

userdel - deleta um usuario

passwd

modifica senha do usuario, ex
passwd everton

finger - exibe informações do usuário

groupadd, groupdel

adiciona, remove um grupo

gpasswd

define senha do grupo ou insere usuario em um grupo, ex
gpasswd -a everton familia (a-adiciona, d-remove)

groups

exibe informações do grupo

who

mostra quem esta logado

whoami

quem sou eu?

last

listagem dos ultimos logins e quem ainda esta on

lastlog

relatorio dos ultimos logins de todos os usuários

sudo

executa comandos de superusuario
/etc/sudoers (configurações do sudo)

PERMISSÕES

dono,grupo,outros (octal: **4-Read, 2-Write, 1-eXecute**), ex:
ls -l /home/usuario

drwxr-xr-x pasta

(padrão 777-022=755, ver **/etc/profile, umask 022**)

-rw-r--r-- arquivo

(padrão 666-022=644, ver **/etc/profile, umask 022**)

Obs: /home/usuario/.profile (padrão do usuario, **umask 022**)

chmod

altera as permissões dos arquivos ou pastas, ex
chmod 744 arquivo (torna rwx r-- r--)

chown

altera o dono, ex
chown -R everton.familia PASTA/

chgrp

altera so o grupo, ex
chgrp -R everton PASTA/

*** GUIA COMANDO SED ***

Opções:

-e sed estendido, para usar {}
-i gravar no mesmo arquivo
-n prox linha

Comandos básicos:

; separador de comandos
! não logico
, separador de endereços
/ separador de endereços
p imprimir
s substitui
c coloca em um intervalo
d deleta
g fazer em toda extensão

Escapes para caracteres especiais:

\a (apito)
\f (avança linha)
\n (quebra de linha)
\r (retorno de carro)
\t (tabulação horizontal)
\v (tabulação vertical)

A sintaxe genérica:

sed [opções] regras [arquivo]

Sendo que regras tem a forma genérica de:
[ende1 [, end2]] comando [argumento]

EXEMPLOS:

Imprimir duplicando:

sed p texto > saida

Imprimir sem duplicar a linha 5:

sed -n 5p texto > saida

Deleta linha:

sed '5d' texto > saida

Deleta linha que tem uma palavra

sed '/palavra/d' texto > saida

Intervalos:

sed '5,10d' texto > saida
sed '5,/palavra/d' texto > saida

Mais 3 linhas:

sed '/palavra/,+3d' texto > saida

Invertendo a logica:

sed '2,5!d' texto > saida

Aplicando vários comandos de uma vez:

sed '5d;10d;/palavra/d' texto > saida

Substitui:

sed 's/isso/aquilo/' texto > saida

Variavel, use aspas duplas:

sed "s/\$HOME/aquilo/" texto > saida

Delimitadores (uso da /):

sed 's/, ,g' texto > saida
sed 's/\ /g' texto > saida

Substituir por uma quebra de linha:

sed 's/ \n/g' texto > saida

Deleta espaços em branco final e inicial:

sed 's/^[\t]*//;s/[\t]*\$//' texto > saida

Trocar várias linhas ou um bloco:

sed '/<head>/,/<\thead>/c TEXTO'
index.html > saida.html

GUIA AWK

AWK é uma linguagem de programação para processamento de texto.

Sintaxe: `$ awk opcao 'PADRAO { ACAO }'`

EXEMPLOS:

Imprimir a 8a coluna do list e ainda deletar as 1as linhas:

```
ll | awk '{print $8}' | sed '1,3d'
```

Imprimir o 2o campo somente da linha q tem o padrao:

```
$ awk '/Everton/ { print $2 }' familia.txt
```

Usando um caracter como delimitador:

```
cat /etc/passwd | awk -F ":" '{print $1}'
```

Usando o separador de campos:

```
ls -l /etc | awk '{print $1 FS $8}'
```

```
ls -l /etc | awk '{print $1 " " $8}'
```

Numerando linhas:

```
ls -l /etc | awk '{print NR FS $1 FS $8}'
```

Filtra linhas com padrão e mostra apenas colunas 1 e 8:

```
ls -l /etc | awk '/conf$/{print $1 " "$8}'
```

