**fuser**

Esse programa é utilizado para descobrir quais processos estão utilizando determinados arquivos ou soquetes.   
  
Sintaxe:   
  
**$ fuser [-a/-s/-c] [-4/-6] [-n [espaço]] [-k [-i] [-[sinal]] ] [-muvf]**   
  
Parâmetros:

* -a: Mostra todos os arquivos especificados na linha de comando. Por padrão, somente os arquivos que estão sendo usados por pelo menos um processo são mostrados;
* -s: Modo silencioso;
* -c: O mesmo que -m, usado para compatilidade com o padrão POSIX;
* -4: Procura apenas por sockets de IPv4;
* -6: Procura apenas por sockets de IPv6;
* -n [espaço]: Especifica o espaço de nomes a usar. Pode ser:
  + file: o modo padrão, procura por arquivos;
  + udp: procura por portas UDP;
  + tcp: Procura por portas TCP:
* -k: Matar os processos que estão acessando o arquivo;
* -i: Pergunta antes de matar o processo;
* -[sinal]: Informa qual o sinal deve ser usado para matar os processos. Só pode ser usado junto com o parâmetro -k;
* -m: Utilize quando estiver se referindo ao um sistema de arquivos montado ou a um dispositivo de blocos montado;
* -u: Exibe o nome do usuário que iniciou o processo que está utilizando o arquivo;
* -v: Modo detalhado, com diversas informações sobre os processos.

Exemplos de uso:   
  
Uma utilidade grande desse programa é a seguinte: suponhamos que você queria desmontar o disquete, e seja surpreendido por uma mensagem de erro, dizendo que o dispositivo de disquete está ocupado:   
  
**$ umount /dev/fd0**  
umount: /media/floppy0: device is busy  
umount: /media/floppy0: device is busy   
  
Se você não conseguir descobrir qual o processo que está "travando" o disquete, você pode fazer isso utilizando o fuser. Você precisará ser o administrador do sistema.   
  
**# fuser -m /dev/fd0**  
/dev/fd0: 2877   
  
Agora, você pode matar o processo:   
  
**# kill 2877**   
  
Ou, se quiser, pode matar o comando automaticamente com o fuser:   
  
**# fuser -m -k /dev/fd0**   
  
Se o processo não for encerrado, utilize o sinal -9 para destrui-lo:   
  
**# fuser -m -k -9 /dev/fd0**   
  
Repare que, nesse caso, utilizamos -m porque se trata de um dispositivo de blocos, /dev/fd0. Se fosse um arquivo comum, esse parâmetro não seria usado.   
  
Outro uso que pode ser usado por administrador de rede é procurar por processos que estejam utilizando determinadas portas. Por exemplo, pra saber qual o processo que está utilizando a porta TCP 445, utilizamos:   
  
**# fuser -n tcp 445**  
445/tcp: 2674

**df**

Mostra o espaço utilizado de cada partição.   
  
Sintaxe:   
  
**$ df [opção] [partição]**   
  
Opções:

* -a: Inclui na verificação os sistemas de arquivos com 0 blocos;
* -B [tamanho]: Usa blocos do tamanho especificado;
* -h: Exibe os tamanhos num formato de fácil compreensão (1K, 23M, 2G);
* -H: Igual ao -h, mas usa múltiplos de 1000 ao invés de 1024;
* -i: Mostra as informações dos inodes;
* -k: O mesmo que -B 1K;
* -l: Só exibe informações dos sistemas de arquivos locais;
* -P: Usa o formato de saída POSIX;
* --sync: Executa o sync antes de obter as informações;
* -t [tipo]: Só mostra informações dos sistemas de arquivos do tipo especificado;
* -T: Mostra qual o tipo do sistema de arquivos de cada partição exibida;
* -x [tipo]: Mostra todos os tipos de sistemas de arquivos exceto o tipo especificado aqui.

