

# Guia Foca GNU/Linux

## Capítulo 10 - Comandos Diversos

---

Comandos de uso diversos no sistema.

---

### 10.1 clear

Limpa a tela e posiciona o cursor no canto superior esquerdo do vídeo.

```
clear
```

---

### 10.2 date

Permite ver/modificar a Data e Hora do Sistema. Você precisa estar como utilizador root para modificar a data e hora. Muitos programas do sistema, arquivos de registro (log) e tarefas agendadas funcionam com base na data e hora fornecidas pelo sistema, assim esteja consciente das modificações que a data/hora pode trazer a estes programas (principalmente em se tratando de uma rede com muitos utilizadores) .

```
date MesDiaHoraMinuto[AnoSegundos]
```

Onde:

MesDiaHoraMinuto[AnoSegundos]

São respectivamente os números do mês, dia, hora e minutos sem espaços. Opcionalmente você pode especificar o Ano (com 2 ou 4 dígitos) e os Segundos.

**+ [FORMATO]**

Define o formato da listagem que será usada pelo comando `date`. Os seguintes formatos são os mais usados:

- `%d` - Dia do Mês (00-31).
- `%m` - Mês do Ano (00-12).
- `%Y` - Ano (dois dígitos).
- `%Y` - Ano (quatro dígitos).
- `%H` - Hora (00-24).
- `%I` - Hora (00-12).
- `%M` - Minuto (00-59).
- `%j` - Dia do ano (1-366).
- `%p` - AM/PM (útil se utilizado com `%d`).

- `%r` - Formato de 12 horas completo (hh:mm:ss AM/PM).
- `%T` - Formato de 24 horas completo (hh:mm:ss).
- `%w` - Dia da semana (0-6).

Outros formatos podem ser obtidos através da página de manual do `date`.

Para maiores detalhes, veja a página de manual do comando `date`.

Para ver a data atual digite: `date`

Se quiser mudar a Data para 25/12 e a hora para 08:15 digite: `date 12250815`

Para mostrar somente a data no formato dia/mês/ano: `date +%d/%m/%Y`

## 10.3 df

Mostra o espaço livre/ocupado de cada partição.

`df [opções]`

onde:

*opções*

`-a`

Inclui sistemas de arquivos com 0 blocos.

`-h, --human-readable`

Mostra o espaço livre/ocupado em *MB*, *KB*, *GB* ao invés de blocos.

`-H`

Idêntico a `-h` mas usa 1000 ao invés de 1024 como unidade de cálculo.

`-k`

Lista em Kbytes.

`-l`

Somente lista sistema de arquivos locais.

`-m`

Lista em Mbytes (equivalente a `--block-size=1048576`).

`--sync`

Executa o `sync` antes de mostrar os dados.

`-T`

Lista o tipo de sistema de arquivos de cada partição

`-t tipo`

Lista somente sistema de arquivos do tipo *tipo*.

-x *tipo*

Não lista sistemas de arquivos do tipo *tipo*.

Exemplos: `df`, `df -h`, `df -t vfat`.

---

## 10.4 ln

Cria links para arquivos e diretórios no sistema. O link é um mecanismo que faz referência a outro arquivo ou diretório em outra localização. O link em sistemas GNU/Linux faz referência reais ao arquivo/diretório podendo ser feita cópia do link (será copiado o arquivo alvo), entrar no diretório (caso o link faça referência a um diretório), etc.

`ln [opções] [origem] [link]`

Onde:

*origem*

Diretório ou arquivo de onde será feito o link.

*link*

Nome do link que será criado.

*opções*

-s

Cria um link simbólico. Usado para criar ligações com o arquivo/diretório de destino.

-v

Mostra o nome de cada arquivo antes de fazer o link.

-d

Cria um hard link para diretórios. Somente o root pode usar esta opção.

Existem 2 tipos de links: *simbólicos* e *hardlinks*.

- O *link simbólico* cria um arquivo especial no disco (do tipo link) que tem como conteúdo o caminho para chegar até o arquivo alvo (isto pode ser verificado pelo tamanho do arquivo do link). Use a opção `-s` para criar links simbólicos.
- O *hardlink* faz referência ao mesmo inodo do arquivo original, desta forma ele será perfeitamente idêntico, inclusive nas permissões de acesso, ao arquivo original.

Ao contrário dos links simbólicos, não é possível fazer um hardlink para um diretório ou fazer referência a arquivos que estejam em partições diferentes.

Observações:

- Se for usado o comando `rm` com um link, somente o link será removido.
- Se for usado o comando `cp` com um link, o arquivo original será copiado ao invés do link.
- Se for usado o comando `mv` com um link, a modificação será feita no link.
- Se for usado um comando de visualização (como o `cat`), o arquivo original será visualizado.

Exemplos:

- `ln -s /dev/ttyS1 /dev/modem` - Cria o link `/dev/modem` para o arquivo `/dev/ttyS1`.
- `ln -s /tmp ~/tmp` - Cria um link `~/tmp` para o diretório `/tmp`.

## 10.5 du

Mostra o espaço ocupado por arquivos e sub-diretórios do diretório atual.

`du [opções]`

onde:

*opções*

`-a, --all`

Mostra o espaço ocupado por todos os arquivos.

`-b, --bytes`

Mostra o espaço ocupado em bytes.

`-c, --total`

Faz uma totalização de todo espaço listado.

`-D`

Não conta links simbólicos.

`-h, --human`

Mostra o espaço ocupado em formato legível por humanos (Kb, Mb) ao invés de usar blocos.

`-H`

Como o anterior mas usa 1000 e não 1024 como unidade de cálculo.

`-k`

Mostra o espaço ocupado em Kbytes.

