

Comandos

Manipulação de diretórios

ls

Lista os arquivos de um diretório.

ls [opções] [caminho/arquivo] [caminho1/arquivo1] ...

Onde:

caminho/arquivo

Diretório/arquivo que será listado.

caminho1/arquivo1

Outro Diretório/arquivo que será listado. Podem ser feitas várias listagens de uma só vez.

opções

-a, --all

Lista todos os arquivos (inclusive os ocultos) de um diretório.

-A, --almost-all

Lista todos os arquivos (inclusive os ocultos) de um diretório, exceto o diretório atual e o de nível anterior.

-B, --ignore-backups

Não lista arquivos que terminam com ~ (Backup).

--color=PARAM

Mostra os arquivos em cores diferentes conforme o tipo de arquivo. PARAM pode ser:

never - Nunca lista em cores (mesma coisa de não usar o parâmetro --color).

always - Sempre lista em cores conforme o tipo de arquivo.

auto - Somente colore a listagem se estiver em um terminal.

-d, --directory

Lista os nomes dos diretórios em vez do conteúdo.

-f

Não classifica a listagem.

-F

Insere um caracter após arquivos executáveis (*), diretórios (/), soquete (=), link simbólico (@) e pipe (|). Seu uso é útil para identificar, de forma fácil, tipos de arquivos nas listagens de diretórios.

-G, --no-group

Oculto a coluna de grupo do arquivo.

-h, --human-readable

Mostra o tamanho dos arquivos em Kbytes, Mbytes, Gbytes.

-H

Faz o mesmo que -h, mas usa unidades de 1000 em vez de 1024 para especificar Kbytes, Mbytes, Gbytes.

-l

Usa o formato longo para listagem de arquivos. Lista as permissões, data de modificação, donos, grupos, etc.

-n

Usa a identificação de usuário e grupo numérica em vez dos nomes.

-L, --dereference

Lista o arquivo original e não o link referente ao arquivo.

-o

Usa a listagem longa sem os donos dos arquivos (mesma coisa que -lG).

-p

Mesma coisa que -F, mas não inclui o símbolo '*' em arquivos executáveis. Essa opção é típica de sistemas Linux.

-R

Lista diretórios e subdiretórios recursivamente.

--full-time

Lista data e hora completa.

Classificação da listagem

A listagem pode ser classificada usando-se as seguintes opções:

-f

Não classifica e usa -au para listar os arquivos.

-r

Inverte a ordem de classificação.

-c

Classifica pela data de alteração.

-X
Classifica pela extensão.

-U
Não classifica. Lista os arquivos na ordem do diretório.

Uma listagem feita com o comando `ls -la` normalmente é mostrada da seguinte maneira:

```
-rwxr-xr-- 1 gleydson user 8192 nov 4 16:00 teste
```

Abaixo as explicações de cada parte:

`-rwxr-xr--`
São as permissões de acesso ao arquivo teste. A primeira letra (da esquerda) identifica o tipo do arquivo. Se tiver um `d` é um diretório, se tiver um `-` é um arquivo normal.

As permissões de acesso é explicada em detalhes em Permissões de acesso a arquivos e diretórios, Capítulo 13.

`1`
Se for um diretório, mostra a quantidade de subdiretórios existentes dentro dele. Caso seja um arquivo, será `1`.

`gleydson`
Nome do dono do arquivo teste.

`user`
Nome do grupo a que o arquivo teste pertence.

`8192`
Tamanho do arquivo (em bytes).

`nov`
Mês da criação/ última modificação do arquivo.

`4`
Dia em que o arquivo foi criado.

`16:00`
Hora em que o arquivo foi criado/modificado. Se o arquivo foi criado há mais de um ano, em seu lugar é mostrado o ano da criação do arquivo.

`teste`
Nome do arquivo.

Exemplos do uso do comando `ls`:
`ls` - Lista os arquivos do diretório atual.

`ls /bin /sbin` - Lista os arquivos do diretório `/bin` e `/sbin`.
`ls -la /bin` - Listagem completa (vertical) dos arquivos do diretório `/bin`, inclusive os ocultos.

cd

Entra em um diretório. É necessário que se tenha a permissão de execução para entrar no diretório.

`cd [diretório]`

Onde:

`diretório` - diretório que deseja entrar.

