

**Ficha 4 – Comandos**

**Diversos**

Comandos de uso diversos no sistema.

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

1. **date**

Permite ver/modificar a Data e Hora do Sistema. Precisa estar como usuário *root* para modificar a data e hora.

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).

Para ver a data actual 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

1. **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).

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

1. **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.

1. **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.

Exemplo: du -h, du -hc.

1. **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 - Oculta a linha de buffers.

-t - Mostra uma linha contendo o total.

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

1. **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.

**-E** - Avtiva o uso de expressões regulares.

**-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.

1. **head**

Mostra as linhas iniciais de um arquivo texto.

head [*opções*]

Onde:

**-c [numero]** - Mostra o [numero] de bytes do inicio do arquivo.

**-n [numero]** - Mostra o [numero] de linhas do inicio do arquivo. Caso não for especificado, o head mostra as 10 primeiras linhas.

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

1. **nl**

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

nl [*opções*] [*arquivo*]

Onde:

Opções:

**-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.

1. **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 efectua uma pausa e permite que 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.

1. **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 efectua uma pausa (semelhante ao more) e permite que 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

1. **sort**

Organiza as linhas de um arquivo texto ou da entrada padrão.

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*.

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.

1. **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.

**-f** - Mostra continuamente linhas adicionadas no final do arquivo.

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

1. **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.

1. **uptime**

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

uptime

1. **dmesg**

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

dmesg | less

1. **echo**

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

echo [*mensagem*]

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

1. **uname**

Retorna o nome e versão do kernel atual.

uname

1. **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.

1. **seq**

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

seq [*opções*] [*primeiro*] [*incremento*] [*último*]

Onde:

*primeiro*

* Número inicial da sequência.

*incremento*

* Número utilizado para avançar na sequência.

*último*

* Número final da sequência.

Opções:

**-f, --format=[formato]** - Formato de saída dos números da sequência. Utilize o estilo do printf para ponto flutuante (valor padrão: %g).

**-s, --separator=[string]** - Usa [string] para separar a sequência de números (valor padrão: \n).

-w, --equal-width - Insere zeros na frente dos números mantendo a sequência 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