

Comandos Linux

Índice

Comandos	1
Globing e Quoting	2
globing e quoting	2
;	2
&&	2
	2
*	2
?	3
[0-9]	3
[a-z] ou [A-Z]	3
{...,...}	3
\	3
“...” ou ‘...’	3
A	4
apropos	4
apt	4
B	5
builtin	5
bash	5
C	6
cat	6
cd	6
cp	6
chmod	7
D	8
dpkg	8
E	9
echo	9
env	9
exit	9
export	9
F	10
find	10
Find e -exec	11
G	13
H	14
hash	14
head	14

help	14
history	14
I	15
J	16
K	17
L	18
less	18
Atalhos e Comandos dentro do less	18
Dicas de Sequência de Comandos	19
login	19
logout	19
ln	20
ls	20
M	22
mkdir	22
more	22
Atalhos e Comandos dentro do more	22
Dicas de Sequência de Comandos	23
mv	23
N	24
O	25
P	26
passwd	26
poweroff	26
pwd	26
Q	27
R	28
rm	28
rmdir	28
rpm	28
S	29
set	29
T	30
tail	30
touch	30
tty	30
U	31
V	32
X	33
W	34
w	34

which	34
Y	35
yum	35
Z	36
Consultar informações usando help, man, info.	37
help	38
man	39
Atalhos de Navegação Dentro do man	39
Onde Encontrar Documentação	40
Onde o man Encontra a Documentação	41

Comandos

Globing e Quoting

globing e quoting

;

; → ; entre dois comandos faz o shell fazer o comando1 depois o comando2.

```
comando1;comando2
```

&&

&& → (E) entre dois comandos faz o shell executar o comando2 se o comando1 ter exit status = 0.

```
comando1 && comando2
```

||

|| → (OU) entre dois comandos faz o shell executar o comando2 se o comando1 dar exit status != 0.

```
comando1 || comando2
```

*

* → antes ou depois de um termo expandirá a busca/seleção para qualquer termo a partir do asteriscos ou antes dele, dependendo da limitação de texto antes ou depois. Exemplos:

```
cat *.txt
```

saída: mostraria todo o conteúdo de cada arquivo .txt de uma pasta.

```
cat text*.txt
```

saída: mostraria todo o conteúdo de todos os arquivos que tem o nome com text(AlgumaCoisa).txt

?

? → antes ou depois de um termo irá expandir a buscar/selecionar o nº de caracteres correspondentes ao nº de ? digitados.

[0-9]

[123] → antes ou depois de um termo irá expandir para buscar/selecionar arquivos onde o espaço que está substituindo sejam os números 1 ou 2 ou 3.

[1239] → antes ou depois de um termo irá expandir para buscar/selecionar arquivos onde o espaço que está substituindo sejam os números 1 ou 2 ou 3 ou 9.

[0-9] → antes ou depois de um termo irá expandir para buscar/selecionar arquivos onde o espaço que está substituindo sejam os números 0 ao 9.

NOTE ! nega o comando. EX: [!0-9] → retornaria coisas que não são nºs.

[a-z] ou [A-Z]

[a-z] → antes ou depois de um termo irá expandir para buscar/selecionar arquivos onde o espaço que está substituindo sejam as letras de “a” a “z” minúsculas. OBS: ! nega o comando. EX: [!a-z] -> retornaria coisas que não são letras

[A-Z] → antes ou depois de um termo irá expandir para buscar/selecionar arquivos onde o espaço que está substituindo sejam as letras de “A” a “Z” maiúsculas.

{...,...}

comando {primeiro*,segundo?} → vai buscar/selecionar resultados correspondentes ao primeiro* e/ou ao segundo? ao mesmo tempo.

\

qualquerCoisa → ao usar o \ antes de qualquer coisa isso indica para o bash que não queremos que essa coisa seja interpretada como normalmente seria.

“...” ou ‘...’

echo ‘parametro’ → ao usa ‘ ou “ no início e final de um parametro indica para o bash que o que está entre ‘ ou “ é um único parametro e não um comando.

