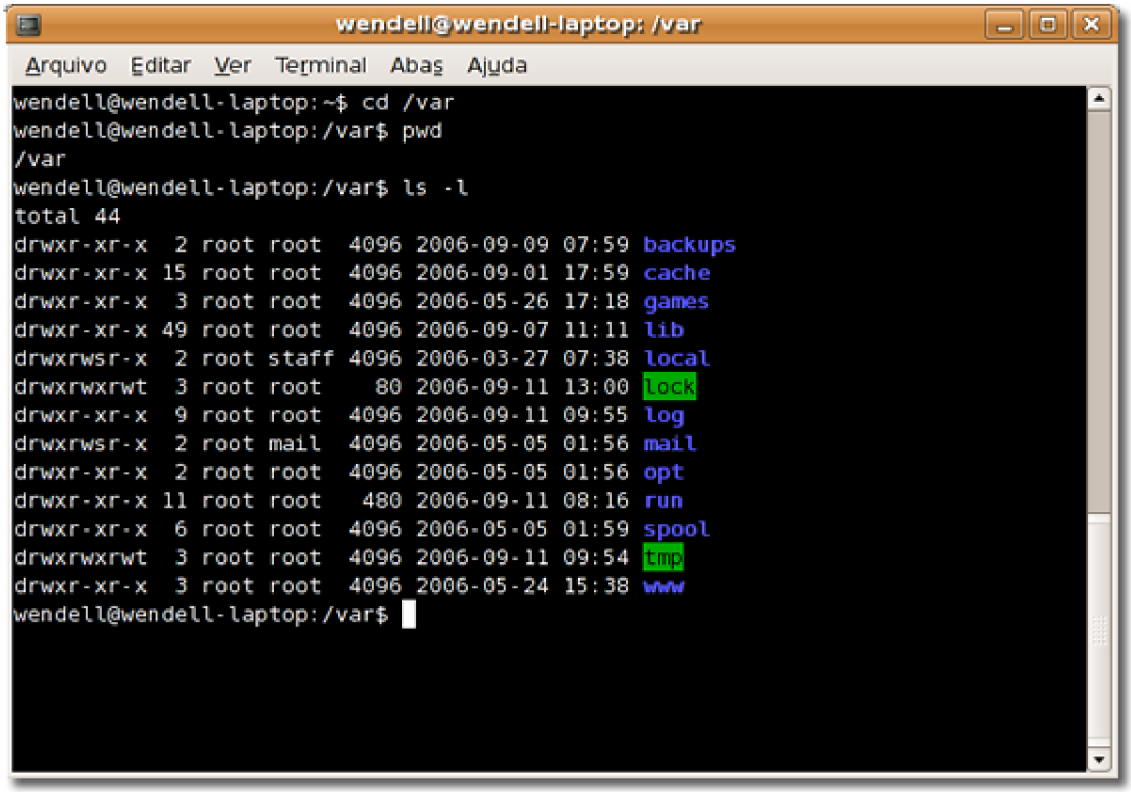
ComandosBasicos

"Muito embora o Linux possua diversas e ótimas interfaces gráfica (GUI's - Graphical User Interfaces) bastante amigáveis, dentre as quais destacamos o Gnome e KDE, como de resto todos os sistemas operacionais Unix, ainda requerem por vezes que façamos uso da linha de comando. O ambiente tradicional do Unix é o CLI (Command Line Interface), onde você digita os comandos para dizer ao computador o que ele deve fazer. Esse modo é extremamente poderoso e rápido, porém implica que você saiba para que serve cada comando e seus diversos parâmetros."

Conteúdo

Conteúdo



Usando essa página

- Essa página ira fazer você se familiarizar com os comandos básicos do GNU/Linux.
- Não é sua intenção ser um guia completo de comandos, somente uma introdução para complementar as ferramentas gráficas do Ubuntu.
- Todos os nomes dos comandos estarão em **negrito**.
- Os comandos que você precisará digitar estarão sempre em **"negrito com aspas"**.
- Todos os comandos nesta página devem ser usados em um terminal ou diretamente no shell.

ATENÇÃO: Lembre-se o Linux diferencia maiúsculas de minúsculas. Portanto, **comando** e **COMANDO** são coisas totalmente diferentes.

Iniciando o Interpretador de Comandos

Abrindo um Terminal

No Gnome vá ao menu Aplicações > Acessórios > Terminal ou pressione simultaneamente as

teclas Alt+F2, e na caixa de texto digite **"gnome-terminal**" e tecle “Enter”.

Abrindo uma seção shell

Tecle simultaneamente Crtl+Alt+F1(...F6) que uma console modo texto será exibido solicitando um login, onde você deverá entrar com seu usuário e senha para ter acesso ao prompt de comando.

Os Comandos do Linux

Como já descrito anteriormente, não temos a pretensão de ser um guia completo de comandos, mais uma fonte de referência que irá abranger os principais comandos separados por categorias de acordo com as tarefas que executam.

Um comando do Linux é uma palavra especial que representa uma ou mais ações. Um interpretador de comandos também é conhecido como shell ou modo texto. Ele é o programa responsável por interpretar essas instruções enviadas pelo usuário e seus programas para o kernel. No Linux, você poderá ter vários interpretadores de comandos (ao contrário do que acontece no Windows que só tem o command.com).

O interpretador de comandos é que executa comandos lidos do teclado ou de um arquivo executável. É a principal ligação entre o usuário. Entre os programas interpretadores de comandos podemos destacar o bash, csh e sh entre outros.

Entre eles o mais usado é o **Bash (Bourne Again Shell)**, criado por S.R. Bourne. Os comandos podem ser enviados de duas maneiras para o interpretador:

- **Interativa** - Os comandos são digitados no teclado pelo usuário e passados ao interpretador de comandos um a um. Neste modo o computador depende do usuário para executar uma tarefa ou o próximo comando.
- **Não-interativa** - São usados arquivos de comandos (scripts) criados pelo usuário para o computador executar os comandos na ordem encontrada no arquivo. Neste modo, o computador executa os comandos do arquivo um por um, e dependendo do término do comando, o script pode verificar qual será próximo comando que será executado e dar continuidade ou não ao processamento.

Esse sistema é muito útil quando temos que digitar por várias vezes seguidas um mesmo comando ou para compilar algum programa complexo.

Uma característica interessante do bash é que ele possui a função de auto-completar os nomes de comandos que foram digitados via entrada padrão. Isso é feito pressionando-se a tecla TAB; o comando é completado e acrescentando um espaço.

Isso funciona sem problemas para comandos internos; caso o comando não seja encontrado, o bash emite um beep. Por exemplo, na sua pasta raiz tente digitar cd pro (aperte TAB)+as(

aperte TAB)+os(aperte TAB)+d(aperte TAB) e veja como foi fácil digitar um caminho para entrar no local: **/proc/asound/oss/devices**.