Exemplos de uso:   
  
Um exemplo simples:   
  
**$ df -h**  
Sist. Arq.              Tam     Usad    Disp    Uso%            Montado em  
/dev/hda2               4,9G    3,1G    1,9G    63%             /  
tmpfs                   59M     0       59M     0%              /dev/shm  
/dev                    4,9G    3,1G    1,9G    63%             /.dev  
none                    5,0M    744K    4,3M    15%             /dev  
/dev/hda1               32G     5,4G    27G     17%             /mnt/windows

**du**

Esse comando, sigla de disk usage (uso de disco), é usado para estimar o espaço em disco usado pelos arquivos. Em outras palavras, usamos o *du* para saber o tamanho dos arquivos.   
  
Sintaxe:   
  
**$ du [opções] [arquivo]** Opções:

* -a: Exibe o tamanho de todos os arquivos dentro dos subdiretórios do diretório especificado, e não somente dos arquivos imediatamente dentro do diretório;
* --aparent-size: Exibe o tamanho aparente, que pode ser maior ou menor que o tamanho real, dependendo de fatores com fragmentação, blocos indiretos e similares;
* -B [tamanho]: Exibe o tamanho dos arquivos em blocos do tamanho especificado;
* -b: Igual a -B 1;
* -c: Exibe uma linha com o tamanho total de todos os arquivos contabilizados;
* -D: Quando usado para links simbólicos, considera o local para onde o link aponta e não o link em si. Só considera arquivos;
* -h: Mostra os tamanhos de forma comprensível (2K, 32M, 1G);
* -H: O mesmo que -h, mas usa blocos de 1000 ao invés de 1024;
* -k: O mesmo que -B 1K;
* -L: O mesmo que -D, mas considera qualquer tipo de arquivo;
* -S: Não mostra o tamanho dos subdiretórios;
* -s: Mostra apenas o tamanho total de cada item (arquivo ou diretório);
* -x: Ignora arquivos que estejam em outro sistema de arquivos;
* -X [padrão]: Não contabiliza os arquivos que correspondam ao padrão informado;
* -m: O mesmo que -B 1M.

Suponha que você esteja no diretório /home/davidson/, que contenha os seguintes arquivos e diretórios:   
  
**$ ls**  
arquivo de teste  Desktop  exe  iso           src    tmp  
deb               doc      img  playlist.m3u  teste   
  
Se quisermos ver o tamanho do arquivo playlist.m3u:   
  
**$ du playlist.m3u**  
8       playlist.m3u   
  
O tamanho é exibido em KB. No caso, o arquivo tem 8 KB.   
  
Se quisermos ver o tamanho do diretório img/:   
  
**$ du img**  
0       img/davidson  
0       img/wallpapers  
592     img/diagramacao  
484     img/partition\_magic  
461     img/qtparted  
1132    img/programacao\_visual  
4018    img   
  
Veja que o tamanho em KB não é muito confortável. Usamos o parâmetro -h para que os valores sejam exibidos de uma forma compreensível:   
  
**$ du -h img**  
0       img/davidson  
0       img/wallpapers  
592K    img/diagramacao  
484K    img/partition\_magic  
461K    img/qtparted  
1,2M    img/programacao\_visual  
4,0M    img   
  
Podemos também querer que sejam contabilizados os arquivos dos subdiretórios img/davidson/ e img/wallpapers/. Nesse caso, usamo o parâmetros -a:   
  