Exemplos:

Usando `cd` sem parâmetros ou `cd ~`, você retornará ao seu diretório de usuário (diretório home).

`cd /`, retornará ao diretório raiz.

`cd -`, retornará ao diretório anteriormente acessado.

`cd ..`, sobe um diretório.

`cd ../[diretório]`, sobe um diretório e entra imediatamente no próximo (por exemplo, quando você está em `/usr/sbin`, digita `cd ../bin`, logo, o comando `cd` retorna um diretório (`/usr`) e entra imediatamente no diretório `bin` (`/usr/bin`).

pwd

Mostra o nome e caminho do diretório atual.

Você pode usar o comando `pwd` para verificar em qual diretório se encontra (caso seu aviso de comandos não mostre isso).

mkdir

Cria um diretório no sistema. Um diretório é usado para armazenar arquivos de um determinado tipo. O diretório pode ser entendido como uma pasta onde você guarda seus papéis (arquivos). Como uma pessoa organizada, você utilizará uma pasta para guardar cada tipo de documento, da mesma forma você poderá criar um diretório vendas para guardar seus arquivos relacionados com vendas naquele local.

```
mkdir [opções] [caminho/diretório] [caminho1/diretório1]
```

Onde:

`caminho`

Caminho onde o diretório será criado.

`diretório`

Nome do diretório que será criado.

opções

--verbose

Mostra uma mensagem para cada diretório criado. As mensagens de erro serão mostradas mesmo que esta opção não seja usada.

Para criar um novo diretório, você deve ter permissão de gravação. Por exemplo, para criar um diretório em /tmp com o nome de teste que será usado para gravar arquivos de teste, você deve usar o comando "mkdir /tmp/teste".

Pode ser criado mais de um diretório com um único comando (mkdir /tmp/teste /tmp/teste1 /tmp/teste2).

rmdir

Remove um diretório do sistema. Este comando faz exatamente o contrário do mkdir. O diretório a ser removido deve estar vazio e você deve ter permissão de gravação para removê-lo.

rmdir [caminho/diretório] [caminho1/diretório1]

Onde:

caminho

Caminho do diretório que será removido.

diretório

Nome do diretório que será removido.

É necessário que esteja num nível acima do diretório(s) que será(ão) removido(s). Para remover diretórios que contenham arquivos, use o comando rm com a opção -r (para maiores detalhes, veja rm, Seção 9.3).

Por exemplo, para remover o diretório /tmp/teste, você deve estar no diretório tmp e executar o comando rmdir teste.

Manipulação de arquivos

cat

Mostra o conteúdo de um arquivo binário ou texto.

cat [opções] [diretório/arquivo] [diretório1/arquivo1] diretório/arquivo

Localização do arquivo que deseja visualizar o conteúdo.

opções

-n, --number

Mostra o número das linhas enquanto o conteúdo do arquivo é mostrado.

-s, --squeeze-blank

Não mostra mais que uma linha em branco entre um parágrafo e outro.

-

Lê a entrada padrão.

O comando cat trabalha com arquivos texto. Use o comando zcat para ver diretamente arquivos compactados com gzip.

Exemplo: cat /usr/doc/copyright/GPL

tac

Mostra o conteúdo de um arquivo binário ou texto (como o cat), mas em ordem inversa.

tac [opções] [diretório/arquivo] [diretório1/arquivo1] diretório/arquivo

Localização do arquivo que deseja visualizar o conteúdo.

opções

-s [string]

Usa o [string] como separador de registros.

-

Lê a entrada padrão.

Exemplo: tac /usr/doc/copyright/GPL.

rm

Apaga arquivos. Também pode ser usado para apagar diretórios e sub-diretórios vazios ou que contenham arquivos.

rm[opções] [caminho] [arquivo/diretório] [caminho1] [arquivo1/diretório1]

Onde:

caminho

Localização do arquivo que deseja apagar. Se omitido, assume que o arquivo esteja no diretório atual.

arquivo/diretório

Arquivo que será apagado.

opções

-i, --interactive

Pergunta antes de remover. Esta é ativada por padrão.

-v, --verbose

Mostra os arquivos na medida em que são removidos.

-r, --recursive

Usado para remover arquivos em subdiretórios. Esta opção também pode ser usada para remover sub-diretórios.