Outro recurso muito interessante do bash, é que você pode repetir um comando executado sem ter que digitá-lo novamente. Isso é possível utilizando o caractere **"!"** na frente do comando que você deseja repetir. O bash vai buscar aquele comando no histórico e se lá tiver algo parecido o comando será executado. Veja o exemplo abaixo com esta seqüência de comandos:

```
tail -f /var/log/squid/access.log
cd /etc/
ls -hl
!tail
```

O comando **"!tail"** irá informar ao shell (bash) para executar o último comando **tail** executado, no caso, **"tail -f /var/log/squid/access.log"**, e você passara a ver novamente os LOG's do Squid em tempo real.

Para execução de muitos comandos é necessário ter privilégios de administrador, então como no Ubuntu o usuário **root** por questões de segurança se encontra desabilitado, será necessário o uso do **"sudo"**. Assim sendo sempre que um comando necessitar deste privilégio, o mesmo estará precedido do **sudo**.

Adicione também o comando **sudo** na frente de todos os comandos, caso esteja trabalhando em um diretório ou em arquivos que não lhe pertencem (arquivos do sistema, por exemplo). Veja RootSudo para maiores informações sobre o **sudo**.

Documentação

- **man** - Formata e exibe uma página man (man page) O comando **man** é usado para mostrar o manual de outros comandos. Tente **"man man"** para ver a página do manual do próprio **man**. Veja a seção "Man & Getting Help" para mais informações.
- **help** - Exibe informações sobre os comandos internos do Bash. Ex.: **"help logout"**
- **info** - Exibe documentação no formato Info, sendo que a navegação pelo documento é feito por meio de comandos internos do Info. Ex.: **"info emacs"**

Data e Hora

- **date** - Exibe e edita a data e a hora atuais do sistema.
  - **"date"** para exibir a data e hora atual.
  - **"sudo date 032914502007"** para alterar a data e hora para 14:50 h de 29/03/2007.
- **cal** - Exibe um simples calendário.

- **hwclock** - Consulta ou define o relógio do hardware (Hardware Clock).
  - **"sudo hwclock -s"** para atribuir ao sistema a data e hora do hardware (BIOS).
  - **"sudo hwclock --set --date=032914502007"** para definir a data e hora do hardware como 14:50 h de 29/03/2007.

Informações do Sistema (Hardware e Processos)

- **df** – Mostra o espaço em disco do sistema de arquivos usado por todas as partições. "**df -h**" é provavelmente o mais útil - usa megabytes (M) e gigabytes (G) em vez de blocos para relatar o tamanhos. (**-h** significa "human-readable").
- **du** – Exibe o tamanho de arquivos e/ou diretórios. Se nenhum arquivo ou diretório for passado como argumento, será assumido o diretório atual. O uso da opção **du -h** tornará a apresentação mais simples de ser interpretada.

  Para verificar o tamanho dos subdiretórios ao invés dos arquivos, utilize o comando abaixo.

  "du -k -h --max-depth=1"
- **free** – Este comando exibe a quantidade de memória livre e usada no sistema. **"free -m"** fornece a informação usando megabytes, que é provavelmente mais útil para computadores atuais.
- **arch** – Exibe a arquitetura do computador. Equivale ao comando **”uname -m”**.
- **lsdev** – Lista o hardware instalado no computador, especificando os endereços de E/S (Entrada/Saída), IRQ e canais DMA que cada dispositivo esta utilizando.
- **lspci** - Exibe informações sobre os barramentos PCI do computador e sobre os dispositivos a ele conectados.
- **lsusb** - Lista informações sobre os barramentos USB do computador e sobre os dispositivos a eles conectados.
- **uname** - Este comando exibe várias informações sobre o sistema, incluindo o nome da maquina, nome e versão do Kernel e alguns outros detalhes. É muito útil para verificar qual é o Kernel usado por você.
  - **”uname -a”** para exibir todas as informações.
  - **”uname -m”** para exibir a arquitetura da maquina. (Equivale ao **”arch”**).
  - **”uname -r”** para exibir o release do sistema operacional.

- **lsb_release** – Este comando fornece informações básicas do sistema operacional (LSB – Linux Standard Base) e sua distribuição.
    - **"lsb_release -a"** para exibir as informações completas do sistema conforme abaixo exemplificado.

```
user@computer:~$ lsb_release -a
LSB Version:    n/a
Distributor ID: Ubuntu
Description:    Ubuntu (The Edgy Eft Release)
Release:        6.10
Codename:       edgy
```

- **top** - Este comando exibe em tempo real informações sobre seu sistema Linux, processos em andamento e recursos do sistema, incluídos CPU, memória RAM e uso do swap, além do número total de tarefas sendo executadas.
    - O **"top"** também nos permite a manipulação dos processos por meio de comandos interativos. Veja abaixo alguns dos comandos interativos mais importantes do **"top"**.
        - **"k"** - Finaliza, ou seja, "mata" um processo.
        - **"m"** - Ativa/Desativa a exibição de informações da memória.
        - **"M"** - Ordena os processos pelo uso da memória residente.
        - **"N"** - Ordena os processos pelos seus PIDs.
        - **"P"** - Ordena os processos pelo uso da CPU (este é o padrão).
        - **"ESPAÇO"** - Atualiza imediatamente a visualização do quadro de processos.
        - **"h"** - Exibe a ajuda dos comandos interativos do **"top"**.
        - **"q"** - Abandona o comando **"top"**.
- **ps** – Apresenta um quadro atual, porém estático dos processos que estão sendo executados no sistema.
    - **"ps aux"** para apresentar todos processos sendo executados, de todos usuários, incluído o nome do usuário a qual o processo pertence, mesmo os desvinculados de TTYs.
- **kill** – Finaliza, ou no popular, "mata" processos sendo executados pelo seu PID, lhes enviando um sinal.
    - **"kill -9 1345"** para finalizar o processo de PID número 1345. Para saber qual PID

de determinado processo que esta sendo executado pode ser utilizado o comando **ps**.

- **killall** – Finaliza processos pelo nome ao invés do PID como faz o comando **kill**. Também assim como o comando **kill**, o **killall** envia um sinal para o processo.
  - **”killall mozilla-firefox”** para finalizar o processo mozilla-firefox, fechando com isso o navegador web Mozilla Firefox. O nome dos processos ativos pode ser observado com uso do comando **ps**.