`-m`

Mostra o espaço ocupado em Mbytes.

`-S, --separate-dirs`

Não calcula o espaço ocupado por sub-diretórios.

`-x`

Não faz a contagem de diretórios em sistemas de arquivos diferentes do atual.

Exemplo: `du -h`, `du -hc`.

---

## 10.6 find

Procura por arquivos/diretórios no disco. `find` pode procurar arquivos através de sua data de modificação, tamanho, etc através do uso de opções. `find`, ao contrário de outros programas, usa opções longas através de um "-".

`find [diretório] [opções/expressão]`

Onde:

*diretório*

Inicia a procura neste diretório, percorrendo seu sub-diretórios.

*opções/expressão*

`-name [expressão]`

Procura pelo nome [expressão] nos nomes de arquivos e diretórios processados.

`-depth`

Processa os sub-diretórios primeiro antes de processar os arquivos do diretório principal.

`-maxdepth [num]`

Faz a procura até [num] sub-diretórios dentro do diretório que está sendo pesquisado.

`-mindepth [num]`

Não faz nenhuma procura em diretórios menores que [num] níveis.

`-mount, -xdev`

Não faz a pesquisa em sistemas de arquivos diferentes daquele de onde o comando `find` foi executado.

`-amin [num]`

Procura por arquivos que foram acessados [num] minutos atrás. Caso for antecedido por "-", procura por arquivos que foram acessados entre [num] minutos atrás até agora.

-atime [num]

Procura por arquivos que foram acessados [num] dias atrás. Caso for antecedido por "-", procura por arquivos que foram acessados entre [num] dias atrás e a data atual.

-gid [num]

Procura por arquivos que possuam a identificação numérica do grupo igual a [num].

-group [nome]

Procura por arquivos que possuam a identificação de nome do grupo igual a [nome].

-uid [num]

Procura por arquivos que possuam a identificação numérica do utilizador igual a [num].

-user [nome]

Procura por arquivos que possuam a identificação de nome do utilizador igual a [nome].

-inum [num]

Procura por arquivos que estão localizados no inodo [num].

-links [num]

Procura por arquivos que possuem [num] links como referência.

-mmin [num]

Procura por arquivos que tiveram seu conteúdo modificado há [num] minutos. Caso for antecedido por "-", procura por arquivos que tiveram seu conteúdo modificado entre [num] minutos atrás até agora.

-mtime [num]

Procura por arquivos que tiveram seu conteúdo modificado há [num] dias. Caso for antecedido por "-", procura por arquivos que tiveram seu conteúdo modificado entre [num] dias atrás até agora.

-ctime  
[num]

Procura por arquivos que teve seu status modificado há [num] dias. Caso for antecedido por "-", procura por arquivos que tiveram seu conteúdo modificado entre [num] dias atrás até agora.

-  
no  
us  
er

Procura por arquivos que não correspondam a identificação do utilizador atual.

-  
n  
o  
g  
r  
o  
u  
p

Procura por arquivos que não correspondam a identificação do grupo do utilizador atual.

Procura por arquivos que possuam os modos de permissão [modo]. Os [modo] de permissão pode ser numérico (octal) ou literal.

O arquivo foi acessado [num] vezes antes de ter seu status modificado.

Procura por arquivos que tiverem o tamanho [num]. [num] pode ser antecedido de "+" ou "-" para especificar um arquivo maior ou menor que [num]. A opção -size pode ser seguida de:

- b - Especifica o tamanho em blocos de 512 bytes. É o padrão caso [num] não seja acompanhado de nenhuma letra.
- c - Especifica o tamanho em bytes.
- k - Especifica o tamanho em Kbytes.

Procura por arquivos do [tipo] especificado. Os seguintes tipos são aceitos:

- b - bloco
- c - character
- d - diretório
- p - pipe
- f - arquivo regular
- l - link simbólico
- s - sockete

A maior parte dos argumentos numéricos podem ser precedidos por "+" ou "-". Para detalhes sobre outras opções e argumentos, consulte a página de manual.

Exemplo:



- `find / -name grep` - Procura no diretório raiz e sub-diretórios um arquivo/diretório chamado `grep`.
  - `find / -name grep -maxdepth 3` - Procura no diretório raiz e sub-diretórios até o 3o. nível, um arquivo/diretório chamado `grep`.
  - `find . -size +1000k` - Procura no diretório atual e sub-diretórios um arquivo com tamanho maior que 1000 kbytes (1Mbyte).
  - `find / -mmin 10` - Procura no diretório raiz e sub-diretórios um arquivo que foi modificado há 10 minutos atrás.
  - `find / -links 4` - Procura no diretório raiz e sub-diretórios, todos os arquivos que possuem 4 links como referência.
- 

## 10.7 free

Mostra detalhes sobre a utilização da memória RAM do sistema.

`free [opções]`

Onde:

*opções*

`-b`

Mostra o resultado em bytes.

`-k`

Mostra o resultado em Kbytes.

`-m`

Mostra o resultado em Mbytes.

`-o`

Ocultar a linha de buffers.

`-t`

Mostra uma linha contendo o total.

`-s [num]`

Mostra a utilização da memória a cada [num] segundos.

O `free` é uma interface ao arquivo `/proc/meminfo`.

---

## 10.8 grep

Procura por um texto dentro de um arquivo(s) ou no dispositivo de entrada padrão.

```
grep [expressão] [arquivo] [opções]
```

Onde:

*expressão*

palavra ou frase que será procurada no texto. Se tiver mais de 2 palavras você deve identifica-la com aspas "" caso contrário o `grep` assumirá que a segunda palavra é o arquivo!

*arquivo*

Arquivo onde será feita a procura.

