Guia de Referência UNIX

Eduardo Marcel Maçan

July 16th, 2022

Contents

Prefácio

Eu escrevi este guia de UNIX há 25 anos, em 1997. Era um mundo bem diferente, embora a Internet já estivesse presente no país ainda não havia se popularizado e estava restrita a grandes centros urbanos. O panorama de tecnologia se dividia entre diferentes fornecedores de UNIX nas grandes corporações, como Sun, IBM, HP, Digital. A Microsoft era um novo entrante neste mercado, o Linux já havia colocado um pé firme nos servidores da grande maioria das startups, mas ainda estava longe da ubiquidade que conquistou hoje.

E naquele cenário de múltiplos fornecedores e múltiplos "sabores" de UNIX que a ideia deste guia se tornou atrativa. Cada UNIX possuia mais ou menos os mesmos comandos, mas eles divergiam entre estilos, opções de comando e funcionalidade. Que tal encontrar o núcleo comum de comandos e opções de linha de comando que permitiria a um usuário uma maior portabilidade ao escrever shell scripts e navegar entre UNIX de diferentes fornecedores?

Este guia foi escrito por um estudante universitário nos seus vinte-e-poucos anos, que teve o privilégio de ter entrado em contato com a Internet, diversos sabores de UNIX comerciais e o Linux em 1993, na UNICAMP. Esta foi a base da minha carreira até aqui e tem potencial para continuar sendo ainda por muito tempo. Ao ter lido os manuais de diversos unix para selecionar os comandos que estavam presentes em todos e depoisem cada comando selecionando as opções de linha de comando comuns, posso afirmar com orgulho que o conteúdo deste guia sobreviveu muito bem aos 25 anos entre sua escrita e esta "versão comemorativa" de um quarto de século.

Agradeço ao Rubens Prates e à Novatec Editora pela oportunidade de tê-lo escrito e pelo literal "`Sim!`" para reeditar seu conteúdo em um novo formato de livre distribuição para seu aniversário.

Este é um release inicial ainda em fase "alfa" que passará por constantes revisões. Novidades e novas versões serão publicadas em https://unix.macan.eng.br

Eduardo Maçan

Arquivos e Diretórios

awk

Descrição Linguagem para processamento de padrões. Para uma descrição detalhada da linguagem AWK e seus comandos, consulte os manuais on-line do sistema (**man awk**).

Sintaxe

```
awk [ -f *arq* ][ -F *c* ] ['prog'] [arquivo ...]
```

Parâmetro	Descrição
-	Toma a entrada de stdin.
-f *arq*	Usa o conjunto de padrões definidos no arquivo *arq* para processar sua entrada.
-F *c*	Muda o caractere separador de campos.
arquivo	Especifica um arquivo de entrada. Um nome de arquivo no formato **string=valor** será tratado como uma atribuição de valor à variável string e será executado no momento em que seria processado caso fosse um arquivo válido.
'prog'	Sequência de declarações *padrão-ação* separadas por ponto e vírgula (;). Uma declaração *padrão-ação* tem o seguinte formato: **padrão { ação }**

Utilização:

Comparam-se os padrões especificados em prog a cada linha dos arquivos de entrada, executando uma ação associada sempre que houver correspondência. Um padrão vazio corresponde a todas as linhas da entrada, a ação vazia imprime a linha que correspondeu ao padrão.

Padrões:

Um padrão é uma combinação booleana de expressões relacionais ou regulares, com sintaxe similar à da linguagem C. Dois padrões especiais, **BEGIN** e **END**, são usados para tomar ações antes do processamento da primeira linha de entrada e após o processamento da última linha, respectivamente.

Ações:

Uma ação é uma sequência de comandos separados por ponto e virgula (;) TAB ou newiline.

Comandos

```
if (expr) comando [ else comando ]
while ( expr ) comando
do comando while (expressão )
for ( expr; expr ; expr) comando
for ( variavel in lista ) comando
break
continue
{ [ comando ...] }
variável=expr
```

```
print [expressão ...]
printf formato [, expressão [, ...]]
next
exit [n]
```

Expressões

Expressões em AWK têm sintaxe semelhante à da linguagem C, sendo válidos os operadores relacionais e de atribuição definidos nesta linguagem, a saber:

Operador	Significado
+	Adição.
-	Subtração.
** * **	Multiplicação.
/	Divisão.
%	Módulo.
^	Exponenciação.
++	Incremento.
–	Decremento.
+=	Soma de valor à variável.
-=	Subtração de valor da variável.
** *= **	Multiplicação de variável por valor.
/=	Divisão de variável por valor.
%=	Módulo de variável por valor.
^=	Exponenciação de variável por valor.
>	Maior que.
>=	Maior ou iguala.
<	Menor que.
<=	Menor ou igual a.
!=	Diferente de.
?:	Condicional.

Funções

Função	Definição
exp(x)	Retorna o valor de e".
log(x)	Retorna o logaritmo natural de x.
sqrt(x)	Retorna a raiz quadrada de x.
index(s,t)	Retorna a posição da primeira ocorrência da string s dentro da string t, ou zero se não houver ocorrências.
int(s)	Converte a strings para um valor inteiro.
linha de	Se s não for especificada, usa toda a entrada.
lenght(s)	Retorna o comprimento da string s, ou da linha de entrada se s não for especificada.
match(s,er)	Retorna a posição da primeira ocorrência da expressão regular er dentro da string s, ou zero se não houver ocorrências.
split(s,a,fs)	Divide a strings em vários elementos de um vetor **a**[**n**] e retorna **n**. A divisão é feita usando a expressão regular fs ou o separador de campos FS se fs não for especificado.

Função	Definição
sprintf(fmt,exp,...)	Formata as expressões de acordo com o formato especificado por fmt e retorna a string resultante (equivalente à função sprintf da linguagem C).
substr(s,m,n)	Retorna uma substring des comncaracteres começando na posição m.
getline	Carrega a próxima linha de entrada em $0, retornando 1 em caso de sucesso, O para fim de arquivo e -1 para erro.

Variaveis e constantes:

Variáveis podem ser elementos de vetor (representados por **x**[**y**], onde **x** é um vetor e **y** o índice) ou campos do arquivo de entrada (representados por **$n**, onde **n** representa o número do campo). Os vetores podem ser indexados por strings, as quais aparecem, obrigatoriamente, entre aspas duplas (“).

Variáveis internas:

Variável	Significado
FILENAME	Nome do arquivo de entrada corrente.
NF	Número de campos no registro (linha de entrada) corrente.
NR	Número do registro corrente.
OFS	Separador de campos da saida (default espaço).
ORS	Separador de linhas de saida (default newline).
RS	Separador de linhas da entrada (default newline).
FS	Separador de campos (default espaço e TAB).

Exemplos:

```
# Mostra os usuarios locais do sistema e seus
# respectivos nomes completos.
awk -F: '{print $1 " -> " $5)' /etc/passwd
```

```
 # Programa - AWK Eduardo M. Macan 1996
 # Calcula as médias dos alunos e a média da
 # classe, a partir  de um arquivo texto contendo
 # um nome e duas notas em cada linha
 # (por exemplo: Eduardo 5.7 7.5).

BEGIN {
    printf("Aluno\t\tMedia Final\n")
}

S1 {
    ++alunos;
    media=($2 + $3) / 2;
    printf ("%s\t\t%.2f\n",$1,media) ;
    total+=media
```

```
}

END {
    printf("Media da turma : %.2f\n",total/alunos)
}
#fim do programa
```

basename

Descrição Retira o path e (opcionalmente) a extensão dos nomes de arquivos.