Arquivos e Diretórios

- **pwd** - O comando **pwd** lhe permite saber em qual diretório você está no momento, onde **pwd** significa "print working directory".
  - Executando **"pwd"** no diretório Desktop mostrará "~/Desktop". Observe que o Terminal do Gnome também mostra esta informação na barra de títulos da janela. Veja a imagem de exemplo no topo desta página.
- **cd** - Este comando nos permite se deslocar entre a árvore de diretórios do sistema. Quando abrimos um terminal ou seção shell, você entra direto no seu diretório pessoal. Para mover-se pelo sistema de arquivos você deve usar o **cd**.
  - **"cd /"** para ir ao diretório raiz.
  - **"cd"** para ir ao seu diretório pessoal.
  - **"cd .."** para acessar um diretório de nível acima do atual.
  - **”cd -”** para voltar ao diretório que se encontrava antes de mudar.
  - Para navegar através múltiplos níveis de diretórios em só comando, use por exemplo, **"cd /var/www"**, que o levará diretamente ao sub-diretório /www do diretório /var.
- **cp** – Copia arquivos e diretórios.
  - **"cp file foo"** para fazer uma cópia exata do arquivo "file" dando-lhe o nome de "foo".
  - **"sudo cp /etc/X11/xorg.conf /etc/X11/xorg.conf-bkp"** para gerar uma cópia de segurança exata do arquivo "/etc/X11/xorg.conf" dando-lhe o nome de "/etc/X11/xorg.conf-bkp".
- **mv** - Este comando move arquivos e diretórios, sendo muito usado também para renomear um determinado arquivo.

    - **”mv arquivo1 arquivo2”** para renomear o arquivo “arquivo1” localizado no diretório pessoal do usuário para “arquivo2” no mesmo local.
    - **"mv foo ~/Desktop"** moverá o arquivo "foo" para seu diretório Desktop sem alterar seu nome. Você deve especificar um novo nome se quiser renomear um arquivo.

- **ls** - Comando utilizado para listar o conteúdo de um diretório. Usado com certas opções, é possível ver o tamanho dos arquivos, quando foram criados, e as permissões de cada um.
    - **"ls ~"** para mostrar os arquivos que estão em seu diretório pessoal.
    - **”ls -hal ~”** para mostrar os arquivos que estão em seu diretório pessoal, inclusive os ocultos (-a) em forma de uma listagem (-l) e com as informações de tamanho mais amigável a nós seres humanos (-h).
- **rm** - Utilize este comando para remover (deletar) arquivos e opcionalmente diretórios. Por padrão o comando **rm** exibe um prompt onde o usuário deve confirmar a exclusão de cada arquivo, digitando a letra “y” seguido de “Enter”.
    - **”rm arquivo1”** para remover o arquivo chamado “arquivo1” do diretório corrente após confirmação no prompt.
    - **”rm -f arquivo1”** para remover o arquivo chamado “arquivo1” do diretório corrente sem que lhe seja exibido o prompt de confirmação.
    - **”rm -R ~/temp/”** para remover de forma recursiva o diretório /temp localizado em sua pasta pessoal e todo seu conteúdo, seja ele arquivos e outras arvores de sub-diretórios.
- **mkdir** - Comando cuja finalidade é permitir a criação de um ou mais diretórios.
    - **"mkdir musicas"** para criar um diretório chamado “musicas” dentro do diretório corrente.
- **chmod** – Altera as permissões de acesso de arquivos e diretórios, não alterando estes atributos de links simbólicos passados na linha de comando, mais sim as permissões dos arquivos aos quais eles se referem. Para maiores detalhes sobre o sistema de permissões de arquivos e diretórios no Linux aconselhamos este link aqui do Guia Foca GNU/Linux.

Leitura (r) Escrita (w) Execução (x) Octal

0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

```
0  (zero) permissão negada
1   permissão de execução
2   permissão de gravação
3   permissão de gravação e execução
4   permissão de leitura
5   permissão de leitura e execução
6   permissão de leitura e gravação
7   soma de todas as permissões
```

- **"chmod 744 file"** para alterar as permissões do arquivo "file" de modo ao Dono ter total permissão (leitura, execução e escrita) enquanto que os usuários pertencentes ao Grupo e os Outros terão permissão apenas de leitura.
- **"chmod -R 744 temp/"** para alterar as permissões de forma idêntica ao exemplo anterior, porém do sub-diretório /temp e todo seu conteúdo de forma recursiva.
- **chown** – Altera o proprietário e o grupo de arquivos e diretórios.
  - **"chown fulano:vendas file"** para alterar o arquivo "file" para ter como Dono o usuário "fulano" e o Grupo como "vendas".
  - **"chown -R ciclano:compras temp/"** para alterar o sub-diretório /temp e todo seu conteúdo de forma recursiva para ter como Dono o usuário "ciclano" e o Grupo como "compras".

- **diff** – Usado para comparar o conteúdo de dois arquivos, exibindo a diferença entre eles.
    - **"diff file foo"** para ver a diferença entre o conteúdo do arquivo "file" e o arquivo "foo".
- **find** – Comando utilizado para procurar por arquivos na arvore de diretórios. Se um caminho não for passado ao comando **find** a busca será feita no diretório corrente.
    - **"find ~/temp/file"** para procurar pela ocorrência de um arquivo chamado "file" no sub-diretório /temp do diretório pessoal do usuário.
- **locate** – Pesquisa em uma base de dados de nomes de arquivos por nomes que satisfaçam um determinado padrão. O comando **slocate** é a versão segura do **locate**, pois não exibe arquivos para os quais o usuário não tenha permissão de acesso. Como a arvore de arquivos e diretórios esta sempre sendo atualizada é necessário que esta base de dados também o seja, por tanto é sempre aconselhável antes de executar estes comandos atualizar a base executando **"updatedb"**.
    - **"locate ~/file"** para pesquisar por um arquivo que corresponda a expressão "file" no diretório pessoal do usuário. Como este comando pesquisa em um banco de dados, se não for passado ao comando o caminho desejado ele pesquisará em toda sua base de dados, correspondente a toda arvore de diretórios do sistema.
- **tar** Usado para armazenar ou extrair arquivos TAR (Tape ARchive). Estes arquivos TAR são os chamados "tarfile" ou "tarball".
    - **"tar cvf my_ogg_files.tar *.ogg"** para criar um arquivo TAR chamado "my_ogg_files.tar" contendo todos os arquivos de extensão ".ogg" do diretório corrente. Notar que a extensão ".tar" não é obrigatória, mais aconselhável para facilitar a identificação do arquivo.
    - **"tar tvf my_ogg_files.tar"** para exibir todo o conteúdo do arquivo TAR chamado "my_ogg_files.tar".
    - **"tar xvf my_ogg_files.tar"** para extrair todo conteúdo do arquivo "my_ogg_files.tar" no diretório corrente.
    - **"tar xvf my_ogg_files.tar musica1.ogg"** para extrair apenas o arquivo chamado "musica1.ogg" do tarball "my_ogg_files.tar" no diretório corrente.
    - **NOTA**: Arquivos que possuem a extensão **.tar.gz** podem ser descompactados e extraídos com as opções **xzvf** do comando **tar**. Isto corresponde a usar o comando **gunzip** para descompactar o arquivo TAR e depois usar o comando **tar xvf** para extrair os arquivos.