*opções*

`-A [número]`

Mostra o [número] de linhas após a linha encontrada pelo `grep`.

`-B [número]`

Mostra o [número] de linhas antes da linha encontrada pelo `grep`.

`-f [arquivo]`

Especifica que o texto que será localizado, esta no arquivo [arquivo].

`-h, --no-filename`

Não mostra os nomes dos arquivos durante a procura.

`-i, --ignore-case`

Ignora diferença entre maiúsculas e minúsculas no texto procurado e arquivo.

`-n, --line-number`

Mostra o nome de cada linha encontrada pelo `grep`.

`-U, --binary`

Trata o arquivo que será procurado como binário.

Se não for especificado o nome de um arquivo ou se for usado um hífen "-", `grep` procurará a string no dispositivo de entrada padrão. O `grep` faz sua pesquisa em arquivos texto. Use o comando `zgrep` para pesquisar diretamente em arquivos compactados com `gzip`, os comandos e opções são as mesmas.

Exemplos: `grep "capitulo" texto.txt, ps ax | grep inetd, grep "capitulo" texto.txt -A 2 -B 2.`

---

## 10.9 head

Mostra as linhas iniciais de um arquivo texto.

`head [opções]`

Onde:

`-c [numero]`

Mostra o [numero] de bytes do início do arquivo.

`-n [numero]`

Mostra o [numero] de linhas do início do arquivo. Caso não for especificado, o `head` mostra as 10 primeiras linhas.

Exemplos: `head teste.txt`, `head -n 20 teste.txt`.

---

## 10.10 nl

Mostra o número de linhas junto com o conteúdo de um arquivo.

`nl [opções] [arquivo]`

Onde:

`-f [opc]`

Faz a filtragem de saída de acordo com [opc]:

`a`

Numera todas as linhas.

`t`

Não numera linhas vazias.

`n`

Numera linhas vazias.

`texto`

Numera somente linhas que contém o [texto].

`-v [num]`

Número inicial (o padrão é 1).

`-i [num]`

Número de linhas adicionadas a cada linha do arquivo (o padrão é 1).

Exemplos: `nl /etc/passwd, nl -i 2 /etc/passwd.`

---

## 10.11 more

Permite fazer a paginação de arquivos ou da entrada padrão. O comando `more` pode ser usado como comando para leitura de arquivos que ocupem mais de uma tela. Quando toda a tela é ocupada, o `more` efetua uma pausa e permite que você pressione `Enter` ou `espaço` para continuar avançando no arquivo sendo visualizado. Para sair do `more` pressione `q`.

`more [arquivo]`

Onde: *arquivo* É o arquivo que será paginado.

Para visualizar diretamente arquivos texto compactados pelo `gzip` `.gz` use o comando `zmore`.

Exemplos: `more /etc/passwd, cat /etc/passwd | more.`

---

## 10.12 less

Permite fazer a paginação de arquivos ou da entrada padrão. O comando `less` pode ser usado como comando para leitura de arquivos que ocupem mais de uma tela. Quando toda a tela é ocupada, o `less` efetua uma pausa (semelhante ao `more`) e permite que você pressione Seta para Cima e Seta para Baixo ou `PgUP`/`PgDown` para fazer o rolamento da página. Para sair do `less` pressione `q`.

`less [arquivo]`

Onde: *arquivo* É o arquivo que será paginado.

Para visualizar diretamente arquivos texto compactados pelo utilitário `gzip` (arquivos `.gz`), use o comando `zless`.

Exemplos: `less /etc/passwd, cat /etc/passwd | less`

---

## 10.13 sort

Organiza as linhas de um arquivo texto ou da entrada padrão. A organização é feita por linhas e as linhas são divididas em *campos* que é a ordem que as palavras aparecem na linha separadas por um delimitador (normalmente um espaço).

`sort [opções] [arquivo]`

Onde:

*arquivo*

É o nome do arquivo que será organizado. Caso não for especificado, será usado o dispositivo de entrada padrão (normalmente o teclado ou um "|").

*opções*

`-b`

Ignora linhas em branco.

-d

Somente usa letras, dígitos e espaços durante a organização.

-f

Ignora a diferença entre maiúsculas e minúsculas.

-r

Inverte o resultado da comparação.

-n

Caso estiver organizando um campo que contém números, os números serão organizados na ordem aritmética. Por exemplo, se você tiver um arquivo com os números

100

10

50

Usando a opção `-n`, o arquivo será organizado desta maneira:

10

50

100

Caso esta opção **não** for usada com o `sort`, ele organizará como uma listagem alfabética (que começam de a até z e do 0 até 9)

10

100

50

-c

Verifica se o arquivo já esta organizado. Caso não estiver, retorna a mensagem "disorder on *arquivo*".

-o *arquivo*

Grava a saída do comando `sort` no *arquivo*.

-m *arquivo1 arquivo2*

Combina o conteúdo de *arquivo1* e *arquivo2* gerando um único arquivo. Os dois arquivos precisam estar ordenados antes de se utilizar esta opção.

-i

Ignora os caracteres fora da faixa octal ASCII 040-0176 durante a organização.

-t *character*

Usa *character* como delimitador durante a organização de linhas. Por padrão é usado um *espaço em branco* como delimitador de caracteres.

+*num1* -*num2*

Especifica qual o campo dentro na linha que será usado na organização. O(s) campo(s) usado(s) para organização estará entre +*num1* e +*num2*. O delimitador padrão utilizado é um *espaço em branco* (use a opção -t para especificar outro). A contagem é iniciada em "0". Caso não for especificada, a organização é feita no primeiro campo. Caso -*num2* não seja especificado, a organização será feita usando a coluna +*num1* até o fim da linha.