Imprime as linhas com mais de 3 campos:

```
ls -l /etc | awk 'NF > 3'
```

Filtra linhas com caracteres da coluna 1 igual a 7:

```
awk 'length($1) == 7' familia.txt
```

Imprime linhas pares:

```
ls -l /etc | awk 'NR % 2 == 0 {print NR " "$0}'
```

Substitui strings:

```
ls -l /etc | awk '/conf$/ {sub(/conf$/, "test"); print $8}'
```

Procura expressão em determinado campo e imprime:

```
ls -l /etc | awk '$8 ~ /^[ae]/ {print $8}'
```

Inserindo strings entre campos:

```
cat /etc/passwd | awk -F: '{print "Login: " $1}'
```

NR != 1 evita que a linha Total seja exibida:

```
ls -lha /etc | awk 'NR != 1 {print "Nome: "$8 " Perm: "$1 " Tamanho: "$5}'
```

Imprime as linhas com mais de 27 caracteres:

```
awk 'length($0) > 27 {print $0}' familia.txt
```

Exibe o número de linhas do arquivo:

```
awk 'END { print NR }' arquivo
```

GUIA SCRIPT SHELL

Início:

```
#!/bin/bash
#!/bin/sh
```

Preparação:

```
chmod 755 script.sh
```

Execução:

```
./script.sh
sh script.sh
```

echo

imprimi, ex:
echo \$USER (imprimi o usuario)
echo ~ (imprimi o home)
echo \$HOME (imprimi o home)
echo \$SHELL (imprimi o shell)

Declaracao de Variáveis:

usa-se com \$, declara sem \$, ex
nome="Everton"
echo "A variavel \\${nome} contem a string \$nome"
echo "A variavel \\${nome} contem a string \${nome}"

Variaveis especiais:

echo \$0 (exibi o nome do proprio script)
echo \$_ (exibi o ultimo comando)
echo \$# (exibi o numero de parametros passados)
echo \$* (exibi os parametros, separados por espaços)

Aspas, ex:

```
echo 'Isso eh uma "string"'
echo "Isso eh uma 'outra string'"
```

Barra invertida \

desabilita metacaracteres, ex
echo ... seja a funcao f(x\) tal que ...
echo "Veja esse \\$ que nao eh variavel !!"

Ponto e virgula ;

separa comandos, ex:
\${ comando1;comando2 }

Crase ``

executa o comando, ex:
echo "seu pwd eh `pwd`"

Pipe |

utiliza saida de um comando como entrada do outro,
ex : \$ ps -aux |less

E &&

executa o segundo se o primeiro tiver sucesso, ex
comando1 && comando2

OU ||

executa o segundo se o primeiro não tiver sucesso.
Ex : comando1 || comando2

Redirecionamento do script, ex

```
FILE=saida.txt
cat >> $FILE << EOF (end of file)
```

Expansões, ex

```
echo Arquitetura i{386,486,586}
echo bo{c,lach}a
echo $((2+2))
numero=10;echo $((2+$numero))
```

Operadores aritmeticos

a+b, a-b, a*b, a/b, a%b, ex:
echo \$((9%2))
echo `expr 2 * 2` (\ para anular metacaracteres)
echo `expr 25 / 5` (atencao: para decimal, use bc)
echo "3 ^ 2" |bc
echo "sqrt(4)" |bc
echo sqrt(25) |bc

Comandos de texto:

-r -w -x verifica as permissões rwx
-z verifica se a string esta vazia
-d verifica se eh diretorio
-e verifica se o diretorio existe
-f verifica se o arquivo existe
-s verifica se tem tamanho maior q zero
-nt verifica se arq1 mais novo arq2
-ot verifica se arq1 mais velho arq2

Comandos condicionais e logicos:

-eq == (equivalent)
-ne != (not equivalent)
-lt < (little)
-gt > (great)
-ge >= (great equivalent)
-le <= (little equivalent)
-a && (and)
-o || (or)

Exemplo if numerico, use (()) :

```
#!/bin/bash
clear
echo "Calculo da media"
echo -n "Digite a nota 1: ";read n1 #-n para nao pular
linha
echo -n "Digite a nota 2: ";read n2
media=`echo "($n1 + $n2)/2"|bc` # cuidado com
scale, o if nao aceita
echo Sua media final eh: $media
if (($media >= 5));then
echo "Aprovado"
else
echo "Reprovado"
fi
```