-f, --force

Remove os arquivos sem perguntar.

-- arquivo

Remove arquivos/diretórios que contêm caracteres especiais. O separador "--" funciona com todos os comandos do shell e permite que os caracteres especiais, como "*", "?", "-", etc., sejam interpretados como caracteres comuns.

Use com atenção o comando rm. Uma vez que os arquivos e diretórios forem apagados, eles não poderão ser mais recuperados.

Exemplos:

rm teste.txt - Apaga o arquivo teste.txt no diretório atual.

rm *.txt - Apaga todos os arquivos do diretório atual que terminam com .txt.

rm *.txt teste.novo - Apaga todos os arquivos do diretório atual que terminam com .txt e também o arquivo teste.novo.

rm -rf /tmp/teste/* - Apaga todos os arquivos e subdiretórios do diretório /tmp/teste, mas mantém o sub-diretório /tmp/teste.

rm -rf /tmp/teste - Apaga todos os arquivos e subdiretórios do diretório /tmp/teste, inclusive /tmp/teste.

rm -f -- --arquivo-- - Remove o arquivo de nome --arquivo--.

cp

Copia arquivos.

cp [opções] [origem] [destino]

Onde:

origem

Arquivo que será copiado. Podem ser especificados mais de um arquivo para ser copiado usando "Curingas" (veja Curingas, Seção 2.12).

destino

O caminho ou nome de arquivo onde será copiado. Se o destino for um diretório, os arquivos de origem serão copiados para dentro do diretório.

opções

i, --interactive

Pergunta antes de substituir um arquivo existente.

-f, --force

Não pergunta. Substitui todos os arquivos caso já existam.

tam.

-r

Copia arquivos dos diretórios e subdiretórios da origem para o destino. É recomendável usar -R em vez de -r.

-R, --recursive

Copia arquivos e subdiretórios (como a opção -r) e também os arquivos especiais FIFO e dispositivos.

-v, --verbose

Mostra os arquivos enquanto estão sendo copiados.

-s, --symbolic-link

Cria link simbólico em vez de copiar.

-l, --link

Faz o link no destino em vez de copiar os arquivos.

-p, --preserve

Preserva atributos do arquivo se for possível.

-u, --update

Copia somente se o arquivo de origem é mais novo que o arquivo de destino ou quando o arquivo de destino não existe.

-x

Não copia arquivos que estejam localizados em um sistema de arquivos diferente de onde a cópia iniciou.

O comando cp copia arquivos da ORIGEM para o DESTINO. Ambos, origem e destino, terão o mesmo conteúdo após a cópia.

Exemplos:

cp teste.txt teste1.txt

Copia o arquivo teste.txt para teste1.txt.

cp teste.txt /tmp

Copia o arquivo teste.txt para dentro do diretório /tmp.

cp * /tmp

Copia todos os arquivos do diretório atual para /tmp.

cp /bin/* .

Copia todos os arquivos do diretório /bin para o diretório em que nos encontramos no momento.

cp -R /bin /tmp

Copia o diretório /bin e todos os arquivos/subdiretórios existentes para o diretório /tmp.

cp -R /bin/* /tmp

Copia todos os arquivos do diretório /bin (exceto o diretório /bin) e todos os arquivos/subdiretórios existentes dentro dele para /tmp.

cp -R /bin /tmp

Copia todos os arquivos e o diretório /bin para /tmp.

mv

Move ou renomeia arquivos e diretórios. O processo é semelhante ao do comando cp, mas o arquivo de origem é apagado após o término da cópia.

mv [opções] [origem] [destino]

Onde:

origem

Arquivo/diretório de origem.

destino

Local para onde será movido ou novo nome do arquivo/diretório.

opções

-f, --force

Substitui o arquivo de destino sem perguntar.

-i, --interactive

Pergunta antes de substituir. É o padrão.

-v, --verbose

Mostra os arquivos que estão sendo movidos.

-u, --update

Move somente arquivos antigos ou novos arquivos.

O comando mv copia um arquivo da ORIGEM para o DESTINO (semelhante ao cp), mas, após a cópia, o arquivo de ORIGEM é apagado.