-k *num1*, *num2*

Esta é uma alternativa ao método acima para especificar as chaves de organização. O uso é idêntico, mas o delimitador é iniciado em "1".

Abaixo, exemplos de uso do comando `sort`:

- `sort texto.txt` - Organiza o arquivo `texto.txt` em ordem crescente.
- `sort texto.txt -r` - Organiza o conteúdo do arquivo `texto.txt` em ordem decrescente.
- `cat texto.txt | sort` - Faz a mesma coisa que o primeiro exemplo, só que neste caso a saída do comando `cat` é redirecionado a entrada padrão do comando `sort`.
- `sort -f texto.txt` - Ignora diferenças entre letras maiúsculas e minúsculas durante a organização.
- `sort +1 -3 texto.txt` - Organiza o arquivo `texto.txt` usando como referência a segunda até a quarta palavra (segundo ao quarto campo) que constam naquela linha.
- `sort -t : +2 -3 passwd` - Organiza o arquivo `passwd` usando como referência a terceira até a quarta palavra (terceiro ao quarto campo). Note que a opção -t especifica o caracter ":" como delimitador de campos ao invés do espaço. Neste caso, o que estiver após ":" será considerado o próximo campo.

---

## 10.14 tail

Mostra as linhas finais de um arquivo texto.

`tail [opções]`

Onde:

-c [numero]

Mostra o [numero] de bytes do final do arquivo.

-n [numero]

Mostra o [numero] de linhas do final do arquivo.

Exemplos: `tail teste.txt`, `tail -n 20 teste.txt`.

---

## 10.15 time

Mede o tempo gasto para executar um processo (programa).

`time [comando]`

Onde: *comando* é o comando/programa que deseja medir o tempo gasto para ser concluído.

Exemplo: `time ls`, `time find / -name crontab`.

---

## 10.16 touch

Muda a data e hora que um arquivo foi criado. Também pode ser usado para criar arquivos vazios. Caso o `touch` seja usado com arquivos que não existam, por padrão ele criará estes arquivos.

`touch [opções] [arquivos]`

Onde:

*arquivos*

Arquivos que terão sua data/hora modificados.

*opções*

`-t MMDDhhmm[ANO.segundos]`

Usa Mês (MM), Dias (DD), Horas (hh), minutos (mm) e opcionalmente o ANO e segundos para modificação do(s) arquivos ao invés da data e hora atual.

`-a, --time=atime`

Faz o `touch` mudar somente a data e hora do acesso ao arquivo.

`-c, --no-create`

Não cria arquivos vazios, caso os *arquivos* não existam.

`-m, --time=mtime`

Faz o `touch` mudar somente a data e hora da modificação.

`-r [arquivo]`

Usa as horas no `[arquivo]` como referência ao invés da hora atual.

Exemplos:

- `touch teste` - Cria o arquivo `teste` caso ele não existir.
- `touch -t 10011230 teste` - Altera da data e hora do arquivo para 01/10 e 12:30.
- `touch -t 120112301999.30 teste` - Altera da data, hora ano, e segundos do arquivo para 01/12/1999 e 12:30:30.

- `touch -t 12011200 *` - Altera a data e hora do arquivo para 01/12 e 12:00.

---

## 10.17 uptime

Mostra o tempo de execução do sistema desde que o computador foi ligado.

```
uptime
```

---

## 10.18 dmesg

Mostra as mensagens de inicialização do kernel. São mostradas as mensagens da última inicialização do sistema.

```
dmesg | less
```

---

## 10.19 mesg

Permite ou não o recebimentos de requisições de `talk` de outros utilizadores.

```
mesg [y/n]
```

Onde: `y` permite que você receba "talks" de outros utilizadores.

Digite `mesg` para saber se você pode ou não receber "talks" de outros utilizadores. Caso a resposta seja "n" você poderá enviar um talk para alguém mas o seu sistema se recusará em receber talks de outras pessoas.

É interessante colocar o comando `mesg y` em seu arquivo de inicialização `.bash_profile` para permitir o recebimento de "talks" toda vez que entrar no sistema.

Para detalhes sobre como se comunicar com outros utilizadores, veja o comando [talk, Seção 11.8](#).

---

## 10.20 echo

Mostra mensagens. Este comando é útil na construção de scripts para mostrar mensagens na tela para o utilizador acompanhar sua execução.

```
echo [mensagem]
```

A opção `-n` pode ser usada para que não ocorra o salto de linha após a mensagem ser mostrada.

---

## 10.21 su

Permite o utilizador mudar sua identidade para outro utilizador sem fazer o logout. Útil para executar um programa ou comando como root sem ter que abandonar a seção atual.

```
su [utilizador]
```



Onde: *utilizador* é o nome do utilizador que deseja usar para acessar o sistema. Se não digitado, é assumido o utilizador `root`.

Será pedida a senha do superutilizador para autenticação. Digite `exit` quando desejar retornar a identificação de utilizador anterior.

---

## 10.22 sync

Grava os dados do cache de disco na memória RAM para todos os discos rígidos e flexíveis do sistema. O cache um mecanismo de aceleração que permite que um arquivo seja armazenado na memória ao invés de ser imediatamente gravado no disco, quando o sistema estiver ocioso, o arquivo é gravado para o disco.

O GNU/Linux procura utilizar toda memória RAM disponível para o cache de programas acelerando seu desempenho de leitura/gravação.

`sync`

O uso do `sync` é útil em disquetes quando gravamos um programa e precisamos que os dados sejam gravados imediatamente para retirar o disquete da unidade. Mas o método recomendado é especificar a opção `sync` durante a montagem da unidade de disquetes (para detalhes veja [fstab, Seção 5.13.1](#)).