A

apropos

apropos palavraBuscada → Busca um comando/manual com determinada palavra-chave.

apt

sudo apt autoremove → remove bibliotecas que não estão sendo utilizadas por outros programas (mesma coisa que **sudo apt-get autoremove**).

sudo apt install nomePacote → instalar pacote (mesma coisa que **sudo apt-get install nomePacote**).

sudo apt search nomePacote → buscar pacote (mesma coisa que **sudo apt-cache search**).

sudo apt remove nomePacote → remove pacote (mesma coisa que **sudo apt-get remove nomePacote**).

sudo apt remove --purge nomePacote → remove pacote e seus arquivos de configurações.

sudo apt remove --purge nomePacote -y → remove pacote e seus arquivos de configurações concordando com alterações.

sudo apt upgrade → atualizar pacotes do sistema (mesma coisa que **sudo apt-get upgrade**)

sudo apt-get autoremove → remove bibliotecas que não estão sendo utilizadas por outros programas.

sudo apt-get dist-upgrade → atualiza somente o kernel do sistema.

apt-cache search nomePacote → buscar pacote.

sudo apt-get install nomePacote → instalar pacote.

sudo apt-get install --only-upgrade nomePacote → atualizar somente o pacote escolhido.

sudo apt-get -f install nomePacote → instala pacote com dependencias, corrige dependencias.

sudo apt-get remove nomePacote → remover pacote.

sudo apt-get update → atualizar lista dos repositórios.

sudo apt-get upgrade → atualizar pacotes do sistema.

B

builtin

builtin comando →

bash

bash → abre um bash. Pode abrir um bash dentro de outro bash. Dica: abra um bash dentro de outro bash para fazer testes.

bash -c →

C

cat

IMPORTANT

Objetivo do cat

O objetivo do cat é servir como concatenador de arquivos.

cat nomeArquivo → exibe conteúdo de nomeArquivo.

cat nomeArquivo1 nomeArquivo2 → exibe e concatena nomeArquivo1 e nomeArquivo2.

cat -n nomeArquivo → exibe conteúdo do nomeArquivo e exibe o número de linhas do documento.

cat -q nomeArquivo → exibe o conteúdo do nomeArquivo sem informações sobre o arquivo. Exibe somente o texto, não exibe o título.

cat -n 5 nomeArquivo → exibe o conteúdo das 5 primeiras linhas. Obs: Também pode ser utilizado qualquer outro número no lugar do 5.

cat /etc/os-release → exibe informações sobre a versão do sistema operacional

cat /etc/lsb-release → exibe informações sobre a versão do sistema operacional debian-like

cat /etc/redhat-release → exibe informações sobre a versão do sistema operacional redhat-like.

cd

cd → muda do diretório atual para a home do usuário.

cd ~ → muda do diretório atual para a home do usuário.

cd ~nomeUsuario → muda do diretório atual para a home no usuário com nome nomeUsuario.

cd ~/pasta → muda do diretório atual para /home/usuario/pasta. É um atalho para usar a /home do usuário.

cd . → muda para o diretório atual.

cd .. → muda para o diretório anterior.

cd endereco/da/pasta/desejada → muda de pasta.

cd - → muda para a pasta anterior em que o usuário acessou no terminal.

cd / → muda para o diretório raiz.

cp

cp nomeArquivo nomeArquivo(copia) → cria uma cópia do nomeArquivo com nome escolhido,

no caso seria, de nomeArquivo(copia)

cp NomeArquivo diretorio1/subdiretorio1 → cria uma cópia de nomeArquivo chamado nomeArquivo em subdiretorio1.

cp -i nomeArquivo nomeArquivo(copia) → antes de copiar arquivos exibe um pergunta para confirmar a operação.