- **gzip** Compacta e opcionalmente descompacta arquivos regulares. Os arquivos compactados com o comando são substituídos por outro de menor tamanho com a extensão **.gz** porém preservando o dono, as permissões e datas de acesso e modificação.
    - **"gzip arq1 arq2"** para compactar os arquivos "arq1" e "arq2" gerando os arquivos "arq1.gz" e "arq2.gz" em substituição aos originais.
    - **"gzip -d arq1"** para descompactar o arquivo "arq1.gz" trazendo de volta o arquivo original "arq1". A presença da opção **-d** equivale ao uso do comando **gunzip**.
- **bzip2** Compacta e opcionalmente descompacta arquivos regulares. Assim como o **gzip**, os arquivos compactados com este comando são substituídos por outro de menor tamanho com a extensão **.bz2** porém preservando o dono, as permissões e datas de acesso e modificação. O algoritmo empregado por este comando permite uma maior compressão e também segurança dos arquivos gerados, porém o processo se torna um tanto quanto mais demorado.
    - **"bzip2 arq1"** para compactar o arquivo "arq1" gerando em substituição o arquivo "arq1.bz2".
    - **"bzip2 -9 arq2"** para compactar o arquivo "arq2" pelo processo de máxima compressão gerando em substituição o arquivo "arq2.bz2".
    - **"bzip2 -d arquivo.bz2"** para descompactar o arquivo "arquivo.bz2" trazendo de volta o(s) arquivo(s) original(is) que tinham sido previamente compactados.

Sistema de Arquivos

- **mount** – Monta um sistema de arquivos tornando-o disponível para as operações de E/S (Entrada/Saída) em arquivos, ou exibe uma lista dos sistemas de arquivos atualmente montados.
    - **"mount"** para listar os sistemas de arquivos atualmente montados.
    - **"sudo mount -t ext3 /dev/hda3 /media/hda3"** para montar a terceira partição primária do disco hda (IDE1) formatado em EXT3 no diretório /media/hda3. É necessário que o diretório /media/hda3 tenha sido previamente criado para que o comando tenha sucesso.
- **umount** – Desmonta um sistema de arquivos previamente montado que não esteja em uso.
    - **"sudo umount /dev/hda3"** para desmontar o dispositivo /dev/hda3. Para que o comando seja executado com sucesso é importante que o dispositivo não esteja em uso, como por exemplo com arquivos abertos ou mesmo estando dentro do

diretório onde o mesmo se encontra montado.

- **fdisk** – Gerencia por meio de uma simples interface de texto orientada por menus as partições de um disco. Ao executar o comando **fdisk *dispositivo*** basta pressionar a tecla **m** no prompt para ter acesso ao menu de opções que é bastante auto-explicativo, devendo se usar as setas de direção para movimentar-se pelo mesmo.
  - **"sudo fdisk -l"** para listar as tabelas de partições para todos dispositivos.
  - **"sudo fdisk /dev/hda"** para gerenciar a partição (ou partições) do dispositivo /dev/hda.
- **fsck** – Verifica e opcionalmente repara um ou mais sistemas de arquivos. O **fsck** na realidade é apenas uma espécie de *front-end* de comandos específicos de acordo com o sistema de arquivos, que na realidade obedecem em geral ao formato **fsck.*nome_do_sistema_de_arquivos***.
- **"sudo fsck -t ext3 /dev/hda3"** para verificar o sistema de arquivos EXT3 do dispositivo /dev/hda3. O mesmo resultado poderia ser alcançado executando o comando da seguinte forma **"fsck.ext3 /dev/hda3"**. O dispositivo deve obrigatoriamente estar desmontado para execução desta operação.
- **mkfs** – Formata um dispositivo (geralmente uma partição de disco) criando um novo sistema de arquivos. O **mkfs**, assim como o **fsck** é apenas uma espécie de *front-end* de comandos específicos de acordo com o sistema de arquivos, que na realidade obedecem em geral ao formato **mkfs.*nome_do_sistema_de_arquivos***.
  - **"sudo mkfs -t ext3 /dev/hda3"** para formatar o dispositivo /dev/hda3 em um sistema de arquivos EXT3. O mesmo resultado poderia ser alcançado executando o comando da seguinte forma **"mkfs.ext3 /dev/hda3"**. O dispositivo deve obrigatoriamente estar desmontado para execução desta operação.
- **badblocks** – Procura por blocos ruins em um dispositivo, geralmente uma partição de disco.
  - **"sudo badblocks /dev/hda3"** para verificar se o dispositivo /dev/hda3 se encontra com blocos ruins. Normalmente, dependendo do tipo e tamanho do dispositivo este procedimento é um tanto demorado, sendo que se nenhuma informação for retornada é porque blocos ruins não foram encontrados. Uma melhor alternativa ao comando seria **"sudo badblocks -o /tmp/file -n /dev/hda3"**, onde o parâmetro **-n** forçaria um teste de leitura e escrita não-destrutivo e o **-o /tmp/file** geraria o arquivo /tmp/file com todas mensagens de saída do comando.

Usuários e Grupos

- **useradd** - Cria um novo usuário ou atualiza as informações padrão de um usuário no sistema Linux. O comando **useradd** cria uma entrada para o usuário no arquivo "/etc/passwd" com informações do seu login, UID (user identification), GID (group identification), shell e diretório pessoal, e a senha criptografada deste usuário é armazenada no arquivo "/etc/shadow".
    - **"sudo useradd fulano"** para criar o novo usuário "fulano" no sistema, cujo diretório pessoal do mesmo será "/home/fulano".
    - **"sudo useradd -d /home/outro_dir fulano"** para criar o novo usuário "fulano" no sistema, porém com seu diretório pessoal se localizando em "/home /outro_dir".
    - **"sudo useradd -s /bin/sh fulano"** para criar o usuário "fulano" definindo seu shell como sendo o sh. O shell padrão do Ubuntu, assim como a maioria das outras distribuições é o bash. Com esta opção "-s" é possível criar um usuário sem que o mesmo possa ter acesso a nenhum shell do sistema, bastando executar o seguinte comando **"useradd -s /bin/false fulano"**.
    - **"sudo adduser -g 600 -G 500,68 fulano"** para criar o usuário "fulano" com grupo padrão de GID 600 e também pertencente aos grupos GID 500 e GID 68. Para saber os GID de cada grupo do sistema consulte o arquivo "/etc/group".
    - **NOTA**: Com a mesma finalidade porém com mais opções informativas sobre o usuário a ser cadastrado existe o comando **adduser**. A configuração padrão usada pelos comandos **useradd** e **adduser** é definida em "/etc/default/useradd" e em "/etc/login.defs".
- **userdel** – Usado para remover uma conta de usuário do sistema, deletando todas entradas deste usuário nos arquivos /etc/passwd, /etc/shadow e /etc/group.
    - **"sudo userdel -r fulano"** para remover o usuário "fulano" do sistema deletando seu diretório pessoal e todo seu conteúdo.
- **usermod** – Altera as informações de um usuário, editando diretamente as informações dos arquivos /etc/passwd, /etc/shadow e /etc/group.
    - **"sudo usermod -d /home/novo_dir fulano"** para criar um novo diretório pessoal para o usuário "fulano" em "/home/novo_dir". Se quiser que o atual diretório do usuário seja movido para o novo diretório utilize a opção "-m" desta forma **"sudo usermod -d /home/novo_dir -m fulano"**.
    - **"sudo usermod -g 800 fulano"** para alterar o grupo padrão do usuário "fulano" para GID 800.
    - **"sudo usermod -s /bin/false fulano"** para alterar o shell do usuário "fulano" para

"/bin/false" não mais permitindo que o usuário faça login no sistema.