Exemplo if string, use [] :

```
#!/bin/bash
echo "Verificar se um arquivo existe e se tem
permissão escrita"
echo -n "Digite o nome e o caminho de um arquivo:
";read arquivo
if [ -z $arquivo ];then
echo "ATENCAO: Digite um nome para o arquivo"
./teste.sh
else
if [ -f $arquivo ];then
if [ -w $arquivo ];then
echo "O arquivo $arquivo EXISTE e tem
permissao de ESCRITA"
else
echo "O arquivo $arquivo existe mas NAO
tem permissao de escrita"
fi
else
echo "O arquivo nao existe!"
fi
fi
```

Exemplo case:

```
#!/bin/bash
clear
echo -n "Digite s ou n: ";read resposta
case $resposta in
s) echo "Voce digitou s" ;;
n) echo "Voce digitou n" ;;
*) echo "Resposta errada" ;;
esac
```

Comandos de repetição com loop:**Exemplo for;** a variável recebe o valor da lista:

```
#!/bin/bash
echo "Tamanho dos meus scripts"
for file in *.sh
do
du -h $file
done
```

Exemplo for; a variável recebe o valor da lista:

```
#!/bin/bash
echo "RENOMEADOR DE ARQUIVOS"
echo -n "Digite a extensao: ";read extensao
echo -n "Digite um prefixo: ";read prefixo
mkdir renomeados
n=1
for file in *.$extensao
do
mv $file renomeados/$prefixo$n.$extensao
echo "$file ==> $prefixo$n.$extensao ... OK"
n=`expr $n + 1`
done
echo "PRONTO !!"
```

Exemplo for: inicial;teste;incremental

```
#!/bin/bash
for((x=1; x <=10; x++))
do
echo "$x"
done
```

Exemplo break (encerra um comando de repetição)

```
#!/bin/bash
for((x=0;x<=10;x++))
do
if (($x==5));then
break
else
echo $x
fi
done
```

Exemplo while:

```
#!/bin/bash
x=0
while (($x<=10));
do
echo $x
x=`expr $x + 1`
done
```

Exemplo while: Loop infinito facil

```
#!/bin/bash
n=1
while true
do
echo $n
n=`expr $n + 1`
done
```

Exemplo until: (testa da direita pra esquerda)

```
#!/bin/bash
echo "Contador com UNTIL"
x=0
until [ $x -gt 10 ];
do
echo "os valores da lista são: $x"
x=`expr $x + 1`
done
```

shift : desloca parametros de entrada das variaveis especiais \$# e \$* , ex:

```
#!/bin/bash
echo "total de parâmetros iniciais: $#"
```

```
echo "os parâmetros sao: $*"
shift
echo "total de parametros após descolar uma
posição: $#"
```

```
echo "os parâmetros restantes sao: $*"
shift 2
echo "total de parametros após descolar duas
posições: $#"
```

```
echo "os parâmetros restantes sao: $*"

```

Exemplo shift:

```
#!/bin/bash
echo "Contador com entrada de parametro"
if (($#!=1)) || (($*<=0));then
echo "ERRO, Use apenas um numero e maior que zero"
exit
else
for (( x=0;x<=$*;x++ ))
do
echo $x
done
fi
```

Funções: Igual a linguagem C

Exemplo funcao com return:

```
#!/bin/bash
function mostra_numero()
{
echo -n "Digite um número: "
read NUMERO
echo "O número é : $NUMERO"
return
}
mostra_numero
```

Exemplo funcao com variavel local e global:

```
#!/bin/bash
x=10 # variável global
soma()
{
local y=5 # variável local
saida=`expr $x + $y`
echo $saida
}
echo "executando a função: `soma $saida`"
echo "ou execute a função da seguinte forma: "
soma
```