NOTE

Como responder

Após executar o comando **cp -i nomeArquivo nomeArquivo(copia)** será perguntado se deseja sobrescrever o arquivo. Responda **y** para sim ou **n** para não.

cp diretorioOrigem/* diretórioDestino → copia todos os arquivos de diretórioOrigem para diretórioDestino. Não copia diretórios.

cp -u diretorioOrigem/* bkpDiretórioDestino → copia só os arquivos que foram modificados de diretorioOrigem para bkpDiretorioDestino

cp -b diretorio1/* diretorio2 → sobresescreve os arquivos de diretorio1, mas antes cria cópias de backup.

cp -r diretorioOrigem diretorioDestino → copia diretorio. Copia o conteúdo diretorioOrigem para o diretorioDestino recursivamente.

cp -r diretorioOrigem diretorioDestino/ → cria uma pasta diretorioOrigem em diretorioDestino, se existir, e copia todo o conteúdo de diretorioOrigem, incluído a própria pasta, para diretorioDestino.

chmod

chmod+x nomeArquivo → permissão de execução para nomeArquivo. Equivalente a **chmod 777 nomeArquivo**.

D

dpkg

sudo dpkg -i nomeArquivo.deb → instala pacote deb , não instala dependências.

dpkg --get-selections → exibe lista simplificada de pacotes que estão e já estiveram no sistema e seu estado.

dpkg -l → lista pacotes que estão ou já estiveram no sistema (mesma coisa que **dpkg --list**).

dpkg -l | grep nomePacote → encontra o pacote especificado na lista de pacotes que estão ou já estiveram no sistema.

dpkg --list → lista pacotes que estão ou já estiveram no sistema.

sudo dpkg -r nomeArquivo.deb → desinstala pacote deb , não desinstala dependências.

sudo dpkg --remove nomeArquivo.deb → desinstala pacote deb , não desinstala dependências (mesma coisa que **sudo dpkg -r nomeArquivo.deb**).

sudo dpkg -P nomeArquivo.deb → remove pacotes deb e arquivos de configurações.

dpkg --purge nomeArquivo.deb → remove pacotes deb e arquivos de configurações (mesma coisa que **sudo dpkg -P nomeArquivo.deb**).

E

echo

echo mensagem → Exibe uma mensagem na tela.

env

env → lista todas as variáveis de ambiente.

env nomeVariavel1=valorVariavel1 nomeVariavelN=valorVariavelN comando → cria um novo ambiente com as variáveis temporárias e executa o comando ou script. Pode-se criar variáveis com nomes diferentes ou até mesmo iguais a variáveis já existentes, pois o escopo desse comando só vale para o novo ambiente criado. Acabou o comando, acabou o ambiente, fim das variáveis e seus valores.

exit

exit → sair de um shell do usuário. Finaliza o shell.

export

export nomeVariavel → Define atributo de exportação para variáveis shell. Transforma uma variável de shell em de variável de ambiente.

export nomeVariavel -n → variável deixa de ser de ambiente. Variável ainda existe, mas como variável de shell.

F

find

find → exibe arquivos que estão localizados a partir do diretório atual.

find -print → exibe arquivos que estão localizados a partir do diretório atual.

find /caminho/absoluto/diretório/arquivo → exibe arquivos que estão localizados a partir do diretório indicado pelo caminho absoluto.

find /caminho -type f → procurar e exibe todos os arquivos a partir do diretório /caminho

find -type d → procura e exibe todas as pastas a partir do diretório atual.

find . → exibe arquivos que estão localizados a partir do diretório atual. Utiliza caminho absoluto.

find -iname nomeArquivoOuDiretorio → procura e depois exibe todos os resultados que contêm nomeArquivoOuDiretorio, a partir da pasta atual. -iname também aceita ?, * e [], para isso utiliza entre aspas a expressão e suporta arquivos com . (ponto).

TIP | -iname não diferencia maiúsculas e minúsculas.

find -name nomeArquivoOuDiretorio → procura e depois exibe todos os resultados que são exatamente nomeArquivoOuDiretorio, a partir da pasta atual. -name também aceita ?, * e [], para isso utiliza entre aspas a expressão e suporta arquivos com .(ponto).