**$ du -h -a img**  
673K    img/bozo.bmp  
44K     img/bozo.zip  
0       img/davidson  
0       img/wallpapers  
16K     img/davidson\_80x60.png  
52K     img/davidson\_cabeludo.jpg  
172K    img/diagramacao/linux\_magazine\_distribuicoes.pdf  
136K    img/diagramacao/linux\_magazine\_segurando\_desktop.pdf  
284K    img/diagramacao/linux\_magazine\_jogos.pdf  
592K    img/diagramacao  
28K     img/partition\_magic/Thumbs.db  
84K     img/partition\_magic/pqmagic001.jpg  
92K     img/partition\_magic/pqmagic002.jpg  
28K     img/partition\_magic/pqmagic003.jpg  
96K     img/partition\_magic/pqmagic004.jpg  
28K     img/partition\_magic/pqmagic005.jpg  
100K    img/partition\_magic/pqmagic006.jpg  
28K     img/partition\_magic/pqmagic007.jpg  
484K    img/partition\_magic  
565K    img/kacique01 .jpg  
16K     img/qtparted/snapshot01.png  
20K     img/qtparted/snapshot04.png  
16K     img/qtparted/snapshot10.png  
12K     img/qtparted/snapshot11.png  
12K     img/qtparted/snapshot12.png  
12K     img/qtparted/snapshot13.png  
32K     img/qtparted/qtparted001.jpg  
40K     img/qtparted/qtparted002.jpg  
32K     img/qtparted/qtparted003.jpg  
24K     img/qtparted/qtparted004.jpg  
28K     img/qtparted/qtparted005.jpg  
48K     img/qtparted/snapshot1.png  
16K     img/qtparted/snapshot2.png  
16K     img/qtparted/snapshot3.png  
52K     img/qtparted/snapshot4.png  
16K     img/qtparted/snapshot5.png  
16K     img/qtparted/snapshot6.png  
20K     img/qtparted/snapshot7.png  
16K     img/qtparted/snapshot8.png  
16K     img/qtparted/snapshot9.png  
461K    img/qtparted  
280K    img/programacao\_visual/ferrari001.png  
284K    img/programacao\_visual/ferrari002.png  
272K    img/programacao\_visual/ferrari003.png  
28K     img/programacao\_visual/Thumbs.db  
268K    img/programacao\_visual/tux.png  
1,2M    img/programacao\_visual  
4,0M    img   
  
Essa lista ficou bem extensa. Se quisermos exibir somente o tamanho total de cada diretório, retiramos o parâmetro -a, acrescentamos o parâmetro -s, e nos referenciamos a img/\* ao invés de img:   
  
**$ du -h -s img/\***  
673K    img/bozo.bmp  
44K     img/bozo.zip  
0       img/davidson  
16K     img/davidson\_80x60.png  
52K     img/davidson\_cabeludo.jpg  
592K    img/diagramacao  
565K    img/kacique01 .jpg  
484K    img/partition\_magic  
1,2M    img/programacao\_visual  
461K    img/qtparted  
0       img/wallpapers   
  
Dessa última forma, porém, o tamanho total do diretório não foi informado. Para isso, basta acrescentar o parâmetro -c:   
  
**$ du -h -s -c img/\***  
673K    img/bozo.bmp  
44K     img/bozo.zip  
0       img/davidson  
16K     img/davidson\_80x60.png  
52K     img/davidson\_cabeludo.jpg  
592K    img/diagramacao  
565K    img/kacique01 .jpg  
484K    img/partition\_magic  
1,2M    img/programacao\_visual  
461K    img/qtparted  
0       img/wallpapers  
4,0M    total

**free**

Comando muito utilizado para análise do desempenho do sistema, o free exibe informações sobre o uso de memória pelo computador.   
  
Sintaxe:   
  
**$ free [-b / -k / -m] [-o] [-s delay ] [-t]**   
  
Parâmetros:

* -b: Exibe as quantidades em bytes;
* -k: Exibe as quantidades em kilobytes;
* -m: Exibe as quantidades em megabytes;
* -o: Não exibe a linha -/+ buffers/cache;
* -s [tempo]: Especifica o intervalo de tempo, em segundos, entre as atualizações das informações. Se esse parâmetro não for usado, é exibida apenas uma informação. Se usado, vai exibindo as informações indefinidamente no intervalo de tempo definido, até que o usuário pressione CTRL + C;
* -t: Exibe uma linha com os valores totais;