---

## 10.23 uname

Retorna o nome e versão do kernel atual.

`uname`

---

## 10.24 reboot

Reinicia o computador.

---

## 10.25 shutdown

Desliga/reinicia o computador imediatamente ou após determinado tempo (programável) de forma segura. Todos os utilizadores do sistema são avisados que o computador será desligado. Este comando somente pode ser executado pelo utilizador `root` ou quando é usada a opção `-a` pelos utilizadores cadastrados no arquivo `/etc/shutdown.allow` que estejam logados no console virtual do sistema.

`shutdown [opções] [hora] [mensagem]`

*hora*

Momento que o computador será desligado. Você pode usar `HH:MM` para definir a hora e minuto, `MM` para definir minutos, `+SS` para definir após quantos segundos, ou `now` para imediatamente (equivalente a `+0`).

O `shutdown` criará o arquivo `/etc/nologin` para não permitir que novos utilizadores façam login no sistema (com exceção do `root`). Este arquivo é removido caso a execução do `shutdown` seja cancelada (opção `-c`) ou após o sistema ser reiniciado.

*mensagem*

Mensagem que será mostrada a todos os utilizadores alertando sobre o reinício/desligamento do sistema.

*opções*

`-h`

Inicia o processo para desligamento do computador.

`-r`

Reinicia o sistema

`-c`

Cancela a execução do `shutdown`. Você pode acrescentar uma mensagem avisando aos utilizadores sobre o fato.

`-a`

Permite que os nomes de utilizadores contidos no arquivo `/etc/shutdown.allow` possam utilizar o `shutdown` para reinicializar/desligar o sistema. Deve ser colocado um nome de utilizador por linha. O limite máximo de utilizadores neste arquivo é de 32.

Este arquivo é útil quando o `shutdown` é usado para controlar o pressionamento das teclas `CTRL+ALT+DEL` no `/etc/inittab`.

`-k`

Simula o desligamento/reinício do sistema, enviando mensagem aos utilizadores.

`-f`

Não executa a checagem do sistema de arquivos durante a inicialização do sistema. Este processo é feito gravando-se um arquivo `/fastboot` que é interpretado pelos scripts responsáveis pela execução do `fsck` durante a inicialização do sistema.

`-F`

Força a checagem do sistema de arquivos durante a inicialização. É gravado um arquivo chamado `/forcefsck` que é interpretado pelos scripts responsáveis pela execução do `fsck` durante a inicialização do sistema.

`-n`

Faz com que o `shutdown` ignore a execução do `init` fechando todos os processos.

`-t [num]`

Faz com que o `shutdown` envie um sinal de término aos processos e aguarde `[num]` segundos antes de enviar o sinal `KILL`.

O `shutdown` envia uma mensagem a todos os utilizadores do sistema alertando sobre o desligamento durante os 15 minutos restantes e assim permite que finalizem suas tarefas. Após isto, o `shutdown` muda o nível de

execução através do comando `init` para 0 (desligamento), 1 (modo monoutilizador), 6 (reinicialização). É recomendado utilizar o símbolo "&" no final da linha de comando para que o `shutdown` seja executado em segundo plano.

Quando restarem apenas 5 minutos para o reinício/desligamento do sistema, o programa `login` será desativado, impedindo a entrada de novos utilizadores no sistema.

O programa `shutdown` pode ser chamado pelo `init` através do pressionamento da combinação das teclas de reinicialização `CTRL+ALT+DEL` alterando-se o arquivo `/etc/inittab`. Isto permite que somente os utilizadores autorizados (ou o `root`) possam reinicializar o sistema.

Exemplos:

- `"shutdown -h now"` - Desligar o computador imediatamente.
- `"shutdown -r now"` - Reinicia o computador imediatamente.
- `"shutdown 19:00 A manutenção do servidor será iniciada às 19:00"` - Faz o computador entrar em modo monoutilizador (`init 1`) às 19:00 enviando a mensagem *A manutenção do servidor será iniciada às 19:00* a todos os utilizadores conectados ao sistema.
- `"shutdown -r 15:00 O sistema será reiniciado às 15:00 horas"` - Faz o computador ser reiniciado (`init 6`) às 15:00 horas enviando a mensagem *O sistema será reiniciado às 15:00 horas* a todos os utilizadores conectados ao sistema.
- `shutdown -r 20` - Faz o sistema ser reiniciado após 20 minutos.
- `shutdown -c` - Cancela a execução do `shutdown`.
- `shutdown -t 30 -r 20` - Reinicia o sistema após 20 minutos, espera 30 segundos após o sinal de término para enviar o sinal KILL a todos os programas abertos.

---

## 10.26 wc

Conta o número de palavras, bytes e linhas em um arquivo ou entrada padrão. Se as opções forem omitidas, o `wc` mostra a quantidade de linhas, palavras, e bytes.

`wc [opções] [arquivo]`

Onde:

*arquivo*

Arquivo que será verificado pelo comando `wc`.

*opções*

`-c, --bytes`

Mostra os bytes do arquivo.

`-w, --words`

Mostra a quantidade de palavras do arquivo.

`-l, --lines`

Mostra a quantidade de linhas do arquivo.

A ordem da listagem dos parâmetros é única, e modificando a posição das opções não modifica a ordem que os parâmetros são listados.