TIP | -name diferencia maiúsculas e minúsculas.

find -name "arquivo*" → procura e exibe todos os arquivos ou pastas que comecem com arquivo.

TIP | É obrigatório o uso de " " nesses casos.

find -type d -name "nomeDiretorio" → procura e retorna somente diretórios que contenham o termo nomeDiretorio.

find -type f -name "nomeArquivo" → procura e retorna somente arquivos que contenham o termo nomeArquivo.

find -type l -name "nomeLinkSimbolico" → procura e retorna somente links simbólicos que contenham o termo nomeDiretorio.

find /caminho -user nomeUsuario → procura em /caminho os arquivos e pastas que o usuário nomeUsuario é dono.

find /caminho -group nomeGrupo → procura em /caminho os arquivos e pastas que o grupo nomeGrupo é dono.

find -size +10k → procura e retorna arquivos maiores que 10kb.

Opções extras:

TIP -00[k|M|G], arquivos menores que 00 kb ou Mb ou Gb.
 +00[k|M|G], arquivos maiores que 00 kb ou Mb ou Gb.

find -atime -7 → procura arquivos que foram acessados no período dos últimos 7 dias.

TIP **find -atime n**, onde n = nº de dias do período que deseja procurar.

find -mtime -7 → procura arquivos que foram modificados no período dos últimos 7 dias.

TIP **find -mtime n**, onde n = nº de dias do período que deseja procurar.

find -ctime -7 → procura arquivos que foram modificados, mudadas as permissões dos arquivos, incluindo grupos, arquivos mudou de diretório, no período dos últimos 7 dias.

TIP **find -ctime n**, onde n = nº de dias do período que deseja procurar.

find -inum 4474556 → procura arquivos que tem o inode especificado.

find -name "*condicao1" find -name "*condicao2" → procura arquivos que satisfaçam as duas condições de busca. Isso equivale ao um E: **find -name "*condicao1" find -a -name "*condicao2"**.

find -name "*condicao1" -o find -name "*condicao2" → procura arquivos que satisfaçam a condição 1 OU a condição 2.

find \(-name "*condicao1" -o find -name "*condicao2" \) -a -name "*condicao3" → procura arquivos que satisfaçam a (condição 1 OU a condição 2) E depois a condição 3.

Precedência

TIP Ao usar mais de um OU(-o) ou/e E(-a) juntos na mesma expressão utilizar parenteses para ter certeza da precedência do E/OU.

Escape dos ()

TIP Quando usar () utilizar a \ antes, então fica assim: **\(\)**.

find -name "*condicao1*" ! -name "*condicao2*" → procura arquivos que satisfazem a sentença inteira, ou seja, retorna arquivos onde a condicao1 é verdadeira e a condicao2 é falsa. ! é equivalente ao NÃO (negação), ele inverte o valor lógico da sentença.

Find e -exec

IMPORTANT

-exec

-exec → indica que após ele virá um conjunto de instruções para se executar em todos os arquivos resultantes do find.

WARNING

tomar cuidado ao utilizar, pois pode ser usado até o rm como comando no final da busca e todos os arquivos que resultarem da busca do find serão excluídos.

find -name "*condicao1*" -exec ls -l "{}" \; → para cada saída do find será apresentado como ls -l.

NOTE

Uso do {}

{ } indica que para cada caminho que o find encontrar será executado o que está depois de -exec.

NOTE

*Uso do " " e *

"" e \ indica escape do bash, pois { } e ; são caracteres especiais do bash.

find -name "*condicao1*" -exec echo "{}" foi encontrado ";" → exibe “foi encontrado” logo após cada saída do find.

find name "*condicao1*" -exec rm -i "{}" \; → remove interativamente cada arquivo que o find encontrar que satisfazem a condicao1.