Sintaxe

```
basename <arquivo> [arquivo ...] [sufixo]
```

Exemplos

```
basename /etc/passwd
basename /home/staff/macan/images/*.gif .gif
```

cat

Concatena e mostra arquivos.

Sintaxe: cat [-etuv] [arquivo ...]

Parâmetro	Descrição
-e	Similar a -v, mas imprime um cifrão ($) ao final de cada linha. Ignorado em alguns sistemas se não usado com -v.
-t	Similar a -v, mas imprime caracteres de tabulação (tabs) como ^I. Ignorado em alguns sistemas se não usado com -v.
-u	Processa a saída caractere a caractere, em vez de utilizar o buffer.
-v	Imprime caracteres especiais de forma visivel (tabs e newlines sao processados).

Exemplos

```
# Copia stdin para o arquivo texto.txt
cat > texto.txt

# Cria o arquivo capitulo 1 como resultado da
# concatenação de texto1 e texto2.
cat texto1 texto2 > capitulo1

# Cria o arquivo arq3, como resultado da
# concatenação de arq1, stdin e arq2.
cat arq1 - arg2 > arq3
```

cd

Muda o diretório de trabalho (função interna da shell). Se o diretório não for especificado, muda para o diretório HOME do usuário.

Sintaxe

```
cd [diretório]
```

Exemplos

```
# Muda o diretório de trabalho para o
# diretório personal.
cd personal

# Muda o diretório de trabalho para o
# diretório HOME do usuário leonardo.
cd ~leonardo

# Muda o diretório de trabalho para o
# diretório /home/staff/macan.
cd /home/staff/macan
```

chgrp

Muda o grupo de um arquivo.

Sintaxe:

```
chgrp [-R] <grupo> <arquivo> [arquivo ...]
```

Parâmetro	Descrição
-R	Muda ogrupo de todos os arquivos e subdiretórios abaixo do diretório especificado.

Obs: O grupo de um arquivo só pode ser mudado pelo dono; os grupos válidos são aqueles a que o dono pertence. Apenas o superusuário (root) pode mudar o grupo de arquivos dos outros usuários.

chmod

Altera permissões de acesso a arquivos.

Sintaxe:

```
chmod [-R] <modo> <arquivo> [arquivo ...]
```

Parâmetro	Descrição
-R	Muda o modo de acesso de todos os arquivos e subdiretórios abaixo do diretório especificado. modo Pode assumir modo simbólico ou absoluto, como descrito a seguir:

Modo simbólico:

O modo simbólico é uma lista de expressões da forma **[identificador ...]** **operando** [**valor**] separada por vírgulas.

Identificador	Descrição
u	Permissões para o dono do arquivo.
g	Permissões para o grupo do arquivo.
o	Permissões para outros grupos.
a	Todos os anteriores (all). Default se o identificador for omitido.

Operando	Descrição
+	Adiciona permissão às permissões existentes no arquivo.
-	Retira permissão das permissões existentes no arquivo.
=	Assinala explicitamente uma permissão (zerando as outras).

Valor	Descrição
r	Permissão para leitura.
w	Permissão para escrita.
x	Permissão para execução.
X	Permissão para execução se o arquivo for um diretório ou já houver permissão.
s	Bit setgid se atribuido a g, setuid se atribuído au.
t	Bit *sticky*.

Modo absoluto:

As permissões neste modo são representadas por um número octal de quatro dígitos, da forma **EUGO**.

Dígito	Significado
E	Atributos especiais.
U	Permissões para o dono do arquivo.
G	Permissões para o grupo do arquivo.
O	Permissões para outros grupos.
Para os	dígitos UGO temos a seguinte interpretação:
Valor O	ctal Significado

0 Nenhuma permissão. 1 Permissão de execução. 2 Permissão de escrita. 3 Permissão de execução e escrita. 4 Permissão de leitura. 5 Permissão de execução e leitura. 6 Permissão de leitura e escrita. 7 Permissão de leitura, escrita e execução.

Para o dígito E temos a seguinte interpretação:

Valor Octal	Significado
0	Nenhum atributo especial ligado.
1	Bit sticky ligado.
2	Bit setgid ligado.
3	Bits sticky e setgid ligados.
4	Bit setuid ligado.
5	Bits sticky e setuid ligados.
6	Bits setuid e setgid ligados.
7	Bits sticky, setuid e setgid ligados.
Atributos es	peciais:

Bit	Significado
setuid	O arquivo é executado como se fosse invocado pelo proprietário; nãofaz sentido para diretórios.
setgid	O arquivo é executado sob seu grupo, mesmo que o usuário invocador não participe dele; todo arquivo criado em um diretório **setgid** é criado com o mesmo grupo do diretório.
sticky	Um arquivo criado sob um diretório com o bit sticky ligado pode ser apagado apenas por seu proprietário. A interpretação do sticky bit pode variar entre sistemas Unix.

Obs: No modo absoluto, os zeros à esquerda são ignorados, no modo simbólico só faz sentido omitir valor utilizando o operador = para zerar os bits de permissão. Apenas o superusuário (root) pode alterar os atributos de um arquivo de outro usuário.

Exemplos:

```
# Adiciona permissão de execução pelo dono
# ao arquivo meu. script
chmod u+x meu script

# Adiciona permissão de leitura e escrita
# para o dono e para o grupo de usuários
# ao qual o arquivo meu script pertence.
chmod ug+rw meu script

# Adiciona permissão de execução e escrita
# pelo dono ao mesmo tempo que retira a
# permissão de escrita do grupo e deixa outros
# grupos apenas lerem o conteúdo do arquivo.
chmod u+wx,g-w,o0=r meu script

# Confere permissão de leitura, escrita
# e execução ao dono do arquivo,
# leitura e execução ao grupo do arquivo e
# nenhuma permissão aos demais grupos.
chmod 750 helloworld

# Equivalente ao anterior.
chmod 0750 helloworld

# Liga o sticky bit do diretório e dá
# permissão de leitura, escrita e
# execução a todos os usuários do sistema.
chmod 1777 -macan/PUB/

# Confere permissão de leitura,
# escrita e execução ao dono e de
# execução para grupo e outros
# ao diretório HOME.
chmod 711 ~
```

cksum

Efetua o cálculo de CRC e mostra o tamanho do arquivo (em bytes).

Sintaxe: **cksum** [*arquivo ...*]

Obs:

CRC significa **Código de Redundância Ciclica**. É um código numérico calculado a partir do conteúdo do arquivo, usado para verificar a integridade de arquivos ou dados em transmissão.

cmp

Compara dois arquivos binarios.

Sintaxe: cmp [-ls] <arquivo1> <arquivo2>

Parâmetro	Descrição
-l	Para cada diferença, mostra o número do byte inicial e os bytes que diferiram no formato octal
-s	Modo silencioso; retorna apenas o status de saída (O=arquivos iguais, 1=arquivos diferentes, >1 erro).

comm

Compara arquivos texto, linha a linha. Os dois arquivos precisam estar ordenados por linha.