    - **"sudo usermod -e 03/04/2007 fulano"** para alterar a data de expiração da conta do usuário "fulano" para 03/04/2007.

- **"finger"** - Exibe informações dos usuários do sistema. Se um usuário não for passado ao comando o mesmo apresentará informações de todos usuários atualmente logados.

    - **"finger fulano"** para exibir informações, como login, diretório pessoal, shell entre outras do usuário "fulano".

- **passwd** – Altera a senha de um usuário exibindo um prompt para que a nova senha seja fornecida, e logo depois repetida para confirmação. O usuário logado pode alterar a própria senha digitando apenas **"passwd"**.

    - **"sudo passwd fulano"** para alterar a senha do usuário "fulano".

    - **"sudo passwd -l fulano"** para bloquear a conta do usuário "fulano".

    - **"sudo passwd -u fulano"** para desbloquear a conta do usuário "fulano".

    - **"sudo passwd -d fulano"** para desativar a senha do usuário "fulano" deixando-o sem uma senha de acesso.

- **groupadd** – Cria um novo grupo no sistema. Deve-se remover os usuários do grupo, antes de apagar o grupo, pois o Linux não faz nenhum tipo de verificação neste sentido.

    - **"sudo groupadd novogrupo"** para criar um novo grupo no sistema chamado "novogrupo".

    - **"sudo groupadd -g 800 novogrupo"** para atribuir ao grupo "novogrupo" o GID 800.

- **groupdel** – Exclui um grupo no sistema.

    - **"sudo groupdel novogrupo"** para excluir o grupo chamado "novogrupo".

- **groupmod** – Altera as informações de um grupo do sistema.

    - **"sudo groupmod -n velho_grupo novo_grupo"** para alterar o nome do grupo "velho_grupo" para "novo_grupo".

    - **"sudo groupmod -g 900 novo_grupo"** para alterar o identificador do grupo chamado "novo_grupo" para GID 900.

- **id** – Exibe os identificadores (IDs) reais e efetivos de usuário e de grupo de um usuário. Se não for especificado ao comando um usuário será exibido as informações do usuário atual.

- **"id fulano"** para exibir os IDs de usuário e grupo do usuário "fulano".

Utilitários de Texto

- **cat** – Utilizado para concatenar arquivos exibindo o resultado na tela, sendo também utilizado para exibir o conteúdo de arquivos.
  - **"cat arq"** para exibir o conteúdo do arquivo chamado "arq". Se desejar que as linhas do arquivo sejam enumeradas use a opção "-n" junto ao comando, desta forma **"cat -n arq"**.
  - **"sudo cat /etc/passwd /etc/group"** para exibir na tela o conteúdo dos arquivos "/etc/passwd" e "/etc/group".
  - **"cat file1 file2 |less"** para exibir na tela o conteúdo dos arquivos "file1" e "file2" porém fazendo a paginação das telas. Neste caso a opção "|less", onde "|" é o chamado pipe, pode ser substituída também por "|more", sendo que ambos comandos serão vistos posteriormente.
  - **"cat arq arq1 arq2 > arq_final"** para concatenar os arquivos "arq", "arq1" e "arq2" e colocar o resultado em outro arquivo chamado "arq_final". Notar que neste comando é feito uso do caractere ">" chamado de redirecionador de saída.
  - **"cat arq3 >> arq_final"** para inserir o conteúdo do arquivo "arq3" ao final do arquivo "arq_final".
  - **NOTA**: O comando **cat** também pode ser usado para criar arquivos quando usado em conjunto com o ">" redirecionador de saída. Para criar um arquivo execute o comando **"cat > novo_arq"** e digite o conteúdo desejado, usando a tecla "Enter" como separador de linhas e "Ctrl+D" para finalizar.
- **less** – Faz a paginação de saídas muito extensas exibindo uma tela por vez.
  - **"less arq"** para exibir o conteúdo do arquivo "arq" de forma paginada. Para navegação e gerenciamento do comando use as teclas abaixo:
    - Para sair do aplicativo digite **q** (quit);
    - Use as teclas **Page-Down**, **Ctrl+F** ou **Space** para avançar nas páginas;
    - Use as teclas **Page-Up** ou **Ctrl+B** para voltar as páginas;
    - Use **Enter** para avançar apenas uma linha por vez;
    - Digite **h** para ver a lista das teclas disponíveis para navegação no comando.
  - **NOTA**: Para redirecionar a saída de outro comando para o **less** efetuar a

paginação, use o "|" (pipe) conforme exemplo **"ls -hl |less"**.

- **more** – Semelhante ao comando **less** também faz a paginação de uma saída muito grande na tela. A sintaxe deste comando é semelhante ao do **less**, inclusive as teclas de navegação e o redirecionamento com uso do "|" (pipe).
- **grep** – Usado para procurar por linhas em um arquivo que contenham expressões que satisfaçam um determinado padrão de busca.
  - **"grep termo arq"** para procurar por entradas no arquivo "arq" que correspondam a expressão "termo".
  - **"grep 'termo1 termo2' arq"** para procurar por entradas no arquivo "arq" que correspondam as expressões "termo1" e "termo2". Notar que quando a expressão é composta de mais de uma palavra deve ser usado aspas simples.
  - **NOTA**: Este comando comumente é utilizado em conjunto com outros comandos canalizados com o "|" (pipe) conforme abaixo exemplificado.
    - **"sudo cat /etc/passwd |grep fulano"** para procurar por uma entrada que corresponda a expressão "fulano" no arquivo "/etc/passwd".
- **tail** – Exibe as últimas linhas da saída de um arquivo. Por padrão se nenhum parâmetro diferente for passado ao comando será exibido as últimas 10 linhas do arquivo.
  - **"tail -50 arq"** para exibir as últimas 50 linhas do arquivo chamado "arq".
  - **"sudo tail -f /var/log/messages "** para continuar exibindo indefinidamente as últimas 10 linhas (padrão) do arquivo "/var/log/messages ". Conforme o exemplo, esta opção "-f" é muito usada para verificar arquivos de log do sistema que estão sendo constantemente atualizados.
  - **NOTA**: Assim como o **tail** que exibe as últimas linhas de um arquivo, existe o comando **head** que faz exibir as primeiras linhas de saída de um arquivo.