G

H

hash

hash → exibe lista de comandos de programas/binários “cacheados” na memória.

hash -r → esquece local de todos os programas já “cacheados”

hash -d nomeComando nomeComando2 → esquece local já “cacheado” do nomeComando e nomeComando2.

head

head nomeArquivo → exibe primeiras linhas do arquivo.

head -n 5 nomeArquivo1 nomeArquivo2 → exibe as 5 primeiras linhas do nomeArquivo1 e do nomeArquivo2.

help

help comando → exibe uma descrição, muitas vezes breve, do que o comando pode fazer junto com instruções de execução do comando.

IMPORTANT

O **help comando** funciona se o comando for um built-in do Shell. Built-ins são os comandos incorporados ao Shell. Para saber se o comando é um built-in use **type comando**.

history

history → lista o histórico dos comandos no terminal.

history #Nº → list os últimos Nº comandos no terminal.

K

L

less

less nomeArquivo -> mostra conteúdo do arquivo

Atalhos e Comandos dentro do less

Comando/Atalho	Propósito
<code>↑</code>	navega para linha de cima
<code>↓</code>	navega para linha de baixo.
<code>E</code> ou <code>J</code>	navega uma linha por vez pra baixo.
<code>Y</code> ou <code>K</code>	navega um linha por vez pra cima.
<code>Espaço</code> ou <code>F</code>	navega por páginas. Próxima página.
<code>B</code>	navega por páginas. Página anterior.
<code>Home</code> ou <code>B</code>	retorna para início do documento.
<code>End</code> ou <code>Shift</code> + <code>G</code>	vai para o fim do documento.
<code>P</code> + <code>0-100</code>	<code>P</code> depois número inteiro entre 0 e 100 pula para a porcentagem correspondente.
<code>H</code>	ajuda.
<code>!</code> comando	com <code>!</code> logo em seguida digitar um comando do terminal, podemos usar comandos do shell dentro do less.
<code>V</code>	abre o editor padrão do sistema na linha atual.
<code>-</code> + <code>I</code>	ignora todos os cases nas buscas
<code>Ctrl</code> + <code>L</code>	redesenha a tela.
<code>R</code>	edesenha a tela.
<code>:</code> + <code>N</code>	próximo arquivo.
<code>:</code> + <code>p</code>	arquivo anterior.
<code>Q</code>	sair.

Comando/Atalho	Propósito						
<code>?</code>	pesquisa reversa (começa pesquisa do final). Comandos de navegação são os mesmos que os da <code>/</code> .						
<code>/</code> <i>Table 1. Atalhos de Navegação da Pesquisa</i> <table> <tr> <th>Atalho</th><th>Propósito</th></tr> <tr> <td><code>N</code></td><td>Próxima incidência</td></tr> <tr> <td><code>Shift + N</code></td><td>Incidência anterior</td></tr> </table>	Atalho	Propósito	<code>N</code>	Próxima incidência	<code>Shift + N</code>	Incidência anterior	pesquisar
Atalho	Propósito						
<code>N</code>	Próxima incidência						
<code>Shift + N</code>	Incidência anterior						
<code>M + letra</code>	Marcar posição						
<code>' + letraMarcada</code>	vai para posição marcada pela letraMarcada						
<code>''</code>	Volta na posição anterior marcada com <code>' + letraMarcada</code>						
<code>& + /</code> TIP <code>& + Enter</code> :Desativa filtros.	Pesquisar (só retorna as linhas onde encontrou a palavra). Pesquisar (só retorna as linhas onde encontrou a palavra).						

Dicas de Sequência de Comandos

- -I & : dica está incompleta.

login

login → Começa uma nova sessão no sistema, abre um novo shell. Será pedido nome do usuário e senha.

logout

logout → finaliza a sessão do usuário, incluindo todos os shells abertos na sessão. Conselhável utilizar o exit.