Vejamos o uso mais comum do free:   
  
**$ free**  
             total       used       free     shared    buffers     cached  
Mem:        118880     116508       2372          0       2432      38216  
-/+ buffers/cache:      75860      43020  
Swap:       514040     127604     386436   
  
Aqui temos as seguintes informações:

* Memória RAM total: 118880 KB
* Memória RAM usada: 116508 KB
* Memória RAM livre: 43020 KB
* Informações em buffer: 2432 KB
* Informações em cache: 38216 KB
* Memória swap total: 514040 KB
* Memória swap usada: 127604 KB
* Memória swap livre: 386436 KB
* Buffers/Cache usados: 75860 KB
* Buffers/Cache livres: 43020 KB

A exibição em KB pode não ser muito confortável. Se quiser que os tamanhos sejam exibidos em MB, utilize o parâmetro -m:   
  
**$ free -m**  
             total       used       free     shared    buffers     cached  
Mem:           116        114          1          0          2         40  
-/+ buffers/cache:         71         44  
Swap:          501        124        377

**time**

Esse é um comando muito útil para medir o desempenho do sistema. Com o time podemos medir o tempo, em segundos, necessários para executar um processo ou programa.   
  
Sintaxe:   
  
**$ time [opções] [comando]**   
  
Opções:

* -o [arquivo]: Grava as estatísticas coletadas durante a execução do comando para o arquivo especificado, para análise posterior;
* -a: Quando usado junto com o parâmetro -o, não apaga o conteúdo do arquivo ao gravar as estatísticas;
* -v: Exibe os detalhes da execução do comando.

Assim, suponha que você tenha um script de backup, e queira saber quanto tempo ele demora para ser executado:   
  
**$ time backup**  
real    0m39.054s  
user    0m28.560s  
sys     0m0.689s   
  
O campo real nos mostra quanto tempo o processo demorou para ser executado. No caso, 39,054 segundos. O campo user informa quanto tempo a CPU gastou processando apenas os dados do comando. Nesse exemplo, 28,560 segundos. O campo sys informa qual o intervalo média de espera da CPU entre cada ciclo de processamento dos dados do comando. Aqui, o tempo foi de 0,689 segundos.

**uptime**

Mostra o período em que o sistema permaneceu em processamento desde que foi ligado.   
  
Uso:   
  
**$ uptime**   
  
O uptime não possui parâmetros. Basta digitar uptime no terminal. Veja um exemplo:   
  
**$ uptime**  
16:40:18 up  7:48,  3 users,  load average: 0.43, 0.29, 0.26   
  
Aqui, vemos que o computador esteve ligado por 16:40 horas, mas só esteve em processamento durante 7:48 horas. Durante esse período, 3 usuários acessaram o sistema, e a carga média do sistema, que vai de 0 a 1, foi de 0,43 no último 1 minuto, 0,29 nos últimos 5 minutos e 0,26 nos últimos 15 minutos.

**dmesg**

Esse comando é utilizado para ver as mensagens de inicialização do sistema. Útil para analisar eventuais mensagens de erro exibidas devido a qualquer problema que esteja ocorrendo durante a inicialização.   
  
Sintaxe:   
  
Como as mensagens são muito extensas, utilize o more ou o less para visualizar as informações:   
  
**$ dmesg | more  
$ dmesg | less**

**echo**

Exibe mensagens na tela. Esse comando é utilizado principalmente para a construção de scripts de sistema.   
  
Sintaxe:   
  
**$ echo [opções] [cadeia de caracteres]**   
  
Opções:

* -n: Não insere uma nova linha;
* -e: Ativa a interpretação de caracteres de escape, listados a seguir:
* \NNN: Código ASCII octal do caracter;
* \\: Barra invertida (\);
* \a: Alerta sonoro (beep);
* \b: Backspace;
* \c: Não exibe a linha de final de arquivo;
* \f: Alimentação de formulário (form feed);
* \n: Nova linha;
* \r: Retorno de carro (carriage return);
* \t: Tabulação horizontal;
* \v: Tabulação vertical.