Sintaxe: comm [-123] <arquivo1> <arquivo2>

Parâmetro	Descrição
-1	Não mostra as linhas que aparecem apenas em arquivo.
-2	Não mostra as linhas que aparecem apenas em arquivo2.
-3	Não mostra as linhas comuns aos dois arquivos.

compress

Comprime arquivos aplicando a codificação adaptativa de Lempel-Ziv. O arquivo comprimido terá a extensão .Z.

Sintaxe:

```
compress [-cvf] [-b bits] <arquivo> [arquivo ...]
```

Parâmetro	Descrição
-b	bits Define o tamanho (em bits) das substrings usadas no processo de compressão (9a 16, default=16).
-c	Comprime para a saída padrão (stdout), sem alterar arquivos.
-f	Comprime sem considerar se o tamanho do arquivo vai ser realmente reduzido. Sobrescreve arquivos com o mesmo nome.
-v	Mostra estatísticas de compressão na saída de

Parâmetro	Descrição
	erro padrao (stderr) .

Exemplo:

```
# Comprime todos os arquivos com extensão txt.
compress -v *.txt
```

cpio

Armazena ou recupera arquivos guardados em arquivos de armazenamento.

Sintaxe:

```
cpio -o [acvB] [C val]
cpio -i [cdfmrstuvS6B] [padrão ...]
cpio -p [adimuv] <diretorio>
```

Parâmetro	Descrição
-6	Processa arquivos no formato antigo de cpio. Usado somente com -i.
-B	Usa blocos de 512 bytes.
-C *val*	Usa blocos de val * 512 bytes.
-S	Troca a posição das metades de cada palavra (4 bytes) nos arquivos.
-a	Mantém o tempo de cesso (access time) dos arquivos fonte.
-c	Escreve e lê informações de cabeçalho no formato ASCII. Um arquivo criado com -c precisa ser lido com -c.
-d	Cria diretórios conforme a necessidade.
-f	Só copia arquivos que não coincidam com os padrões especificados.
-i	Lê da entrada padrao um arquivo criado por **cpio -o** e extrai os arquivos armazenados.
-l	Usa links em vez de copiar arquivos, sempre que possivel. Deve ser usado com -p.
-m	Mantém a data de modificação dos arquivos. Não funciona com diretórios.
-o	Lê nomes de arquivos da entrada padrão (stdin) e os copia para a saída padrão, junto com a informação necessária para recuperá-los com **cpio -i**.
-p	Lê nomes de arquivos da entrada padrão (stdin) e copia os arquivos para o diretório especificado.
-r	Renomeia arquivos interativamente. Um ponto (.) mantém o mesmo nome do arquivo, enter faz cpio ignorar o arquivo.
-s	Troca a posição dos bytes dos arquivos. Deve ser usada com -i.
-t	Cria uma tabela de conteúdo. Não copia arquivos.
-u	Copia incondicionalmente, substituindo arquivos homônimos que sejam mais novos.
-V	Lista o nome dos arquivos.

Exemplos:

```
# Copia todos os arquivos listados pelo
# arquivo  texto arquivos.txt
# para o arquivo de armazenamento arq.cpio.
cpio -o < arquivos.txt > arq.cpio

# Extrai todos os arquivos armazenados
# em arq.cpio.
cpio -i < arg.cpio
```

cp

Copia um ou vários arquivos. Quando vários arquivos estão sendo copiados, destino deve se referir a um diretório.

Sintaxe:

```
cp [-ipr] <arquivo> [arquivo ...] <destino>
```

Parâmetro	Descrição
-i	Solicita confirmação antes de copiar cada arquivo.
-p	Mantém na cópia as datas de modificação e as permissões do arquivo original.
-r	Copia todos os arquivos e subdiretórios abaixo do diretório especificado. Neste caso, destino deve se referir a um diretório.

Exemplos:

```
# Copia todo o diretório html do usuário leonardo
# para o diretório /www.
cp -r ~leonardo/html/ /www
```

cut

Seleciona trechos de cada linha de arquivos texto.

Sintaxe:

```
cut -flista [-dc] [-s] [arquivo ...]
cut -clista [arquivo ...]
```

Parâmetro	Descrição
-	Indica explicitamente que a entrada deve ser tomada de stdin.
-c *lista*	Seleciona caracteres de cada linha do arquivo. *lista* é uma lista de valores inteiros ou intervalos separados por virgulas.
-f *lista*	Seleciona campos em cada linha do arquivo. Veja também o parâmetro -c.
-d*c*	Especifica o delimitador de campos como sendo o caractere *c* em vez do caractere de tabulação.
-s	Ignora linhas sem delimitadores de campos.

Exemplos:

```
# Mostra **userid** e nome completo dos usuários locais
cut -d: -f1,5 /etc/passwd

echo "Eduardo Marcel Macan"|cut -c-7,15-

echo "blues&rock&bolero"|cut -d\& -f1,2
```

dd

Copia, converte e formata arquivos.

Sintaxe:

```
dd [opção=valor...]
```

Parâmetro	Descrição
bs=n	Configura ibs=n bytes e obs=n bytes.
cbs=n	Converte n bytes de cada vez.
count=n	Copia apenas n blocos de entrada.
ibs=n	Lê n bytes de entrada de cada vez.
if=arq	Lê o arquivo arq em vez de stdin.
obs=n	Escreve n bytes no arquivo de saída de cada vez.
of=arq	Escreve no arquivo arq em vez de escrever em stdout e não trunca o arquivo.
seek=n	Busca o n-ésimo bloco de tamanho obs do arquivo de entrada antes de começar a cópia.
skip=n	Ignora os n primeiros blocos de tamanho igual a ibs na entrada.
conv=chave[,chave...]	Converte o arquivo segundo a lista de chaves separadas por vírgulas. Chaves podem ser: **ascii** Converte de EBCDIC para ASCII. **ebcdic** Converte de ASCII para EBCDIC. **ibm** Converte de ASCII para EBCDIC usando tabela alternativa. **block** Preenche registros terminados por newline com espaços até que estes atinjam otamanho especificado por cbs. **unblock** Substitui espaços finais em registros de tamanho fixo cbs por newline. **lcase** Converte para minúsculas. **ucase** Converte para maiúsculas. **swab** Troca pares de bytes de posição. **noerror** Ignora a ocorrência de erros. **notrunc** Não trunca o arquivo de saída. **sync** Preenche cada bloco de entrada com NULs até atingir o tamanho especificado por ibs. **n** pode apresentar um multiplicador, que pode ser m (megabytes), k (kilobytes), b (blocks - 512 bytes) ou c (characters - bytes).

Exemplos:

```
# Converte arquivo de ebcdic para ascii minúsculas.
dd if=ebcdic.txt of=ascii.txt conv=ascii, lcase

# Converte a saída do comando ls para maiúsculas.
ls | dd conv=ucase
```

df

Mostra dados de ocupação dos sistemas de arquivo especificados oudo sistema de arquivo onde residem os arquivos passados como parâmetro. Chamado sem parâmetros, df mostra a ocupação de todos os sistemas de arquivos montados (mounted).