TIP

Atalho

Pode-se usar também a combinação de teclas `Ctrl + D`, mas não é sempre que vai estar configurada.

ln

ln arquivoOriginal arquivoHardLink → Cria um hardlink com nome ArquivoHardLink do arquivo com nome de arquivoOriginal.

ln -s arquivoOriginal arquivoLinkSimbolico → Cria um link simbólico com nome ArquivoLinkSimbolico do arquivo com nome arquivoOriginal.

ls

ls → Exibe os arquivos do diretório atual em uma lista com colunas.

ls caminho/do/diretório → Exibe os arquivos do diretório indicado por **caminho/do/diretorio** em uma lista com colunas.

ls --color → Saida da listagem das pastas fica colorida.

ls --color=auto → exibe os arquivos do diretório em uma lista com colunas com cores:

Table 2. Siginficado das cores (Não é um padrão)

Cor	Siginificado
Branco	Arquivos
Azul	Pastas
Azul-claro	link
Verde	Executável
Vermelho	Pacotes

ls --format=long → exibe os arquivos do diretório em uma lista de uma única coluna com bastante detalhes sobre os arquivos.

ls --sort=extension → exibe os arquivos do diretório em uma lista ordenada pelo tipo de extensão do arquivo., do que tem extensão para o que não tem em ordem alfabética.

ls --sort=size → Exibe os arquivos do diretório em uma lista ordenada pelo tamanho, do maior para o menor.

ls --sort=time → exibe os arquivos do diretório em uma lista ordenada pelo tempo de criação do arquivo, do mais recente para o mais antigo.

ls * → Exibe todas, menos arquivos ocultos, as pasta e um nível de subpasta de cada pasta do diretório atual.

ls .* → Exibe todos arquivos ocultos e as pasta ocultas e um nível de subpasta de cada pasta do diretório atual.

ls *.extensão → Exibe todos arquivos que terminem com a extensão indicada. Por exemplo **ls *.txt**, retornaria todos os arquivos com a extensão txt.

ls -a → lista pasta e arquivos, incluindo `.`, `..` e ocultas.

ls -color=never → Não exibe cor na listagem de pasta e arquivos.

ls -d → exibe lista somente dos diretórios.

ls -F → lista arquivos e pastas com identificadores visuais (para monitores de uma cor) ao final do nome do arquivo:

Table 3. Significado dos Caracteres

Caracter	Significado
Sem caracter	Arquivo normal
/	Diretório
*	Executável
@	Link

ls -l → mesma função do `ls -lformat=long`.

ls -la → exibe todo os arquivos do diretório, incluindo os ocultos, em uma lista de uma única coluna com bastante detalhes sobre os arquivos.

ls -ld nomeDiretório → exibe informações completas somente do nomeDiretório.

ls -lh → exibe os arquivos do diretório em uma lista de uma única coluna com bastante detalhes sobre os arquivos e com tamanhos dos arquivos instelégiveis para humanos usando potência de 1000, e não de 1024 (por exemplo: 1K, 200M, 1G).

ls -lSr → exibe os arquivos do diretório em uma lista de uma única coluna com bastante detalhes sobre os arquivos em ordem decrescente de tamanho.

ls -p → Lista somente pasta com o identificador visual / no final do nome. Outros tipos de arquivos não tem caracteres de identificação visual.

ls -r → inverter a ordem de exibição dos arquivos. Será de z-a no lugar de a-z.

ls -R → `ls` recursivo. Exibe o conteúdo dos diretórios recursivamente. O `ls` vai listar todas as pastas mães e filhas.

ls -RI → `ls` recursivo. Exibe o conteúdo dos diretórios recursivamente. O `ls` vai listar todas as pastas mães e filhas no formato de uma lista de uma única coluna com bastante detalhes sobre os arquivos.

ls -S → ver `ls --sort=size`.

ls -Sr → Exibe os arquivos do diretório em uma lista com colunas classifiacados por ordem decrescente de tamanho.

ls -t → ver `ls --sort=time`

ls -X → ver `ls --sort=extension`.