Monitoramento de Acesso

- **w** – Mostra quem esta logado no sistema e o que esta fazendo. Se não for especificado um usuário ao comando, será exibido informações de todos usuários logados.
  - **"w"** para exibir todos usuários logados e o que estão executando neste momento.
  - **"w fulano"** para mostrar informações do usuário "fulano" se o mesmo estiver logado no sistema.
- **who** – Semelhante ao comando **w** mostra quais usuários estão logados no sistema.

    - **"who -m"** para mostrar o nome do usuário logado no sistema.
    - **"who -q"** para mostrar a quantidade total e nomes dos usuário conectados ao sistema.
- **whoami** - Este comando fornece o mesmo resultado do comando **"who -m"**.
- **last** – Mostra todas informações referente as entradas (login) e saídas (logout) de usuários do sistema.
    - **"last -a"** para exibir estas informações mostrando o nome da maquina de onde foi efetuado os logins.
    - **"last -d"** para exibir estas informações mostrando o endereço IP da maquina de onde foi efetuado os logins.
    - **"last reboot"** para exibir um registro de todas as reinicializações efetuadas no sistema.
- **lastlog** – Exibe informações referente ao último login de cada usuário cadastrado no sistema. Caso nenhum argumento seja passado, o comando **lastlog** exibe todas as informações armazenadas no arquivo "/var/log/lastlog" de todos os usuários do sistema.
    - **"sudo lastlog -u fulano"** para exibir informações referentes apenas ao último login do usuário "fulano.
    - **"sudo lastlog -t 5"** para exibir a lista dos usuários que logaram no sistema nos últimos 5 dias informando o dia e a hora do último acesso de cada um desses usuários.

Rede

- **ifconfig** – Permite configurar as interfaces de rede, sendo o comando utilizado na inicialização do sistema para configuração destas interfaces. Caso nenhum argumento seja passado junto ao comando, o mesmo apenas irá exibir o estado das interfaces atualmente definidas.
    - **"sudo ifconfig eth0"** para exibir o estado e informações da interface de rede eth0.
    - **"sudo ifconfig eth1 down"** para desativar a interface de rede eth1.
    - **"sudo ifconfig eth1 up"** para ativar a interface de rede eth1.
    - **"sudo ifconfig eth0 192.168.3.1 netmask 255.255.255.0 up"** para configurar a interface de rede eth0 com endereço IP 192.168.3.1 e máscara da rede 255.255.255.0, ativando-a.

    - **"sudo ifconfig eth1 hw ether 00:D0:D0:67:2C:05"** para alterar o endereço MAC (MAC Address) da interface de rede eth1 para " 00:D0:D0:67:2C:05". É necessário que a placa de rede esteja desativada **"sudo ifconfig eth1 down"** para esta operação.

    - **"sudo ifconfig eth0:1 10.0.0.2 netmask 255.255.255.0 up"** para adicionar um segundo endereço de rede, com IP 10.0.0.2 e máscara 255.255.255.0 a interface eth0.

- **arp** – Manipula o cache ARP (Address Resolution Protocol) do kernel.

    - **"sudo arp 192.168.3.1"** para exibir as entradas para o host 192.168.3.1. Se um host não for especificado, será exibido todas as entradas do cache.

    - **NOTA**: Esta ferramenta é muito útil quando se faz necessário descobrir o endereço MAC de um determinado host da rede.

- **ping** Envia requisições ICMP para um determinado host. É uma ferramenta largamente utilizada para testar a conectividade entre uma maquina/rede local e maquinas/redes remotas.

    - **"ping -c 5 200.106.28.125"** para verificar se a maquina cujo endereço IP é 200.106.28.125 se encontra conectada e alcançável. É importante ressaltar que muitos servidores, principalmente de redes empresariais, podem bloquear requisições de pacotes ICMP em seu firewall, podendo assim parecer que determinada rede não se encontra alcançável.

- **route** – Permite exibir a tabela de roteamento (configuração das rotas) IP do kernel, sendo que com uso das opções **add** e **del** permite também modificar esta tabela inserindo ou deletando registros.

    - **"sudo route"** para exibir a tabela das rotas atualmente ativas.

    - **"sudo route add -net 192.120.10.0 netmask 255.255.255.0 dev eth0"** para adicionar uma rota para rede 192.120.10.0 via interface de rede eth0.

    - **"sudo route del -net 192.120.10.0 netmask 255.255.255.0 dev eth0"** para remover a rota anteriormente adicionada.

Módulos carregáveis do Kernel

- **lsmod** Lista todos módulos do kernel atualmente carregados na memória. Na realidade, o comando **lsmod** apenas lista o conteúdo do arquivo "/proc/modules".

- **modinfo** – Exibe informações sobre um determinado módulo carregado do kernel.

    - **"sudo modinfo ip_tables"** para exibir informações do módulo "ip_tables" que se encontra carregado na memória do sistema.

- **modprobe** – Usado para gerenciar, ou seja, adicionar e remover módulos carregáveis do kernel. O **modprobe** lê o arquivo de dependências de módulos gerado pelo **depmod**, portanto devemos sempre antes executar o comando **"sudo depmod -a"**.

    - **"sudo modprobe iptable_nat"** para carregar na memória o módulo "iptable_nat".

    - **"sudo modprobe -r ndiswrapper"** para remover da memória o módulo "ndiswrapper".

Shell (Bash) e Utilitários de Terminal

- **alias** Tem como finalidade atribuir um "alias" (em inglês, significa outro nome) a outro comando, permitindo nomear um conjunto de comandos, a ser executado pelo sistema por um único nome. Caso nenhum parâmetro seja passado ao comando será listado todos alias atualmente definidos e ativos no sistema.

    - **"alias ls='ls -hal --color'"** para definir uma alias **ls** para o comando **ls -hal** que irá mostrar os arquivos que estão no diretório correntel, inclusive os ocultos (-a) em forma de uma listagem (-l) e com as informações de tamanho mais amigável a nós seres humanos (-h) e diferenciado por cores.

    - **"alias fd='mount /dev/fd0 /mnt/floppy; cd /mnt/floppy && ls'"** para criar um alias chamado **fd** que montará um disquete, acessando e listando seu conteúdo. Observe que, neste exemplo, foram usados dois diferentes separadores de comandos: **ponto-e-vírgula** e **&&**. Comandos separados por **;** são executados em seqüência. Comandos separados por **&&** são executados de forma condicional, ou seja, o comando após o separador só é executado se o comando anterior tiver sido executado com sucesso.

    - **"alias mcdrom='mount /mnt/cdrom'"** para criar um alias chamado **mcdrom** que ao ser executado monta o CD em uso.