Sintaxe:

```
df[-k] [arquivo...] [filesystem...]
```

Parâmetro	Descrição
-k	Mostra a ocupação em kilobytes em vez de blocos de disco (algumas versões de df).

Exemplo:

```
df /tmp / /var
```

diff

Mostra as diferenças entre dois arquivos

```
diff [-bcefir] [-D str] [-S arq] <arquivo1> <arquivo2>
```

Parâmetro	Descrição
-b	Ignora sequências de espaços em branco e caracteres de tabulação (TAB).
-c	Produz uma listagem das diferengas usando linhas de contexto.
-D str	Produz uma versão mista dos dois arquivos, usando controles do pré-processador de C e str como macro de controle.
-e	Cria uma sequência de comandos do editor ed que permite recriararquivo2a partir doarquivo1.
-f	Produz uma saída similar à opção -e, mas de interpretação mais simples.
-	Considera letras maiúsculas e minúsculas equivalentes.
-S arq	Inicia a comparação de dois diretórios a partir do arquivo arg.
-r	Também processa os subdiretórios quando comparando diretórios.

Exemplo:

```
diff -f /etc/passwd.old /etc/passwd
```

du

Mostra quantos blocos de disco são ocupados por arquivos ou diretórios. Se um argumento for um diretório du, mostrará a ocupação de seus subdiretórios.

Sintaxe:

```
du [-ask] [arquivo ...]
```

Parâmetro	Descrição
-a	Mostra uma entrada para cada arquivo que não seja um diretório.
-s	Mostra apenas o valortotal de ocupação de cada diretório especificado.
-k	Mostra a ocupação em kilobytes em vez de blocos de disco (algumas versões de du).

Exemplo:

```
du -s -macan
```

file

Determina o tipo dos arquivos examinando seu conteúdo.

Sintaxe:

```
file [-cLz] [-m arq] [-f arq] [arquivo ...]
```

Parâmetro	Descrição
-c	Procura erros no arquivo de números mágicos (magic numbers) antes de usá-lo.
-f *arq*	Determina otipodos arquivos listados no arquivo *arq*.
-L	Segue links simbólicos.
-m *arq*	Usa o arquivo especificado por *arq* como arquivo de números mágicos, em vez de /etc/magic.
-z	Determina o tipo dos arquivos que foram comprimidos.

Exemplo:

```
# Investiga o tipo do arquivo postscriptfile.ps.
file postscriptfile.ps
```

find

Percorre os diretórios (e seus subdiretórios) especificados mostrando os arquivos com as características desejadas.

Sintaxe:

```
find <diretório> [diretório ...] [expressão]
```

Parâmetro	Descrição
expressão	As seguintes primitivas podem ser combinadas para definir expressão.
-atime n	Verdadeiro se o arquivo tiver sido acessado há n dias.
-ctime n	Verdadeiro se o arquivo tiver sido mudado há n dias.
-exec c;	Verdadeiro se o comando c retornar zero após sua execução.
-fstype x	Verdadeiro se o arquivo residir em um sistema de arquivos do tipo x.

Parâmetro	Descrição
-group x	Verdadeiro se o arquivo pertencer ao grupo x.
-inum n	Verdadeiro se o número do i-node do arquivo for igual a n.
-links n	Verdadeiro se houver n links para o arquivo.
-mtime n	Verdadeiro se o arquivo tiver sido modificado há n dias.
-ok c;	Similar a -exec, porém pedindo confirmação prévia.
-name x	Verdadeiro se x coincidir com o nome do arquivo corrente.
-newer x	Verdadeiro se o arquivo tiver sido modificado mais recentemente do que o arquivo x.
-nouser	Verdadeiro se o arquivo corrente não pertencer a nenhum usuário listado no arquivo /etc/passwd.
-nogroup	Verdadeiro se o arquivo corrente não pertencer a nenhum grupo listado no arquivo/etc/group.
-perm [-]modo	Verdadeiro se coincidirem as permissões do arquivo e modo. Se precedido por -, apenas os bits ligados de modo serão comparados.
-print	Sempre verdadeiro. Imprime o nome do arquivo corrente.
-prune	Não entra em subdiretórios.
-size n[c]	Verdadeiro se o tamanho do arquivo for de n blocos. Se seguido pela letra c, o tamanho será interpretado em bytes.
-type t	Verdadeiro se o tipo do arquivo corresponder a t, onde t é um entre: **b** Block device. **c** Character device. **d** Diretório. **l** Link. **p** Arquivos comuns (plain files). **f** FIFO. **s** Socket.
-user x	Verdadeiro se o arquivo pertencer ao usuário x.
(exp)	Verdadeiro se a expressão entre parênteses for verdadeira. Os parênteses serão interpretados pela shell se não forem precedidos por.
operadores	Pode-se combinar diversas expressões usando os seguintes operadores (emordem decrescente de precedéncia)
!exp	Operador de negação (o caractere. ! é interpretado pela shell e deve ser precedido por).
exp -and exp	Operador 'E' lógico. Retorna verdadeiro se ambas as expressões forem verdadeiras.
exp -a exp	Algumas versões de find usam -a no

Parâmetro	Descrição
	lugar de -and.
exp -or exp	Operador 'OU' lógico. Retorna verdadeiro se pelo menos uma das duas expressões for verdadeira.
exp -o exp	Algumas versões de find usam -o no lugar de -or.

Exemplos:

```
# Lista todos os arquivos no diretório HOME
# do usuário macan que não têm extensão .c.
find ~macan \! -name "*.c" -print

# Lista todos os arquivos no diretório HOME
# com permissão de execução pelo proprietário.
find ~ -perm -100 -print
```

grep

Encontra ocorrências de um padrão dentro de arquivos.

Sintaxe:

```
grep [-cilnsv] <padrão> [arquivo ...]
```

Parâmetro	Descrição
-c	Mostra apenas o número de linhas que contêm o padrão.
-i	Não diferencia letras maiúsculas de minúsculas.
-l	Mostra apenas os nomes dos arquivos que contêm o padrão.
-n	Mostra cada linha que contém o padrao precedida por seu numero dentro do arquivo.
-s	Não mostra mensagens de erro produzidas por tentativas de acesso a arquivos.
-v	Mostra apenas as linhas que nao contém o padrão especificado.

Obs: Recomenda-se que *padrão* apareça entre aspas simples ('), pois alguns caracteres (notadamente $,*,[, ^, | ,(,) e) tem significado especial para a shell e podem ser interpretados erroneamente.

Exemplos:

```
# Mostra o número de usuários locais que
# se chamam "Mauro".
grep -ci 'Mauro' /etc/passwd

# Mostra os arquivos no diretório html
# que contêm a string Maçan (em HTML).
grep -l 'Ma&ccedil;an' html/*
```

head

Mostra as primeiras linhas de arquivos.

Sintaxe:

```
head [-c n | -n n |-b 1 ] [-f] [arquivo ...]
```

Parâmetro	Descrição
-b n	Mostra os últimos n blocos do arquivo.
-c n	Mostra os últimos n caracteres do arquivo.
-f	Faz com quehead não pare detentar ler, mesmo encontrando o fim do arquivo, assumindo que o mesmo está crescendo.
-n n	Mostra as últimas n linhas do arquivo.
-#[b|c|l]	Onde # é o numero de linhas (default), caracteres ou blocos a serem mostrados. Disponivel em algumas versões.