Exemplos de uso:   
  
**$ echo "Projeto Bozolinux"**  
Projeto Bozolinux   
  
**$ echo -e "Primeira linha\nSegunda linha\n\tTerceira linha com tabulação horizontal"**  
Primeira linha  
Segunda linha  
        Terceira linha com tabulação horizontal   
  
Com o uso de direcionadores, pode-se usar o echo para inserir texto em arquivos, com a seguinte sintaxe:   
  
**$ echo [mensagem] [> / >>] [arquivo]**   
  
Onde > apaga o conteúdo o arquivo, se existir, e >> adiciona o texto no final do arquivo.   
  
Exemplos de uso:   
  
**$ echo -n > /etc/modules**   
  
Apaga o conteúdo do arquivo /etc/modules.   
  
**$ echo -e "192.168.0.35\tdavidson.bozolinux.org\tdavidson" >> /etc/hosts**   
  
Adiciona a seguinte linha no arquivo /etc/hosts:   
  
192.168.0.35       davidson.bozolinux.org davidson

**su**

Esse comando é utilizado para um usuário assumir os privilégios de outro usuário do sistema.   
  
Sintaxe:   
  
**$ su [opções] [usuário]**   
  
A digitar o comando é necessário digitar a senha do usuário selecionado. Se nenhum usuário for especificado, o sistema entende que o usuário quer assumir os privilégios de administrador do sistema (root).   
  
Uma opção muito útil é o parâmetro -c, que permite executar um comando específico com os privilégios do usuário selecionado, e ao término da execução perder esses privilégios.   
  
Exemplos de uso:   
  
Um exemplo é a compilação de um pacote. O comando make install, que faz a instalação propriamente dita do pacote, só pode ser executado pelo usuário root. Assim, temos que fazer o seguinte:   
  
**$ su**  
password:  
**# make install  
# exit**   
  
Ao invés disso, podemos simplesmente digitar:   
  
**$ su -c "make install"**  
password:   
  
O comando será executado e, ao seu término, perdemos os privilégios de administrador de sistema.

**sync**

Esse comando é utilizado para gravar os dados armazenados em cache nos locais apropriados. O uso mais comum é para gravar os dados em unidades de disco removível, geralmente disquete.   
  
Uso:   
  
O *sync* não possui parâmetros. Para usá-lo, execute:   
  
**$ sync**

**reboot**

Reinicia o computador. Por padrão, somente o root pode executar esse comando.   
  
Sintaxe:   
  
**# reboot**

**shutdown**

Usado para desligar o sistema. Por padrão, somente o root pode executar o shutdown.   
  
Sintaxe:   
  
**# shutdown [opções] [hora] [mensagem de alerta]**   
  
Opções:

* -t [tempo]: Espera o tempo especificado (em segundos) entre matar os processos e mudar de nível de execução;
* -k: Não desliga o sistema, apenas envia a mensagem de alerta a todos os usuários que estão conectados;
* -r: Reinicia o sistema após o desligamento;
* -h: Desliga o computador;
* -f: Não roda o utilitário fsck no caso de reiniciar o sistema;
* -F: Força o uso do fsck no reinício do sistema;
* -c: Cancela um processo de desligamento que esteja sendo executado no momento.

Em [hora] você pode especificar o horário exato para o sistema desligar, como 12:34, por exemplo, ou então utilizar +[n], para desligar o sistema daqui a [n] minutos. Para desligar o sistema imediatamente, use +0 ou a palavra now.   
  
A mensagem de alerta será enviada a todos os usuários conectados ao sistema, para que eles tenham tempo de salvar seus arquivos e se desconectarem.   
  
Exemplos de uso:   
  
O uso mais comum do shutdown é para desligar o sistema o computador imediatamente:   
  
**# shutdown -h now**