

Comandos date e cal

São comandos que retornam informações de data do sistema.

Imprimir a data/hora atuais:
<code>\$ date</code>
Imprimir datas passadas:
<code>\$ date --date="yesterday"</code> <code>\$ date --date="1 week ago"</code>
Imprimir datas futuras:
<code>\$ date --date="next tue"</code>
Mostrar o calendário do mês com o dia atual grifado:
<code>\$ cal</code>
Mostrar o calendário do mês 5 (Maio) do ano corrente:
<code>\$ cal -m5</code>
Mostrar o calendário do mês 12 (Dezembro) do ano de 2012:
<code>\$ cal -m12 2012</code>

Comando wc

O comando `wc` é um contador de linhas, palavras e bytes.

A entrada do comando pode ser um arquivo ou um texto escrito pelo usuário.

Se a entrada do arquivo não é dada, ao apertar ENTER, o comando aguarda o usuário entrar com um texto. Ao terminar o texto, aperte CTRL+D.

<code>\$ echo Uma frase não muito longa. > arquivotxt1</code> <code>\$ echo Apenas mais uma linha. >> arquivotxt1</code>
Contando linhas, palavras e bytes do arquivo:
<code>\$ wc arquivotxt1</code>
Contando apenas as linhas do arquivo:
<code>\$ wc -l arquivotxt1</code>
Contando apenas as palavras do arquivo:
<code>\$ wc -w arquivotxt1</code>

Contando apenas os bytes do arquivo:

```
$ wc -c arquivotxt1
```

Comando sort

Organiza as linhas de um arquivo de acordo com o critério dado.

```
$ echo Vamos fazer um teste. > arqtxt  
$ echo Escreva algo no arquivo. >> arqtxt  
$ echo Agora escreva um número qualquer no início de  
uma linha. >> arqtxt  
$ echo 123 testando >> arqtxt  
$ echo Muito bem. Repita com outro número. >> arqtxt  
$ echo 42 apenas >> arqtxt  
$ clear  
$ cat arqtxt
```

Organizar linhas do arquivo em ordem alfabética (1-9, aA-zZ):

```
$ sort arqtxt
```

Organizar linhas do arquivo em ordem alfabética reversa (Zz-Aa, 1-9):

```
$ sort -r arqtxt
```

Organizar linhas do arquivo em ordem numérica (001-999, a-Z):

```
$ sort -n arqtxt
```

Comando tail e head

O comando *head*, sem parâmetro, imprime as primeiras 10 linhas de um arquivo.

Com parâmetro, podem-se especificar as primeiras *n* linhas do arquivo.

O comando *tail*, sem parâmetro, imprime as últimas 10 linhas de um arquivo.

Com parâmetro, podem-se especificar as últimas *n* linhas do arquivo.

```
$ cat arqtxt >> arqtxt2  
$ cat arqtxt2 >> arqtxt  
$ sort arqtxt > arq_alfabetica
```

Utilizando head e tail, sem parâmetro e com parâmetro -n

```
$ head arq_alfabetica
$ tail arq_alfabetica
$ head -n2 arq_alfabetica
$ tail -n2 arq_alfabetica
```

Comandos more/less

Ambos os comandos mostram o conteúdo de arquivos na tela.

Entretanto, o comando *more* só permite avançar linhas/telas, enquanto no comando *less* é possível avançar e retornar, realizar buscas no texto e até edições no arquivo visualizado.

Imprimir todo conteúdo na tela:

```
$ cat /etc/protocols
```

Mostrar a primeira tela e aguardar comando do usuário para avançar o conteúdo:

```
$ more /etc/protocols
```

Mostrar o arquivo tela por tela permitindo ao usuário avançar e retornar do início ao fim do arquivo (para sair pressione "q").

```
$ less /etc/protocols
```

Comandos lshw, vmstat, free

Mostra uma lista detalhada das configurações de hardware da máquina. É recomendável executar o comando como root.

Listar na tela as configurações detalhadas de hardware da máquina:

```
$ sudo lshw
```

Existem arquivos do sistema que guardam informações sobre o hardware. Há também outros comandos relacionados a informações apenas de memória.

Mostrar informações apenas sobre o processador:

```
$ more /proc/cpuinfo
```

Mostrar informações apenas sobre a memória:

```
$ more /proc/meminfo
```

Mostrar outras estatísticas sobre o uso da memória, processos e CPU:

```
$ vmstat  
$ vmstat -s
```

Mostrar quantidade de memória livre no sistema:

```
$ free -h
```

Comandos du e df

O comando du estima o uso de espaço de disco ocupado por um ou mais arquivos.

O comando df reporta o uso do espaço pelo sistema de arquivos.

Verificar o uso do espaço dos arquivos no diretório pessoal do usuário:

```
$ du ~/*  
$ du -h ~/*
```

Verificar o uso do espaço do diretório pessoal como um todo:

```
$ du -s ~/  
$ du -sh ~/
```

Mostrando o espaço ocupado pelo sistema de arquivos (mostra partições):

```
$ df  
$ df -h
```

Redirecionamento de saída e Pipe

Veja os slides da aula sobre redirecionamento.

<pre>\$ cd /</pre>
Redirecionamento de saída para um arquivo:
<pre>\$ ls -r > ~/lista1.txt \$ cd ~ \$ ls -r >> lista1.txt</pre>
Pipe (redirecionando a saída de um comando para entrada do outro):
<pre>\$ cat lista1.txt sort</pre>
Combinando pipe e redirecionamento de saída:
<pre>\$ cat lista1.txt sort > lista_organizada</pre>
Saída padrão (tela):
<pre>\$ echo Este texto irá para a tela do computador.</pre>
Redirecionando saída:
<pre>\$ echo Este texto irá para um arquivo. > redir_saida</pre>
Pipe da saída para outro comando:
<pre>\$ echo Este texto será entrada do próximo comando. wc</pre>

Mais um exemplo

```
$ wget http://wallpaper.pickywallpapers.com/1600x900/cat-napping-on-a-keyboard.jpg  
$ mv cat-napping-on-a-keyboard.jpg kitten1  
$ wc -l kitten1  
$ head -n500 kitten1 > kitten2  
$ tail -n600 kitten1 | head -n300 >> kitten2  
$ head -n1000 kitten1 | tail -n500 >> kitten2
```