Exemplo:

```
# Mostra as primeiras linhas do arquivo mydecls.h..
head mydecls.h
```

ln

Cria links para arquivos ou diretórios. In cria tanto links simbólicos (soft links) como diretos (hard links). Default=links diretos.

Sintaxe:

```
ln [-fs] <destino> <nome>
ln [-fs] <arquivo> [arquivo ...] <diretório>
```

Parâmetro	Descrição
-f	Força a criação do link, ignorando acessibilidade ou existência de arquivos.
-s	Cria um link simbólico (soft link).

ls

Lista o conteúdo de diretórios.

Sintaxe:

```
ls [-aAcCdfFiLqrRstu1] [arquivo ...]
```

Parâmetro	Descrição
-a	Inclui entradas do diretório cujos nomes comecem com ".", normalmente omitidas.
-A	Lista todas as entradas, exceto "." e "..", esta opção é default para o superusuário (root).
-c	Usa a data de modificação do arquivo para a ordenação ou impressão.
-C	Formata a saída em múltiplas colunas; default quando a saída é o terminal.
-d	Trata diretórios como arquivos comuns e não segue links simbólicos.
-f	Não ordena a saída.
-F	Diferencia os tipos de arquivos concatenando

Parâmetro	Descrição
	caracteres a seus nomes: / Diretórios. * Arquivos executáveis. @ Links simbólicos. = Sockets. | Pipes.
-i	Imprime o número do i-node de cada arquivo.
-l	Lista usando o formato longo.
-L	Se o argumento for um link simbólico, lista o arquivo ou diretório referenciado.
-q	Força a impressão de caracteres não gráficos nos nomes dos arquivos, como, por exemplo, pontos de interrogação (?).
-r	Reverte a ordenação para obter ordem alfabética inversa, ou arquivos mais antigos primeiro.
-R	Lista também o conteúdo dos subdiretórios abaixo do diretório especificado.
-s	Mostra o número de blocos de disco usados pelos arquivos.
-t	Ordena os resultados cronologicamente.
-u	Usa a data do ultimo acesso ao arquivo paraa ordenação ou impressão.
-1	Mostra um elemento por linha de saida.

Obs: O formato longo mostra informações sobre o tipo de arquivo e suas permissões de acesso como uma string de dez caracteres, onde o primeiro identifica o tipo dos arquivos da seguinte forma:

Caractere	Tipo
b	Block device.
e	Character device.
d	Diretório.
l	Link.
p	FIFO (named pipe).
s	Socket da familia AF_UNIX.
-	Arquivo comum (plain file).

Os outros nove caracteres referem-se cada um a um bit de permissão de acesso, os três primeiros às permissões do proprietário, os três seguintes às permissões do grupo do arquivo e os últimos às permissões de acesso para outros grupos. Podem assumir os seguintes valores:

Caractere	Permissão
r	permissão de leitura.
w	permissão de escrita.
x	permissão de execução.
-	bit de permissão desligado.

Os caracteres de permissão de execução do proprietário e do grupo serão mostrados como “s” se os bits de setuid e setgid do arquivo

estiverem ligados, respectivamente. O sticky bit ligado será mostrado como um "t" no lugar do caractere de permissão de execução para outros grupos. As letras "S"e"T" maiúsculas serão mostradas no lugar de "s"e"t" quando o bit de permissão de execução correspondente não estiver ligado.

Exemplos:

```
ls -la /etc/passwd ls /etc
ls -lad /etc de sla-fero
```

mkdir

Cria diretórios.

Sintaxe:

```
mkdir [-p] <diretório> [diretório ...]
```

Parâmetro	Descrição
-p	Cria os diretórios intemediários do path se necessário.

Exemplos:

```
# Cria o diretório livro e componentes
# intermediários do path se necessário.
mkdir -p ./pub/docs/livro

# Cria vários diretórios.
mkdir dirl ../dir2 /usr/local/src/dir3
```

mkfifo

Cria FIFOs (named pipes). Para maiores informações sobre named pipes, consulte os manuais on-line do sistema e a leitura recomendada ao final deste guia.

Sintaxe:

```
mkfifo [nome ...]
```

more

Mostra arquivos texto página a página.

Sintaxe:

```
more [-csu] [arquivo ...]
```

Parâmetro	Descrição
-c	Escreve a partir do topo da tela em vez de rola-la.
-s	Substitui múltiplas linhas em branco consecutivas por apenas uma.
-u	Trata caracteres de retrocesso (backspace) e sequéncias CR-LF de maneira especial.

Exemplo:

```
# Útil para investigação de diretórios com muitas entradas.
ls -la |more

more /etc/passwd
```

mv

Move ou renomeia arquivos e diretórios. Se múltiplos arquivos forem especificados, destino deverá necessariamente ser um diretório.

Sintaxe:

```
mv [-fi] <arquivo> [arquivo ...] <destino>
```

Parâmetro	Descrição
-f	Não pede confirmação antes de sobrescrever um arquivo ja existente.
-i	Pede confirmação antes de mover um arquivo que irá sobrescrever outro.

Exemplos:

```
mv livro livro.old

mv linux/docs/*HOWTO ../linux/
```

paste

Mostra lado a lado o conteúdo de arquivos.

Sintaxe:

```
paste [-d lista] [-s] [arquivo ...]
```

Parâmetro	Descrição
-d lista	Usa os caracteres da string lista como separadores em vez de TAB. Quando o último caractere for utilizado, o primeiro será utilizado i novamente. Os seguintes caracteres especiais podem ser usados como separadores: **\n** Quebra de linha (newline). **\t** Tabulação (TAB). \\ Barra invertida (backslash). **\0** String vazia (não o caractere NUL).
-s	Concatena todas as linhas de cada arquivo em separado, na ordem em que foram especificados na linha de comando.
-	Usa stdin como entrada.

Obs: Caso um arquivo seja menor do que os outros, ele será tratado como se tivesse um número infinito de linhas vazias após sua última linha.

patch

Atualiza arquivos com as modificações indicadas pela saída do comando diff.

Sintaxe:

```
patch [arqorig]
```

Obs: **patch** lerá o arquivo de diferenças gerado por diff da entrada padrão e o aplicará ao arquivo arqorig, gerando assim uma versão idêntica do arquivo contra o qual arq1 foi comparado por diff. Caso *argorig* não seja especificado, patch perguntará ao usuário a qual arquivo devem ser aplicadas as alterações.

pwd

Mostra o diretório corrente.

Sintaxe:

```
pwd
```

rmdir

Remove diretorios vazios.

Sintaxe:

```
rmdir <diretorio> [diretorio ...]
```

rm

Remove arquivos.

Sintaxe:

```
rm [-firR] <arquivo> [arquivo ...]
```

Parâmetro	Descrição
-f	Efetua a eliminação sem solicitar confirmação.
-i	Solicita confirmação antes de remover.
-r ou -R	Causa a remoção de todos os arquivos e subdiretórios abaixo do diretório especificado (inclusive o próprio).

Exemplos:

```
rm -rf -macan/src/compilador

rm -1 *.g1f *. jpg
```

sort

Ordena linhas de arquivos.