M

mkdir

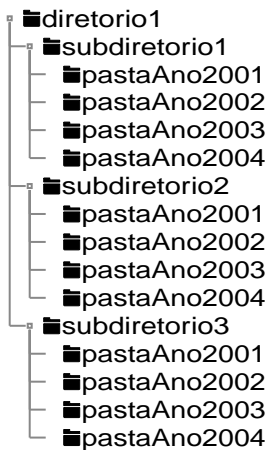
mkdir nomeDiretorio → criar um um diretório na pasta atual

mkdir diretorio1/subdiretorio/nomeDiretorio → cria diretorio na pasta direto1/subdiretorio

mkdir -p diretorio1/subdiretorio1/subdiretorio2 → força a criação de pasta, mesmo se ,por exemplo, subdiretorio1 e subdiretorio2 não existirem.

mkdir nomeDiretorio -v → criar uma pasta e no final exibe uma mensagem que a pasta foi criada.

mkdir -p diretorio1/subdiretorio{1,2,3}/pastaAno200{1,2,3,4} → exemplo de criação de diretórios onde cria da seguinte maneira:



more

more nomeArquivo → mostra conteúdo do arquivo.

Atalhos e Comandos dentro do more

Comando/Atalho	Propósito
Enter	Navega linha por linha
Espaço	navega por páginas. Próxima página.
B	navega por páginas. Página anterior.
H	ajuda.
! comando	com ! logo em seguida digitar um comando do terminal, podemos usar comandos do shell dentro do more.
Q	sair.

Comando/Atalho	Propósito				
<div>/</div> <p><i>Table 4. Atalhos de Navegação da Pesquisa</i></p> <table> <tr> <th>Atalho</th><th>Propósito</th></tr> <tr> <td>N</td><td>Próxima incidência</td></tr> </table>	Atalho	Propósito	N	Próxima incidência	pesquisar
Atalho	Propósito				
N	Próxima incidência				
V	abre editor vi na linha atual do arquivo				

Dicas de Sequência de Comandos

- -I & : dica está incompleta.

mv

mv Nomearquivo diretorio/Destino → move arquivo ou pasta para diretorio/Destino.

mv nomeArquivo novoNomeArquivo → renomeia nomeArquivo (ou pasta) para novoNomeArquivo. Funciona se novoNomeArquivo não existir, se novoNomeArquivo existir irá sobrescrever o novoNomeArquivo.

mv -f nomeArquivo novoNomeArquivo → Caso novoNomeArquivo já exista não irá perguntar se deseja sobrescrever os arquivos.

mv -vi Nomearquivo diretorio/Destino → forma mais segura de renomear um arquivo ou pasta. Irá perguntar se caso sobrescreva o arquivo.

mv -n nomeArquivo novoNomeArquivo → forma segura de renomear arquivo ou pasta. Se arquivo ou pasta existir não sobrescreve.

N

0

P

passwd

passwd → permite que usuário mude a sua própria senha.

passwd nomeUsuario → permite que superusuário, no caso pode ser o usuário root, mudar a senha do usuário especificado.

poweroff

poweroff → desliga o sistema.

pwd

pwd → exibe o caminho atual em que está localizado.

Q

R

rm

rm nomeArquivo → remove/exclui nomeArquivo

rm -fr nomePasta → exclui pasta e subpastas recursivamente e forçadamente sem pergutar.

rmdir

rmdir nomeDiretorio → apaga pasta nomeDiretorio, somente se ela estiver vazia.

rmdir diretorio1/subdiretorio/nomeDiretorio → apaga somente pasta nomeDiretorio, se estiver vazia.

rmdir -p diretorio1/subdiretorio1/subdiretorio2/nomeDiretorio → força o apagamento das que estão vazias.

rmdir -v diretorio1 → remove diretorio1 e exibe uma mensagem relatando a remoção de diretorio1

rpm

rpm -e nomepacote → remove pacote, não remove bibliotecas.

rpm -ihv nomePacote.rpm → instala pacote rpm com a opção de exibi mais detalhes sobre a instalação e com uma barra de carregamento da instalação.

rpm -q nomePacote → consulta se pacote está instalado no sistema.

rpm -qi nomePacote → consulta se pacote está instalado no sistema e exibe mais detalhes sobre pacote.

rpm -qR nomePacote → lista todas as bibliotecas que o pacote/programa está usando, ou seja, lista as dependências do pacote.