Exemplos:

```
mv teste.txt teste1.txt
```

Muda o nome do arquivo teste.txt para teste1.txt.

```
mv teste.txt /tmp
```

Move o arquivo teste.txt para /tmp. Lembre-se de que o arquivo de origem é apagado após ser movido.

```
mv teste.txt teste.new (supondo que teste.new já exista)
```

Copia o arquivo teste.txt por cima de teste.new e apaga teste.txt após terminar a cópia.

Comandos Diversos

clear

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

```
clear
```

date

Permite ver/modificar a Data e Hora do Sistema. Você precisa estar como usuário 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 usuários) .

```
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.

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 em vez de blocos.

-H

Idêntico a -h, mas usa 1000 em vez 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 sistema de arquivos do tipo tipo.

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

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 ao 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 em vez 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.

du

Mostra o espaço ocupado por arquivos e subdiretórios 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) em vez 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 subdiretórios.

-x

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

Exemplo: du -h, du -hc.

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 seus subdiretórios.

opções/expressão

-name [expressão]

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

-depth

Processa os subdiretórios primeiro antes de processar os arquivos do diretório principal.

-maxdepth [num]

Faz a procura até [num] subdiretórios dentro do diretório em 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 arquivos que foram acessados [num] minutos atrás. Caso sejam antecidos por "-", procura arquivos que foram acessados entre [num] minutos atrás até agora.

-atime [num]

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

-gid [num]

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

-group [nome]

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

-uid [num]

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

-user [nome]

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

-inum [num]

Procura arquivos que estejam localizados no inodo [num].

-links [num]

Procura arquivos que possuam [num] links como referência.

-mmin [num]

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

-mtime [num]

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

-nouser

Procura arquivos que não correspondam à identificação do usuário atual.

-nogroup

Procura arquivos que não correspondam à identificação do grupo do usuário atual.

-perm [modo]

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

-used [num]

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

-size [num]

Procura arquivos que tiverem o tamanho [num]. [num] pode ser antecido por "+" 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.

-type [tipo]

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

b - bloco

c - caracter

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 subdiretórios, um arquivo/diretório chamado grep.

`find / -name grep -maxdepth 3` – Procura, no diretório raiz e subdiretórios até o 3o. nível, um arquivo/diretório chamado `grep`.

`find . -size +1000k` – Procura, no diretório atual e subdiretórios, um arquivo com tamanho maior que 1000 kbytes (1Mbyte).

`find / -mmin 10` – Procura, no diretório raiz e subdiretórios, um arquivo que foi modificado há 10 minutos atrás.

`find / -links 4` – Procura, no diretório raiz e subdiretórios, todos os arquivos que possuam 4 links como referência.

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] segundo.

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

grep

Procura 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 deverá identificá-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 está 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`.

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 seja especificado, o `head` mostrará as 10 primeiras linhas.

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

nl

Mostra o número de linhas junto ao 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 contêm 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.

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.

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

sort

Organiza as linhas de um arquivo texto ou da entrada padrão. A organização é feita por linhas as quais são divididas em campos que correspondem à ordem na qual as palavras aparecem na linha, sendo separadas por um delimitador (normalmente um espaço).

sort [opções] [arquivo]

Onde:

arquivo

É o nome do arquivo que será organizado. Caso não seja 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 esteja organizando um campo que contenha números, estes 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 seja usada com o sort, ele a organizará como uma listagem alfabética (que começa de a até z e de 0 até 9)

10

100

50

-c

Verifica se o arquivo já está organizado. Caso não esteja, 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 da linha que será usado na organização. O(s) campo(s) usado(s) para organização estará(ão) 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 seja 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 redirecionada à 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 caractere ":" como delimitador de campos em vez do espaço. Neste caso, o que estiver após ":" será considerado o próximo campo.

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.

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.

touch

Muda a data e hora em 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 Minutos (MM), Dias (DD), Horas (hh), minutos (mm) e, opcionalmente, o ANO e segundos para modificação do(s) arquivo(s) em vez 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 em vez da hora atual.

Exemplos:

touch teste - Cria o arquivo teste caso ele não exista.

touch -t 10011230 teste - Altera a data e hora do arquivo para 01/10 e 12:30.

touch -t 120112301999.30 teste - Altera a 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.

uptime

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

uptime

dmesg

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

dmesg|less

mesg

Permite ou não os recebimentos de requisições de talk de outros usuários.

mesg [y/n]

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