Sintaxe:

```
sort [+p1 [-p2]] [-k p1[,p2]] [-bedfimMnru]
    [-o arq] [-te] [Tdir] [arquivo ...]
```

Parâmetro	Descrição
+p1 [-p2]	Especifica a posição da chave de ordenação nas linhas do arquivo, entre p1 e p2. Conta a partir de 0.
-b	Ignora brancos (espaços e newlines) antes das chaves de ordenação.

Parâmetro	Descrição
-c	Verifica se os arquivos ja estao ordenados; nao ordena.
-d	Considera apenas letras e digitos como parte das chaves de ordenacao.
-f	Converte letras minúsculas em maiúsculas antes de efetuar as comparações.
-i	Considera parte das chaves apenas os caracteres visíveis num terminal.
-k p1[,p2]	Similar a +p1 [-p2], porém campos e caracteres contam a partir de 1.
-m	Mescla (merge) arquivos já ordenados; não efetua ordenação.
-M	Ordena usando uma string como, por exemplo, nome de mês; implica -b.
-n	Ordena usando o valor numérico da string; implica -b.
-o arg	Escreve sua saída no arquivo arq em vez de stdout.
-r	Reverte a ordenação.
-tc	Use c como separador de campos. Default=espaços e newlines.
-T *dir*	Armazena arquivos temporários no diretório dir.
-u	Elimina linhas com campos repetidos. Em conjunto com -m.

Obs: p1 e p2 têm a sintaxe f.c, onde f representa um campo e c um caractere contado a partir do início do campo (para +p) ou do fim do campo (para -p).

Exemplos:

```
# Ordena o conteúdo de arquivo, usando o
# segundo campo (palavra, no sentido de texto)
# como chave de ordenação.
sort -k 2,2 arquivo

# Ordena o arquivo arq e armazena
# a saida em arg.ord.
sort arq -ro arq.ord

# Mostra o arquivo local de senhas ordenado
# por user id (o terceiro campo separado
# por dois pontos).
cat /etc/passwd|sort -nt: -k 3,3
```

split

Divide um arquivo em várias partes.

Sintaxe:

```
split [-I nl] [-a suf] [arquivo [prefixo]]
split -b n[k|m] [-a suf] [arquivo [prefixo]
```

Parâmetro	Descrição
-a suf	Especifica o numero de letras a serem usadas para compor o sufixo; default=2.
-b n[k|m]	Especifica o número de bytes de cada parte a ser quebrada; os modificadores k (Kilobytes) e m (Megabytes) são válidos.
-l nl	Especifica o número de linhas a serem usadas para quebrar um arquivo. Só faz sentido com arquivos texto; default= 1000.
prefixo	Especifica o prefixo a ser usado no nome de cada parte do arquivo original; default=x.

Obs: As opções -b e -| são mutuamente exclusivas.

Exemplo:

```
# Quebra o arquivo browser.tgz em vários blocos
# de aprox. 1.4Mb.
split -b1400k browser.tgz brow
```

strings

Encontra texto em arquivo binários.

Sintaxe:

```
strings [-ao] [arquivo ...]
```

Parâmetro	Descrição
-a	Procura strings ao longo de toda a extensão do arquivo e não somente nos segmentos de texto e dados dos arquivos objeto.
-o	Imprime o offset de cada string dentro do arquivo.

Exemplo:

```
strings /usr/lib/libc.a
```

tail

Mostra as linhas finais de um arquivo. Algumas versões de tail aceitam no máximo um arquivo como entrada.

Sintaxe:

```
tail[-cn|-nn|-bn][-f] [arquivo ...]

tail [-#[b|c|I] ] [-f] [arquivo ...]
```

Parâmetro	Descrição
-b n	Mostra os últimos n blocos do arquivo.
-c n	Mostra os últimos n caracteres do arquivo.
-f	Faz com que tail não pare de ler o arquivo mesmo encontrando seu fim. Útil quando o arquivo está aumentando constantemente.
-n n	Mostra as últimas n linhas do arquivo.

Parâmetro	Descrição
-#[b|c|l]	Onde # é o número de linhas (default), caracteres ou blocos a serem mostrados. Presente apenas em algumas versões do comando tail.

Exemplo:

```
# Lista as últimas mensagens do sistema.
tail /var/adm/messages
```

tar

Armazena vários arquivos e diretórios dentro de um único arquivo ou dispositivo.

Sintaxe:

```
tar [txcrumphvw] [f arq] [-b bIk] [-C dir] [arquivo ...]
```

Parâmetro	Descrição
-C dir	Muda o diretório de trabalho para o diretório dir. Pode aparecer mais de uma vez entre nomes de arquivos.
-b blk	Especifica o tamanho do bloco de dados.
-c	Cria um novo arquivo tar.
-f arq	Usa o arquivo ou dispositivo arq para armazenamento ou recuperação.
-h	Armazena cópia dos arquivos aos quais foi feita referência em vez de armazenar cópia dos links.
-m	Não extrai a data de modificação dos arquivos armazenados.
-p	Extrai as permissões de acesso originais dos arquivos.
-r	Inclui arquivos ao final (append) de um arquivo tar.
-t	Lista o conteúdo de um arquivo tar.
-u	Inclui somente (append) arquivos que não estejam presentes, ou que sejam mais novos do que o arquivo tar.
-v	Lista arquivos processados.
-w	Solicita confirmação antes de cada ação.
-x	Extrai arquivos armazenados em um arquivo tar.

Obs: Os parâmetros podem ser agrupados logo após o comando, (não necessariamente precedidos por -). Nesse caso, quaisquer valores associados devem ser informados na mesma ordem que os parâmetros correspondentes. Compare os quatro últimos exemplos.

Exemplos:

```
# Cria o arquivo files.tar, muda para o diretorio /etc
# e inclui o arquivo arq1 em files.tar, muda para o
# diretório /usr e inclui o arquivo arg2, bem como
# imprime os nomes dos arquivos em stdout.

tar cvf files.tar -C /etc arql -C /usr arq2
```

```
# Todos os exemplos abaixo incluem todos os arquivos
# com extensão .gif no arquivo images.tar,
# com blocagem de 512 bytes.

tar cvfb images.tar 512 *.gif

tar cvf images.tar: -b 512 *.gif

tar -cv -f images.tar -b 512 *.gif

tar -c -v -f images.tar -b 512 *.gif
```

touch

Atualiza a data de acesso do arquivo. Caso o arquivo não exista, touch irá criá-lo vazio por default.

Sintaxe:

```
touch [-c] <arquivo> [arquivo ...]
```

Parâmetro	Descrição
-c	Não cria o arquivo caso ele não exista.

Exemplo:

```
touch helloworld
```

uncompress

Descompacta arquivos compactados porcompress, restaurando seus nomes originais.

Sintaxe:

```
uncompress [-cv] [arquivo ...]
```

Parâmetro	Descrição
-c	Descompacta direcionando a saída para a saída padrão (stdout), sem alterar arquivos.
-v	Imprime suas ações na saída de erro padrão (stderr).

Exemplo:

```
# Descomprime os arquivos comprimidos no
# exemplo do comando compress.

uncompress -v \*.txt.Z
```

uniq

Notifica ou filtra a ocorréncia de linhas repetidas adjacentes em um arquivo texto.