Exemplo:

- `wc /etc/passwd` - Mostra a quantidade de linhas, palavras e letras (bytes) no arquivo `/etc/passwd`.
  - `wc -w /etc/passwd` - Mostra a quantidade de palavras.
  - `wc -l /etc/passwd` - Mostra a quantidade de linhas.
  - `wc -l -w /etc/passwd` - Mostra a quantidade de linhas e palavras no arquivo `/etc/passwd`.
- 

## 10.27 seq

Imprime uma seqüência de números começando em [primeiro] e terminando em [último], utilizando [incremento] para avançar.

```
seq [opções] [primeiro] [incremento] [último]
```

Onde:

*primeiro*

Número inicial da seqüência.

*incremento*

Número utilizado para avançar na seqüência.

*último*

Número final da seqüência.

*opções*

`-f, --format=[formato]`

Formato de saída dos números da seqüência. Utilize o estilo do printf para ponto flutuante (valor padrão: %g).

`-s, --separator=[string]`

Usa [string] para separar a seqüência de números (valor padrão: \n).

`-w, --equal-width`

Insere zeros na frente dos números mantendo a seqüência alinhada.

Observações:

- Se [primeiro] ou [incremento] forem omitidos, o valor padrão 1 será utilizado.
- Os números recebidos são interpretados como números em ponto flutuante.
- [incremento] deve ser positivo se [primeiro] for menor do que o último, e negativo caso contrário.
- Quando utilizarmos a opção --format, o argumento deve ser exatamente %e, %f ou %g.

Exemplos: `seq 0 2 10`, `seq -w 0 10`, `seq -f%f 0 10`, `seq -s" , " 0 10`

---

## 10.28 chattr

Modifica atributos de arquivos/diretórios. Não confunda atributos de arquivo com permissões de acesso ([Permissões de acesso a arquivos e diretórios, Capítulo 13](#)), os atributos são diferentes e definem outras características especiais para os arquivos/diretórios especificados.

`chattr [opções] [atributos] [arquivos/diretórios]`

Onde:

*arquivos/diretórios*

Arquivos/Diretórios que terão os atributos modificados. Podem ser usados curingas

*opções*

-R

Modifica atributos em subdiretórios

-V

Mostra detalhes sobre a modificação de atributos.

*atributos*

Os atributos de arquivos/diretórios podem ser especificados da seguinte maneira:

- + - Adiciona o atributo
- - - Remove o atributo
- = - Define o atributo exatamente como especificado

Os atributos são os seguintes:

- A - Não modifica a hora de acesso de arquivos. Poder aumentar consideravelmente a performance em Notebooks devido a diminuição de I/O no disco rígido. Quando especificada em diretórios, faz com que todos os arquivos e subdiretórios residentes nele não tenham a hora de acesso modificada.

Este atributo funciona apenas em kernels 2.2 e superiores

- `a` - Append-Only - Arquivos com este atributo podem somente ser gravados em modo incrementais (o conteúdo poderá somente ser adicionado ao final do arquivo). Eles não poderão ser removidos, renomeados e novos links não poderão ser criados para estes arquivos.

Em diretórios faz com que os arquivos sejam apenas adicionados. Somente o root pode especificar ou retirar este atributo.

- `c` - Permite compactação nos arquivos especificados de forma transparente para o utilizador. Durante a leitura, o kernel retorna dados descompactados e durante a gravação os dados são compactados e gravados no disco.

Este atributo ainda não foi totalmente implementado no código atual do kernel.

- `d` - Este atributo não é usado pelo kernel, mas faz com que o programa `dump` evitar backup dos arquivos marcados com este atributo.
- `i` - Imutável - Arquivos imutáveis não podem ser modificados, os dados também não podem ser gravados para estes arquivos, não podem ser removidos, renomeados. Até mesmo o utilizador root não poderá modificar estes arquivos.

Em diretórios, faz com que arquivos não possam ser adicionados ou apagados. Somente o utilizador root pode especificar ou retirar este atributo.

- `s` - O arquivo especificado é marcado como "apagamento seguro"; quando o arquivo é apagado, seus blocos são zerados e gravados de volta no disco (eliminando qualquer possibilidade de recuperação).
- `S` - Faz a gravação imediatamente para o arquivo especificado. É como especificar a opção "sync" na montagem do sistema de arquivos ext2, mas afeta somente os arquivos especificados. Não tem efeito em diretórios.
- `u` - O arquivo especificado é marcado como recuperável. Quando o arquivo é apagado, seu conteúdo é salvo para permitir futura recuperação.

Este atributo ainda não foi implementado totalmente no código atual do kernel.

Os atributos de arquivos/diretórios são visualizados através do utilitário `lsattr`. Existem patches para os kernels da série 2.2 que adicionam o suporte experimental aos atributos "c" e "u".

Exemplos:

- `chattr +Aacdissu teste.txt` - Adiciona todos os atributos
- `chattr =ASs teste.txt` - Define os atributos para "ASs"
- `chattr +i -A teste.txt` - Retira o atributo "A" e adiciona "i"
- `chattr = teste.txt` - Retira todos os atributos

---

## 10.29 lsattr

Lista atributos de um arquivo/diretório. Os atributos podem ser modificados através do comando `chattr`.

`lsattr [opções] [arquivos/diretórios]`

Onde:

*arquivos/diretórios*

Arquivos/diretórios que deseja listar os atributos. Podem ser usados curingas.

*opções*

`-a`

Lista todos os arquivos, incluindo ocultos (iniciando com um ".").

`-d`

Lista os atributos de diretórios ao invés de listar os arquivos que ele contém.

`-R`

Faz a listagem em diretórios e subdiretórios.

`-v`

Mostra versões dos arquivos.

Caso seja especificado sem parâmetros, o `lsattr` listará os atributos de todos os arquivos e diretórios do diretório atual. O `lsattr` mostrará mensagens de erro caso seja usado em um diretório de pontos de montagem ou arquivos que não sejam *ext2*.