S

set

set -o emacs → configura atalhos do terminal para atalhos de emacs, já é o padrão.

set -o vi → configura atalhos do terminal para atalhos de emacs

T

tail

tail nomeArquivo → imprime as 10 últimas linhas de um arquivo.

tail n +30 nomeArquivo → imprime a partir da linha 30 do nomeArquivo. o número de linhas para mostrar pode variar.

tail n 30 nomeArquivo → imprime as últimas 30 linhas do nomeArquivo. o número de linhas para mostrar pode variar.

tail -f nomeArquivo → lê sempre as últimas movimentações do arquivo.

touch

touch nomeArquivo → cria um arquivo vazio. Touch em um arquivo existente atualiza a data de última modificação e acesso do arquivo.

touch nomeDiretorio → Touch em um diretório já existente atualiza a data de última modificação e acesso da pasta.

tty

tty → exibe o nome do terminal em que o usuário está.

U

W

w

w → exibe informações dos usuários atuais que estão na máquina e seus processos.

which

which comando → exibe caminho do executável.

Y

yum

yum check-install nomepacote → verifica se houve erros na instalação do pacote.

yum check-update nomePacote → verifica se existem atualizações para o pacote especificado.

yum clean packages → remove pacotes que estão obsoletos.

yum clean all → remove bibliotecas e pacotes que não estão sendo utilizadas por outros programas.

yum deplist nomePacote → exibe lista de dependências do pacote.

yum remove nomePacote → remove pacote, não remove bibliotecas.

yum search nomePacote → buscar pacote.

yum update → atualiza informações dos pacotes do sistema e depois instala as atualizações disponíveis dos pacotes.

yum upgrade → somente instala as atualizações disponíveis dos pacotes.

yum install nomePacote -y → instalar pacote com confirmação automática.

Z

**Consultar informações usando help,
man, info.**

help

Para usar o **comando + help** ou parecido irá depender do tipo de comando que quer utilizar junto com o help. Cada tipo de comando tem uma sintaxe diferente:

Bultin do shell

- **help + comando**

Programas externos ao shell

- **comando --help** (mais usado)
- **comando -h**
- **comando -?**

man













O man(ual) online está presente na maioria dos comandos.

man comando → exibe um manual com informações do comando.

man man → manual do man.

man -k palavraChave → faz uma busca pela palavraChave em todos os presentes manuais do sistema.

Atalhos de Navegação Dentro do man

Atalho	Propósito						
	Navega para linha de cima.						
	navega para linha de baixo						
 <i>Table 5. Atalhos de Navegação da Pesquisa</i> <table><tr><th>Atalho</th><th>Propósito</th></tr><tr><td></td><td>Próxima incidência</td></tr><tr><td> + </td><td>Incidência anterior</td></tr></table>	Atalho	Propósito		Próxima incidência	 + 	Incidência anterior	pesquisar
Atalho	Propósito						
	Próxima incidência						
 + 	Incidência anterior						

Onde Encontrar Documentação

Para saber o local (diretórios) em que está o(s) manual(ais) de uma comando usar: `whereis comando`

Depois procure arquivos que tenham nomes que fazem referência a documentação de manuais.

Onde o man Encontra a Documentação

ver \$MANPATH

Se a variável \$MANPATH é vazia ele procura um arquivo de configuração e este arquivo está em `less /etc/manpath.config`. No `/etc/manpath.config` o man encontra instruções de como encontrar os manuais dos comandos.

Existe um comando que mostra, de forma resumida, onde o man deve procurar os manuais:

```
~$ manpath  
/usr/local/man:/usr/local/share/man:/usr/share/man
```