Interface de menus, exemplo:

```
#!/bin/bash
while true
do
clear
echo "-----"
echo " Escolha a opção "
echo "-----"
echo "[1] Exibe a data"
echo "[2] Exibe os arquivos do diretório atual"
echo "[3] Exibe o calendário"
echo "[4] Exibe o editor de texto nano"
echo "[5] Sair"
echo "===== "
echo -n "Escolha uma das opções entre [1-5]"
read OPCAO ; case $OPCAO in
1) echo "Hoje é `date`" ; echo "Pressione uma tecla" ; read ;;
2) echo "Os arquivos em `pwd` são:" ; ls -lha
echo "Pressione uma tecla..." ; read ;;
3) cal ; echo "Pressione uma tecla..." ; read ;;
```

```
4) nano ;;
5) clear; exit 0 ;;
*) echo "Atenção!!! Por favor selecione 1,2,3,4 ou 5" ;
echo "Pressione uma tecla..." ; read ;;
esac
done
```

Caixas de diálogos gráficos:

zenity, opções:

--entry	Exibe uma caixa de texto de entrada
--calendar	Exibe um calendário
--text-info	Exibe uma caixa texto informação
--error	Exibe uma caixa de diálogo de erro
--file-selection	Exibe para selecionar um arquivo
--info	Exibe uma caixa de informação
--list	Exibe uma caixa de listagem
--notification	Exibe um ícone de notificação
--progress	Exibe uma barra de progresso
--question	Exibe uma caixa de pergunta
--warning	Exibe uma caixa de aviso
--scale	Exibe uma caixa de escala

OBS.: Para obter detalhes de cada caixa de diálogo digite: zenity --help-OPÇÃO , ex:
zenity --help-entry, zenity --help-calendar ..

Exemplos de caixas:

```
$ zenity --entry --text="Digite seu nome" --entry-text="Digite aqui"
$ zenity --entry --text="Digite sua senha" --hide-text
$ zenity --calendar --day=27 --month=7 --year=2012
$ zenity --text-info --filename=/etc/passwd --title "PASSWD" --width=400 --height=200
$ zenity --error --text="Mensagem de erro" --no-wrap
$ zenity --file-selection --multiple --directory --separator=,
$ zenity --info --text="Mensagem de informação..." --no-wrap
$ zenity --list --text="Listagem" --checklist --column="Escolha" --column="Produto" TRUE DVD FALSE Livro
$ zenity --list --text="Listagem" --radiolist --column="Escolha" --column="Sexo" FALSE Masculino FALSE Feminino
$ find /etc/ -name *.conf | zenity --list --title "Procurando arquivos" --text "Localizando arquivos .conf" --multiple --column "Arquivos"
$ zenity --notification --text="Mensagem de notificação..."
$ ls -lha |zenity --progress --text="Porcentagem..." --percentage=10 --pulsate
$ zenity --question --text="Deseja prosseguir ?"
$ zenity --text-info --filename=familia.txt --editable
$ zenity --warning --text="Mensagem de aviso"
$ zenity --scale --text="Texto" --value=10 --min-value=0 --max-value=100
```


Exemplos de script com Caixas:

```
#!/bin/bash
NOME=`zenity --entry --text="Digite seu nome:"`
zenity --question --text="Deseja criar o arquivo
$NOME.txt ?"
SEL=$?
case $SEL in
0) touch $NOME.txt ; zenity --info --text="Arquivo
criado $NOME.txt" ; ls -l ;;
1) zenity --warning --text="Operação cancelada" ;;
esac
```

Exemplo de Menu com Caixas:

```
#!/bin/bash
# script com menus gráficos
while true
do
clear
OPCAO=`zenity --list --text="Script com menus"
--width=400 --height=300 --radiolist
--column="Escolha" --column="Escolha a ação"
FALSE "Localizar todos arquivos *.conf em /etc"
FALSE "Abrir Arquivos" FALSE "Exibir Calendário"
FALSE "Abrir navegador Firefox" FALSE SAIR`
case $OPCAO in
"Localizar todos arquivos *.conf em /etc") find /etc
-name *.conf | zenity --progress --pulsate; echo
"Pressione uma tecla" ; read ;;
"Abrir Arquivos") zenity --file-selection --multiple ;
echo "Pressione uma tecla..." ; read ;;
"Exibir Calendário") zenity --calendar ; echo
"Pressione uma tecla..." ; read ;;
"Abrir navegador Firefox") firefox ;;
SAIR) exit 0 ;;
*) echo "Atenção!!! Por favor selecione uma das
opções" ; echo "Pressione uma tecla..." ; read ;;
esac
done
```