Exemplo: `lsattr -d, lsattr -R, lsattr -R *.txt`

---

## 10.30 cut

Mostra seções de cada linha do arquivo dependendo das opções passadas ao programa.

`cut [opções] [arquivo]`

Onde:

*arquivo*

Arquivo que será verificado pelo comando `cut`.

*opções*

`-b, --bytes [bytes]`

Mostra somente a lista de [bytes] do arquivo.

`-c, --characters [numero]`

Mostra somente o [número] de caracteres no arquivo. É semelhante a opção "-b" mas tabs e espaços são tratados como qualquer caracter.

`-f, --field [campos]`

Mostra somente a lista de [campos].

`-d, --delimite [delimitador]`

Para uso com a opção -f, os campos são separados pelo primeiro caracter em [delimitador] ao invés de tabulações.

`-s`

Para uso com a opção -f, somente mostra linhas que contém o caracter separador de campos.

Devem ser especificadas opções para o funcionamento deste comando. Os bytes, campos e delimitadores podem ser especificados através de intervalos de caracteres (usando a-z), através de vírgulas (a,b,d) ou da combinação entre eles.

- `cut -b 1,3 /etc/passwd` - Pega a primeira e terceira letra (byte) de cada linha do arquivo `/etc/passwd`
- `cut -b 1,3-10 /etc/passwd` - Pega a primeira letra (byte) e terceira a décima letra de cada linha do arquivo `/etc/passwd`.
- `cut -c 1,3-10 /etc/passwd` - Pega o primeiro caracter e terceiro ao décimo caracter de cada linha do arquivo `/etc/passwd`.

---

## 10.31 cmp

Compara dois arquivos de qualquer tipo (binário ou texto). Os dois arquivos especificados serão comparado e caso exista diferença entre eles, é mostrado o número da linha e byte onde ocorreu a primeira diferença na saída padrão (tela) e o programa retorna o código de saída 1.

`cmp [arquivo1] [arquivo2] [opções]`

Opções:

`arquivo1/arquivo2`

Arquivos que serão comparados.

`opções`

`-l`

Mostra o número do byte (hexadecimal) e valores diferentes de bytes (octal) para cada diferença.

-5



Não mostra nenhuma diferença, só retorna o código de saída do programa.

Use o comando `zcmp` para comparar diretamente arquivos binários/texto compactados com `gzip`.

Exemplo: `cmp teste.txt teste1.txt`.

---

## 10.32 dirname

Obtém o nome do diretório através do caminho passado ao programa.

```
dirname [diretório/arquivo]
```

```
dirname /usr/bin/dirname, dirname /tmp/*.  
[...]
```

---

## 10.33 diff

Compara dois arquivos e mostra as diferenças entre eles. O comando `diff` é usado somente para a comparação de arquivos em formato texto. As diferenças encontradas podem ser redirecionadas para um arquivo que poderá ser usado pelo comando `patch` para aplicar as alterações em um arquivo que não contém as diferenças. Isto é útil para grandes textos porque é possível copiar somente as modificações (geradas através do `diff`, que são muito pequenas) e aplicar no arquivo para atualiza-lo (através do `patch`) ao invés de copiar a nova versão. Este é um sistema de atualização muito usado na atualização dos código fonte do kernel do GNU/Linux.

```
diff [diretório1/arquivo1] [diretório2/arquivo2] [opções]
```

Opções:

*diretório1/arquivo1* *diretório2/arquivo2*

Arquivos /diretórios que serão comparados. Normalmente é usado como primeiro arquivo/diretório o mais antigo e o mais novo como segundo.

*opções*

`-lines [num]`

Gera a diferença com [num] linhas de contexto. Por padrão o `diff` gera um arquivo com 2 linhas que é o mínimo necessário para o correto funcionamento do `patch`.

`-a`

Compara os dois arquivos como arquivos texto.

`-b`

Ignora espaços em branco como diferenças.

`-B`

Ignora linhas em branco inseridas ou apagadas nos arquivos.

-i

Ignora diferenças entre maiúsculas e minúsculas nos arquivos.

-H

Usa análise heurística para verificar os arquivos.

-N

Em uma comparação de diretórios, se o arquivo apenas existe em um diretório, trata-o como presente mas vazio no outro diretório.

-P

Em uma comparação de diretórios, se o arquivos apenas existe no segundo diretório, trata-o como presente mas vazio no primeiro diretório.

-q

Mostra somente se os dois arquivos possuem diferenças. Não mostra as diferenças entre eles.

-r

Compara diretórios e sub-diretórios existentes.

-S [nome]

Inicia a comparação de diretórios pelo arquivo [nome]. É útil quando cancelamos uma comparação.

-t

Aumenta a tabulação das diferenças encontradas.

-u

Usa o formato de comparação unificado.

Use o comando `zdiff` para comparar diretamente arquivos compactados pelo utilitário `gzip`

Use o comando `sdiff` para visualizar as linhas diferentes entre os dois arquivos em formato texto simples.

Exemplo:

- `diff texto.txt texto1.txt` - Compara o arquivo `texto.txt` com `texto1.txt` e exibe suas diferenças na tela.
- `diff -Bu texto.txt texto1.txt` - Compara o arquivo `texto.txt` com `texto1.txt` ignorando linhas em branco diferentes entre os dois arquivos e usando o formato unificado.