Sintaxe:

```
uniq [-cdu] [-f n] [-s n] [argent [arqsai] ]
```

Parâmetro	Descrição
-c	Precede cada linha de saída com o número de vezes que a mesma ocorreu na entrada.
-d	Mostra apenas as linhas que aparecem repetidas na entrada.
-f n	Ignora os n primeiros campos (palavras separadas por espaços, tabs ou newlines) na comparação entre linhas.
-s n	Ignora os n primeiros caracteres em cada linha de entrada. Se usado com -f, os n primeiros caracteres após os campos ignorados também serão ignorados.
-u	Não mostra as linhas que se repetem na entrada.

Obs: Serão usados argumentos adicionais como, por exemplo, o nome de um arquivo de entrada e um de saída, respectivamente.

uudecode

Decodifica um arquivo preparado por uuencode.

Sintaxe:

```
uudecode [arquivo ...]
```

Obs: uudecode ignora quaisquer cabeçalhos de e-mail ou texto adicional que tenham sido acrescentados no início ou no final dos arquivos.

uuencode

Converte arquivos binários em um equivalente ASCII, o qual poderá ser utilizado para transmissão via e-mail.

Sintaxe:

```
uuencode [arquivo] <nome>
```

Parâmetro	Descrição
arquivo	Arquivo de entrada. Se omitido, os dados serão lidos da entrada padrão (stdin).
nome	O nome que o arquivo terá ao ser decodificado.

Obs: Apenas as permissões de leitura e escrita serão preservadas no arquivo decodificado.uencodegera arquivos, em média, 35% maiores do que seus originais binários. Veja também uudecode.

Exemplos:

```
uuencode Livro.tar.gz livro.tgz | mail editor

uuencode forest.gif trees.gif > image.uu
```

vi

Editor de textos.

Sintaxe:

`vi [arquivo]`

Utilização:

Há dois modos de trabalho no vi, o modo de comando e o modo de edição ou entrada. No modo de comando pode-se movimentar pelo texto, copiar e eliminar linhas e caracteres; o modo de edição é onde entramos o texto propriamente dito.

Vários comandos nos levam ao modo de edição, mas apenas a tecla de escape (<esc>) nos leva ao modo de comando. Pode-se fazer com que um comando se repita automaticamente um determinado número de vezes, bastando digitar o número de repetições antes do nome do comando. Os comandos que permitem essa característica estarão precedidos por [n]; omitir n equivale a digitar 1 antes do comando.

Comando	Descrição
[n]h	Move o cursor n caracteres à esquerda.
[n]l	Move o cursor n caracteres à direita.
[n]k	Move o cursor n caracteres acima.
[n]j	Move o cursorn caracteres abaixo.
/texto	Procura a cadeia texto no arquivo e move o cursor para a sua primeira letra.
[n]a	Insere texto após o cursor.
[n]i	Insere texto antes do cursor.
[n]o	Insere uma nova linha abaixo do cursor e inicia a inserção de texto.
[n]O	Insere uma nova linhaacimado cursor inicia a inserção de texto.
[n]yy	Copia a linha em que se encontra o cursor.
u	Desfaz a última edição feita no texto.
[n]p	Insere a linha copiada após a linha onde está o cursor.
[n]dd	Elimina a linha onde está o cursor (a linha apagada pode ser inserida novamente com p).
[n]x	Elimina o caractere sobre o cursor.
:w [nome]	Atualiza o arquivo editado ou grava-o com outro nome caso tenha sido especificado um nome.
:w nome	Grava o texto no arquivo especificado por nome.
:q[!]	Sai do editor de textos. Se o texto foi modificado e não foi salvo o caractere opcional ! fará com que qualquer modificação seja ignorada.
<esc>	Entra no modo de comando ou cancela comandos não terminados.
[n].	Repete n vezes o último comando que modificou o texto.
número	Move ocursor para a linha número.
0 (zero)	Move o cursor para o primeiro caractere dalinha.
[n]A	Entra no modo de edição, inserindo o texto ao final da linha.
[n]G	Move o cursor até a linhan ou até a última linha caso n não tenha sido especificado.
[n]J	Junta n linhas consecutivas do texto a partir do cursor.
U	Restaura a linha atual ao estado em que ela se encontrava antes da ultima visita do cursor.
[n]~	Inverte a caixa (case) dosn caracteres seguintes

Comando	Descrição
	ao cursor.

Obs: Note que alguns caracteres especiais podem utilizar mais de uma posição na tela e, que algumas linhas de texto podem utilizar mais do que o número de caracteres mostrados em uma linha do terminal; as modificações se aplicam a toda a linha de texto sobre a qualo cursor se encontrarenão à linha do terminal.

wc

Conta o número de palavras, linhas ou bytes (caracteres) de um arquivo.

Sintaxe:

```
wc [-clw] [arquivo ...]
```

Parâmetro	Descrição
-c	Mostra onúmero de caracteres (bytes) do arquivo.
-l	Mostra o número de linhas de um arquivo.
-w	Mostra o numero de palavras de um arquivo.

Obs: Por default wc irá imprimir todas as contagens. Palavras são strings separadas por espaços, TABs ou newlines e linhas são strings terminadas por newline.

Exemplos:

```
wc -l /etc/passwd
```

```
wc -c ./*
```

Comandos de Terminal

banner

Escreve com letras grandes.

Sintaxe:

```
banner [string]
```

Parâmetro	Descrição
string	Texto a ser escrito em letras garrafais.

clear

Limpa o terminal.

Sintaxe:

```
clear
```

echo

Escreve no terminal (Função interna da shell).

Sintaxe:

```
echo [-n] [string ...]
```

Parâmetro	Descrição
-n	Não imprime quebra de linha após argumentos (presente em algumas versões de shell).

Exemplo:

```
echo 'Hello World!!!'
```

mesg

Controla o recebimento de mensagens pelo terminal. Sem argumentos mesg, mostra o estado atual.

Sintaxe:

```
mesg [y|n]
```

Parâmetro	Descrição
y	Liga o recebimento de mensagens.
n	Desliga o recebimento de mensagens.

stty

Altera as opções de I/O de um terminal. Se invocado sem parâmetros, stty mostrará a configuração atual do terminal na saída padrão de erros (stderr).

Sintaxe:

```
stty[-ag] [opção ...]
```

```
stty [carcontrole string]...
```

Utilização: Na segunda sintaxe descrita, stty mapeia um caractere de controle para uma dada string.

Caractere	Descrição
eof	Identifica o caractere de fim de arquivo.
eol	Identifica o caractere de fim de linha.
erase	Identifica o caractere de retrocesso (backspace).
intr	Identifica o caractere de interrupção de processo.
kill	Identifica o caractere que apaga toda a linha do terminal.
quit	Identifica o caractere de término de processo.
susp	Identifica o caractere de suspensão de processo.