    - **NOTA**: Estes aliases são criados apenas para a sessão ativa do usuário, ou seja, ao deslogar do sistema os mesmos se perderão. Para criar aliases permanentes ao sistema edite o arquivo **.bashrc** de seu diretório pessoal e inclua no mesmo os comando desejados. Em contrapartida ao comando **alias** existe o comando **unalias** que faz justamente o inverso, removendo os alias criados.

- **apropos** Pesquisa por um padrão na base de dados do comando **whatis** que veremos logo abaixo, informando quais comandos do Linux correspondem a uma determinada expressão.

    - **"apropos apropos"** (1) - search the whatis database for strings (Procura por expressões na base de dados whatis), ou seja exibe todos comandos Linux que tenham alguma correspondência a expressão "apropos", no caso apenas o comando **apropos**.

- **login** Permite a um usuário efetuar o logon (estabelecer uma conexão) no sistema, bem como ser utilizado para efetuar o logon com um usuário diferente do atual.
    - **"login fulano"** para efetuar o login do usuário "fulano".
    - **"login -p fulano"** para efetuar o login do usuário "fulano" sem destruir o ambiente do atual usuário.
- **logout** Finaliza um login shell no console ou terminal. No modo gráfico, este comando encerra a sessão do usuário podendo fechar a janela do terminal, e em modo texto encerra a sessão do usuário levando-o de volta ao prompt de login do sistema.
    - **"logout"** O mesmo resultado pode ser alcançado executando o comando **"exit"**.
- **su** Permite alternar entre os usuários cadastrados do sistema, alterando o ID de usuário e grupo do atual usuário para outro usuário especificado.
    - **"su fulano"** permite alternar para o usuário "fulano" após senha de login correta.
    - **"su fulano -c 'vim /home/fulano/arq1'"** permite executar o comando vim abrindo o arquivo "/home/fulano/arq1" como sendo o usuário "fulano". O uso desta opção **-c** não começa um novo shell, apenas executa um comando como sendo o outro usuário especificado.
- **sudo** Permite a um usuário autorizado conforme configurado no arquivo "/etc/sudoers", a executar comandos como se fosse o super-usuário (root) ou outro usuário qualquer. Veja RootSudo para maiores detalhes.
- **uname** Exibe várias informações sobre o sistema. Caso nenhuma opção seja fornecida junto ao comando, apenas o nome do sistema operacional será exibido, equivalente a opção **-s**.
    - **"uname -a"** para exibir todas informações sobre o sistema.
- **whatis** Pesquisa em uma base de dados que contem uma curta descrição dos comandos do sistema. Esta base de dados com os comandos do sistema é criada e atualizada com o comando **"sudo makewhatis"**
    - **"whatis sudo halt"** para obter uma descrição resumida dos comandos **sudo** e **halt**.
- **whereis** Usado para localizar o binário, o arquivos-fonte e a página **man** (manual) dos

comandos do sistema.

  - **”whereis ls”** para descobrir onde se encontra o arquivo binário, os fontes e o manual (**man**) do comando **ls**.

- **which** Exibe o caminho completo na hierarquia de diretórios para os comandos do sistema.

  - **”which firefox”** para exibir o diretório onde se encontra o programa “firefox”.

- **clear** Limpa a tela movendo o cursor para primeira linha. Não existem parâmetros passados junto a este comando.

- **echo** Permite exibir textos na tela. Este comando também exibe toda estrutura de diretórios e arquivos em ordem alfabética, porém sem formatar em colunas a listagem.

  - **”echo 'Olá mundo!'”** envia para saída de tela a expressão “Olá mundo!”.

  - **”echo /etc/*”** para listar todo conteúdo do diretório “/etc”.

- **halt, reboot, shutdown** Respectivamente encerra, reinicializa e encerra ou reinicializa o sistema.

  - **”sudo halt”** para encerrar o sistema.

  - **”sudo reboot”** para reiniciar imediatamente o sistema. Este comando equivale aos comandos **”sudo init 6”** e **”sudo shutdown -r now”**.

  - **”sudo shutdown -h now”** para encerra o sistema imediatamente.

  - **”sudo shutdown -h +15”** para encerrar o sistema daqui a 15 minutos.

  - **”sudo shutdown -r 20:30 'O sistema será reiniciado as 20:30 horas!'”** para reiniciar o sistema as 20:30 horas enviando a mensagem "O sistema será reiniciado as 20:30 horas!" a todos usuários logados.

  - **NOTA**: O comando **”sudo init 0”** também pode ser usado para encerramento do sistema. O comando **shutdown** é a forma mais segura de reiniciar e finalizar o sistema, advertindo os usuários logados e bloqueando novos logons.

Opções

O comportamento padrão para um comando pode ser modificado por adicionar uma ***--opção*** para o comando. O comando **ls**, por exemplo, tem uma opção **-s** , de forma que **''ls -s''** incluirá o tamanho dos arquivos na listagem realizada. Há também uma opção **-h** para que esses dados estejam em um formato "legível para humanos".

As opções podem ser agrupadas, sendo possível, por exemplo usar **"ls -sh"**, que funcionará exatamente da mesma forma que **"ls -s -h"**. Muitas opções têm uma versão longa, prefixadas por dois traços em vez de um, assim **"ls --size --human-readable"** é o mesmo comando dado anteriormente.

Dicas e Truques

Teclas de controle e atalhos

Teclas	Ação
Ctrl + f	Move o cursor uma palavra para frente
Ctrl + b	Move o cursor uma palavra para trás
Ctrl + a	Para ir ao início da linha de comando
Ctrl + e	Para ir ao final da linha de comando
Ctrl + t	Inverte o caractere sob o cursor com o anterior
Ctrl + u	Limpa a linha de comando corrente
Ctrl + y	Re-insere o último trecho de comando apagado
Ctrl + r	Faz uma busca incremental no histórico de comandos utilizados
Ctrl + c	Termina a execução do comando corrente
Ctrl + d	Encerra entrada de dados pelo teclado fazendo **logout**
Ctrl + m	Equivalente a tecla **Enter**
Ctrl + l	Limpa a tela, equivalente ao comando **clear**

Ctrl + s	Inibe a exibição de informações na tela de saída
Ctrl + q	Ativa a exibição de informações na tela de saída, inibida pelo Ctrl + s
Ctrl + z	Põe o processo corrente em background (segundo plano)

NOTA: Para maiores detalhes, veja aqui nossa página exclusiva sobre atalhos de teclado no bash.

Teclas de emergência do GNU/Linux

Quem é que já não se deparou com um travamento causado por mal-funcionamento de hardware no Linux? Este tópico ensina a usar as teclas de emergência do kernel.

NOTA: As teclas de emergência do kernel são comandos de baixo nível pouco conhecidos que podem desempenhar uma função primordial na vida de usuários Linux.