- `diff texto.txt texto1.txt >texto.diff` - Compara o arquivo `texto.txt` com `texto1.txt` e gera um arquivo chamado `texto.diff` contendo a diferença entre eles. Este arquivo poderá ser usado pelo `patch` para aplicar as diferenças existente entre os dois no arquivo `texto.txt`.
  - `diff -r /usr/src/linux-2.2.13 /usr/src/linux-2.2.14 >patch-2.2.14.diff` - Compara o diretório e sub-diretórios `linux-2.2.13` e `linux-2.2.14` e grava as diferenças entre eles no arquivo `patch-2.2.14.diff`.
- 

## 10.34 pr

Página arquivos texto ou a entrada padrão para impressão. Este comando faz a paginação de um arquivo texto e opcionalmente ajusta o número de colunas e mostra o resultado na saída padrão.

`pr [opções] [arquivo]`

Onde:

*arquivo*

Arquivo que será paginado para impressão.

*opções*

`+ [NUM]`

Inicia a numeração de páginas na página `[PAGINA]`

`- [NUM]`

Mostra a saída com `[NUM]` colunas.

`-c`

Imprime o caracter CTRL como "`^`" na saída padrão.

`-F, -f`

Usa avanço de página ao invés de linhas em branco para separar páginas.

`-e[caracter][tamanho]`

Usa o caracter `[caracter]` como tabulação (o padrão é `tab`) e o espaço da tabulação `[tamanho]`.

`-h [nome]`

Mostra `[nome]` ao invés do nome do arquivo no cabeçalho.

`-l [num]`

Define o número máximo de linhas por página para `[num]`.

-m

Imprime vários arquivos em paralelo, um por coluna.

-r

Ocultar mensagens de erro de abertura de arquivos.

-w [num]

Ajusta a largura da página para [num] colunas (o padrão é 72).

Exemplo: `pr -l 50 -h "Teste do comando pr" teste.txt.`

---

## 10.35 patch

Atualiza arquivos texto através das diferenças geradas pelo comando `diff`.

`patch [opções] [arquivo.diff]` ou `patch [opções] < [arquivo.diff]`

Onde:

*arquivo.diff*

Arquivo contendo as diferenças geradas pelo comando `diff`.

*opções*

-p [num]

Nível do diretório onde o `patch` será aplicado, se igual a 0, o `patch` assume que os arquivos que serão atualizados estão no diretório atual, se 1, assume que os arquivos que serão atualizados estão no diretório acima (..), se 2, 2 diretórios acima ...

-b

Cria cópias de segurança dos arquivos originais ao aplicar o patch.

-binary

Lê e grava arquivo usando modo binário.

-d [dir]

Muda para o diretório [dir] antes de aplicar o patch.

-E

Remove arquivos vazios após a aplicação do patch.

-n

Interpreta o arquivo de patch como um `.diff` normal.

-N

Não desfaz patches já aplicados.

-s

Não mostra mensagens de erro.

-u

Interpreta o patch em formato unificado.

As diferenças são aplicadas em arquivos originais gerados pelo comando `diff`. É importante entender os comandos `patch` e `diff` pois são comandos muito utilizados para desenvolvimento feito por equipes de pessoas.

Exemplo:

- `patch -p0<texto.diff` - Aplica as diferenças contidas no arquivo `texto.diff` nos arquivos originais.
- `patch -p0 texto.txt texto.diff` - Aplica as diferenças contidas no arquivo `texto.diff` nos arquivos originais. Faz a mesma coisa que o comando anterior.

---

## 10.36 whereis

Localiza o arquivo que contém uma página de manual. A pesquisa é feita usando-se os caminhos de páginas de manuais configuradas no sistema (normalmente o arquivo `/etc/manpath.config`).

`whereis [comando]`

Exemplo: `whereis ls`, `whereis cd`.

---

## 10.37 which

Mostra a localização de um arquivo executável no sistema. A pesquisa de arquivos executáveis é feita através do path do sistema. Para maiores detalhes, veja [path, Seção 7.2](#).

`which [comando]`

Exemplos: `which ls`, `which shutdown`, `which which`.

---

## 10.38 zforce

Renomeia extensão de arquivos para `.gz`. Este comando é útil quando fazemos downloads de arquivos compactados pelo `gzip` mas que não estão identificados pela extensão `.gz`.

`zforce [arquivos]`

Quando é usado o `zforce` verifica se o arquivo é um arquivo compactado pelo `gzip`, caso seja, é verificado se já tem a extensão `.gz`, caso não tiver, acrescenta a extensão.

---

## 10.39 gzexe

Cria arquivos compactados `gzip` auto-extrácteis. Este comando é usado para compactar arquivos executáveis que se auto-descompactam assim que são solicitados. É útil para sistemas ou unidades de disco que possuem pouco espaço disponível. Este comando deve somente ser usado para arquivos executáveis.

`gzexe [arquivo]`

Onde: *arquivo* é o arquivo executável que será compactado.

Quando `gzexe` é executado, uma cópia do arquivo original é gravada com o formato `nome_do_arquivo~`.

Exemplo: `gzexe /tmp/teste`.

---

## 10.40 znew

Recompacta arquivos do formato `compress` (`.Z`) para o formato `gzip` (`.gz`). Após a re-compactação, os arquivos de origem `.Z` são apagados.

`znew [opções] [arquivo]`

Onde:

*arquivo.Z*

Arquivo compactado pelo `compress` que será re-compactado para o `gzip`.

*opções*

`-f`

Substitui o arquivo `.gz` caso já exista.

`-t`

Teste os novos arquivos criados antes de apagar os arquivos `.Z`.

`-v`

Mostra o nome e porcentagem de compactação para cada arquivo processado.

`-9`

Usa a máxima compactação.

`-P`

Usa pipes durante a conversão para reduzir o espaço ocupado no disco. A data e hora do arquivo não é mantida caso esta opção seja usada.

`-K`

Mantém o arquivo `.Z` caso seja menor que o arquivo `.gz`.