Digite mesg para saber se você pode ou não receber "talks" de outros usuários. Caso a resposta seja "n", você poderá enviar um talk para alguém, mas o seu sistema se recusará a 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 usuários, veja o comando talk.

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 ter sido mostrada.

su

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

su [usuário]

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

Será pedida a senha do superusuário para autenticação. Digite exit quando desejar retornar a identificação de usuário anterior.

sync

Grava os dados do cache de disco na memória RAM para todos os discos rígidos e flexíveis do sistema. O cache é um mecanismo de aceleração que permite o armazenamento de um arquivo na memória em vez 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.)

uname

Retorna o nome e a versão do kernel atual.

uname

reboot

Reinicia o computador.

shutdown

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

shutdown [opções] [hora] [mensagem]

hora

Momento em que o computador será desligado. Você

pode usar HH:MM para definir a hora e o 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 usuários 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 usuários, 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 o fato aos usuários.

-a

Permite que os nomes dos usuários contidos no arquivo /etc/shutdown.allow possam utilizar o shutdown para reiniciar/desligar o sistema. Deve ser colocado um nome de usuário por linha. O limite máximo de usuários 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 usuários.

-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 usuários do sistema, alertando sobre o desligamento durante os 15

minutos restantes e, assim, permite que finalizem suas tarefas. Após isso, o shutdown muda o nível de execução através do comando init para 0 (desligamento), 1 (modo monousuário), 6 (reinicialização). É recomendável o uso do 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 usuários 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. Isso permite que somente os usuários autorizados (ou o root) possam reinicializar o sistema.

Exemplos:

"shutdown -h now" - Desliga 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 monousuário (init 1) às 19:00, enviando a mensagem A manutenção do servidor será iniciada às 19:00 a todos os usuários 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 usuários 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 e espera 30 segundos após o sinal de término para enviar o sinal KILL a todos os programas abertos.

wc

Conta o número de palavras, bytes e linhas em um arquivo ou entrada padrão. Se as opções estiverem omitidas, o wc mostrará 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. Sendo modificada a posição das opções, não ocorre alteração da ordem na qual 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.

seq

Imprime uma seqüência de números que começam em [primeiro] e terminam 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] estiverem 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% 0 10, seq

-s", " 0 10.

chattr

Modifica atributos de arquivos/diretórios. Não confunda atributos de arquivos 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 à 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 usuário. 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` evite 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, removidos ou renomeados. Até mesmo o usuário `root` não poderá modificar estes arquivos.

Em diretórios, faz com que arquivos não possam ser adicionados ou apagados. Somente o usuário `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.

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 desejam listar os atributos. Podem ser usados curingas.

opções

-a

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

-d

Lista os atributos de diretórios em vez 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`

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 [numero] de caracteres no arquivo. É semelhante à 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] em vez de tabulações.

-s

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

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

cmp

Compara dois arquivos de qualquer tipo (binário ou texto). Os dois arquivos especificados serão comparados 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 ao 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.

-s

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

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

Exemplo: cmp teste.txt teste1.txt.

dirname

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

dirname [diretório/arquivo]

dirname /usr/bin/dirname, dirname /tmp/.*

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 contenha 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 atualizá-lo (através do patch) em vez de copiar a nova versão. Este é um sistema de atualização muito usado na atualização dos códigos 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 existem 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 subdiretórios 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 existentes 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 subdiretórios linux-2.2.13 e linux-2.2.14 e grava as diferenças entre eles no arquivo patch-2.2.14.diff.

pr

Pagina 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 em vez 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] em vez do nome do arquivo no cabeçalho.

-l [num]

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

-m

Imprime vários arquivos em paralelo, sendo 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.`

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 estarão no diretório atual; se 1, assume que os arquivos que serão atualizados estarão 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.

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`.

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`.

which [comando]

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

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 se o arquivo é um arquivo compactado pelo `gzip`. Caso seja, é verificado se já tem a extensão `.gz`; caso não seja, acrescenta a extensão.

gzexe

Cria arquivos compactados `gzip` auto-extractíveis. Este comando é usado para compactar arquivos executáveis que se autodescompactam assim que são solicitados. É útil para sistemas ou unidades de disco que possuam 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`.

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 recompactado para o `gzip`.

opções

-f

Substitui o arquivo `.gz` caso já exista.

-t

Testa 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`.