Desligando o computador

A primeira combinação de emergência é usada para sincronizar os discos e desligar o computador instantaneamente evitando problemas nos sistemas de arquivos. Ela é ideal para quem precisa desligar o computador rapidamente sem danificar seus sistemas de arquivos, ou quando a máquina trava e por qualquer motivo não permite um desligamento natural através do init.

```
Mantendo ALT pressionado, tecle Print Screen e depois O.
```

Reiniciando o computador

Assim como o Ctrl+Alt+Del do MS-DOS o kernel do Linux também possui uma chamada de emergência que permite reiniciar a máquina, com a vantagem de sincronizar os discos evitando danos no sistema de arquivos. Veja como fazer:

```
Mantendo ALT pressionado, tecle Print Screen e depois B.
```

Sincronizando os discos

Se você acha que a força vai cair e precisa trabalhar até a ultima hora mas tem medo de danificar seu sistema de arquivo, poderá sincronizar seus discos de tempos em tempos.

Para sincronizar discos em caso de emergência:

```
Mantendo ALT pressionado, tecle Print Screen e depois S.
```

Segurança

Se por algum motivo algo está ameaçando a segurança do seu sistema, como a execução acidental de um script malicioso como root ou de programa desconhecido, poderá colocar os discos como somente leitura e evitar danos mais sérios.

```
Mantendo ALT pressionado, tecle Print Screen e depois U.
```

Otimizando o desempenho do history com navegação contextual

Como sabemos o ambiente shell do GNU/Linux, no caso o bash, mantém no arquivo **.bash_history** uma lista com o histórico dos últimos comandos digitados. Com isso e o uso das teclas direcionais *UP* e *DOWN* nos permitem "navegar" por esta lista, de modo a retornar com um comando já utilizado e que esteja em nosso histórico armazenado.

Porém por padrão esta navegação será por toda gama de comando já utilizados, o que por vezes faz com que percamos até mais tempo necessário do que se digitarmos novamente o comando.

Com uma dica simples veremos então como fazer com que esta navegação seja otimizada de forma a permitir uma filtragem no histórico de comandos bastando inserir alguns caracteres do mesmo antes de usarmos as setas de navegação.

Agora as setas farão uma procura por contexto. Se você não digitar nada, o efeito será o mesmo que antes, mas se você digitar um caractere e pressionar a seta, ele só irá mostrar os comandos que comecem com aquele caractere. Portanto com este ajuste, se você digitar "ls" e pressionar a seta ele vai navegar apenas nos comandos que começam com "ls".

Para que isso funcione desta forma primeiramente iremos criar no diretório $HOME do usuário desejado o arquivo oculto de nome **.inputrc** com o seguinte conteúdo abaixo:

```
“\e[A”: history-search-backward
“\e[B”: history-search-forward
```

Agora basta fechar a seção atual e abrir uma nova para que a navegação no histórico dos comandos passe a funcionar desta forma mais otimizada.

Notas:

- Por padrão o Linux armazena no **.bash_history** os últimos 500 comandos utilizados, mais este número pode ser modificado editando o seu arquivo **.bashrc** e adicionado as

seguintes linhas:

```
export HISTFILESIZE=XXXX
export HISTSIZE=XXXX
```

Onde, XXXX deve ser substituído pela quantidade desejada.

- Como configuração padrão do sistema como um todo existe o arquivo **/etc/inputrc**, ou seja, caso se deseje que estas novas configurações passem a valer para todos usuários do sistema basta adicionar aquelas 2 linhas do **.inputrc** neste arquivo.

Usando "grep" com resultados coloridos

Quem costuma usar o **grep** para fazer filtragens, pode se beneficiar desta pequena e simples dica, fazendo a saída dos resultados ficarem coloridas em destaque.

```
grep --color=auto
```

Vamos a um exemplo pratico para entender melhor:

```
ps aux |grep --color=auto tty
```

Nota:

Quem gostar do resultado e desejar deixar como padrão, basta editar seu arquivo **~/.bashrc** criando um **alias** para o comando **grep** conforme abaixo demonstrado.

1. Abra o arquivo em seu editor de texto favorito.

```
vim ~/.bashrc
```

2. Adicione a linha baixo no mesmo, e salve o arquivo.

```
alias grep='grep --color=auto'
```

3. Agora, basta executar o comando abaixo que este recurso será padrão para este seu usuário.

```
source ~/.bashrc
```

Obtendo ajuda

Nosso maior aliado

Os comandos **--help** e **man** podem ser consideradas as duas ferramentas mais importantes em uma linha de comando.

Praticamente todos os comando entendem a opção **-h** (ou **--help**), a qual produzirá uma descrição breve e útil do comando e suas opções, e então volta para o terminal. Tente **"man -h"** ou **"man --help"** para ver isso em ação.

Todo comando e quase toda aplicação em Linux terá um arquivo man (manual), e encontrá-lo será muito simples. Basta digitar **"man comando"** para surgir um manual extenso para o comando especificado. Por exemplo, **"man mv"** mostrará o manual de **mv** (Move).

Mova para cima ou para baixo no arquivo **man** utilizando as teclas **Page UP** e **Page Down** ou as setas no teclado, e retorne para a linha de comando teclando **q**.

"man man" mostrará a entrada do manual para o comando **man**, e este é um bom lugar para começar!

"man intro" é especialmente útil, pois mostrará a "Introdução para comandos do usuário" que é uma introdução breve e bem escrita sobre a linha de comando.

Além disso, há as páginas de **info**, que geralmente serão mais detalhados, se aprofundando mais do que as páginas **man**. Tente **"info info"** para uma introdução às páginas info.

Procurando por arquivos "man"

Se você não está certo de qual comando ou aplicação você precisa usar, você pode tentar procurando os manuais (arquivos "man").

- **"man -k foo"** irá procurar manuais para *foo*. Tente **"man -k nautilus"** para ver como isso funciona.
    - Observe que isso é o mesmo que o comando **apropos**.
- **"man -f foo"** procura apenas os títulos dos manuais do seu sistema. Tente **"man -f gnome"**, por exemplo.
    - Isso é o mesmo que o comando **whatis**.

Outras fontes de consulta

Para maiores informações e detalhes sobre os comandos aqui apresentados, além da consulta as páginas de manuais do seu sistema recomendamos uma visita aos sites abaixo, que também serviram de poderosa fonte de pesquisa para desenvolvimento desta página.

- Guia Foca GNU/Linux.
- Man pages, tutoriais básicos de BASH, e shell script.
- Guia de Referência do Linux.
- Wikipédia-PT.
- Man pages em português

Informações adicionais

- AptGet - Howto - usando o apt-get para instalar pacotes pela linha de comando.
- AdicionandoRepositorios - adicionando os repositórios Universe/Multiverse usando a linha de comando.

Créditos

Wikifier: arlei **Atualizado em:** 12/08/2007 **Mantenedor:** arlei

Time de Documentação do Ubuntu Brasil

CategoryDocumentacao

ComandosBasicos (editada pela última vez em 2011-09-19 23:23:10 por localhost)