Parâmetro	Descrição
-a	Mostra o estado atual de todas as opções.
-g	Mostra o estado atual num formato que pode ser usado como argumento para o comando stty.
Opções	(a presença do caractere - desabilita a opção)
[-]istrip	Limita (ou não) caracteres de entrada a 7 bits.
[-]inlcr	Mapeia (ou não) NL para CR na entrada.
[-]igncr	Ignora (ou não) CR na entrada.
[-]icrnl	Mapeia (ou nao) CR para NL na entrada.
[-]onlcr	Mapeia (ou não) NL para CR-NL na saída.
[-]isig	Habilita (ou desabilita) achecagem de caracteres contra os caracteres especiais INTR, QUIT e SUSP.
[-]icanon	Habilita (ou desabilita) a entrada canônica (processamento de ERASE e KILL)
[-]echo	Ecoa ao terminal (ou não) todo caractere digitado.
[-]echoe	O caractere ERASE deve (ou não) apagar visualmente o caractere anterior ao cursor no terminal.
[-]echok	Ecoa (ou nao) um newline após um caractere KILL.
[-]echonl	Ecoa (ou não) newlines mesmo que echo esteja desligado.
[-]noflsh	Habilita (ou desabilita) descartes após INTR, QUIT e SUSP.
[-]echoctl caso contrá controle.	Seligado, ecoa caracteres de controle como?X, rio ecoa os próprios caracteres de
[-]echoke	O caractere KILL deve (ou não) apagar visivelmente a linha corrente do terminal.
rows n	Configura a altura do terminal em linhas.
columns i	Configura a largura do terminal em i colunas.
ek	Volta os caracteres KILL e ERASE para os defaults do sistema.
size	O tamanho do terminal será impresso em uma linha; primeiro o número de linhas, depois o de colunas.
sane	Configura todos os modos para valores razoáveis para utilização interativa em terminais.
[-]crt	Configura ou desliga todos os modos adequados a um device CRT.
[-]tostop	Impede (ou não) a saída de dados de processos em background para o terminal.

Exemplos:

```
# Muito útil quando a tecla backspace não funciona
# corretamente. O ^H do exemplo foi obtido pressionando-se
# a tecla BackSpace para  associá-la à ação de retrocesso
# do cursor.

erase ^H

stty tostop
```

talk

Permite uma conversa com outro usuário.

Sintaxe:

```
talk usuário [ term ]
```

Parâmetro	Descrição
term	Indica o terminal ao qual se destina a conversa.

Obs: Se ambos os usuários estiverem na mesma máquina, então usuário será o login da pessoa com quem se quer falar. Se alguém estiver em outra máquina, o formato a ser utilizado sera usuario@nome.da.maquina. Para interromper a conversa basta pressionar o caractere de interrupção (geralmente CTRL-C). Para impedir o recebimento de mensagens use o comando mesg.

Exemplo:

```
talk noel@polo.norte.org
```

tput

Usa os dados de termcap para disponibilizar características dependentes do tipo do terminal para a shell de maneira transparente.

Sintaxe:

```
tput [-T tipo] atributo
```

Parâmetro	Descrição
-T tipo	Indica o tipo do terminal. Normalmente desnecessária, usa o valor da variável de ambiente TERM como default.
atributo	Indica o atributo das tabelas de características de terminal (terminfo). Abaixo alguns atributos válidos:
bel	Emite um sinal sonoro (bip).
blink	Envia a sequência de comandos que instrui o terminal a mostrar o texto piscando.
bold	Envia a sequência de comandos que instrui o terminal a mostrar os caracteres em negrito (bold).
civis	Torna o cursor invisível.
clear	Envia a sequência de limpeza de terminal para o terminal atual.
cnorm	Restaura os atributos do cursor.
cols	Mostra o número de colunas do terminal.
cup y x	Envia a sequência que move o cursor para a posição X,Y do terminal.
lines	Mostra o número de linhas do terminal.
rmso	Sequência para cancelar texto reverso.
smso	Sequência para iniciar texto reverso.

Obs: tput envia as sequências de terminal para stdout, sendo possivel redirecionar sua saída para arquivos ou dispositivos, oque pode ser

muito útil. Alguns terminais não implementam todas as características disponibilizadas por tput.

Exemplos:

```
# Digite o texto acima em uma linha ou em
# um script e veja a string "Oi!"
# aparecer no meio da tela.

tput clear; tput 12 34;echo "Oi!"; tput 24 0
```

tset

Sintaxe:

```
tset [-Qs] [-] [-e c] [-k c] [-i c] [-m map] [term]
```

Parâmetro	Descrição
term	Especifica um tipo de terminal,
-I	Não envia as strings de inicialização para o terminal.
-Q	Não mostra os valores dos caracteres erase e kill.
-s	Gera os comandos da shell para configurar a variável TERM.
-	Mostra apenas o nome do terminal.
-e c	Configura o caractere erase como sendo c.
-k e	Configura o caractere kill como sendo c.
-m map	Mapeia portas (ports) a terminais. **map** tem o formato **[porta] [op veloc]:tipo**, onde op é um operador. Oteste será feito contra otipo da porta e sua velocidade. Caso o teste seja positivo, configura o tipo de terminal correspondente. Se tipo começar com uma interrogação, tset pedirá confirmação ao usuário. Os operadores válidos são:

Operador	Significado
>	Maior do que.
<	Menor do que.
=	Igual a.
@	Igual a (equivalente a =).
!	Negação (usado em conjunto com outro operador).

Obs: Os valores do mapeamento devem aparecer entre aspas simples (') para evitar interpretação de caracteres especiais pela shell.

Exemplo:

```
# Mostra os comandos da shell necessários para
 # configurar o terminal como vt100 se estiver
 # associado a uma linha discada a menos de 1200 bauds,
 #  vt220 a 9600 bauds e vt100 caso seja um xterm.

tset -IQs -m 'dialup<1200:vt100'
     -m 'dialup@9600:vt220' 'xterm:vt100'
```

tty

Mostra o nome completo do terminal do usuário. O valor de saída será zerosea entrada padrão (stdin) for um terminal, caso contrário seráum.

Sintaxe:

```
tty [-s]
```

Parâmetro	Descrição
-s	Modo silencioso. Não imprime nada, apenas devolve o valor de retorno.

wall

Escreve uma mensagem para o terminal de todos os usuários conectados ao sistema. Se invocado sem parâmetros, wall lerá a entrada padrão até encontrar um caractere de fim de arquivo, geralmente "D (ctrl-D).

Sintaxe:

```
wall [parâmetro]
```

Parâmetro	Descrição
parâmetro	Em alguns sistemas denota a mensagem a ser escrita, em outros um arquivo que contém a mensagem.

write

Envia uma mensagem a um terminal.

Sintaxe:

```
write usuário [terminal]
```

Parâmetro	Descrição
term	Indica para qual terminal deve ser enviada a mensagem.

Obs: write continuará copiando linhas para o outro terminal até encontrar o caractere de fim de arquivo (EOF, normalmente CTRL-D); neste momento, write escreverá EOT no outro terminal e terminará sua execução.

Exemplo:

```
write suzie
Oi, vamos almocar? :)
^D
```

Controle de Processos do Usuário

atq

Mostra a fila de execução do at. Em alguns sistemas é equivalente ao comando at -l.

Sintaxe:

```
atq
```

atrm

Remove um job da fila de execução doat. Em alguns sistemas equivale ao comando at -r.

Sintaxe:

```
atrm <job> [job ...]
```

at

Executa comandos em outra hora.

Sintaxe:

```
at [-m] [-f script] hora [data] [+ incr]
```

Parâmetro	Descrição
hora	Pode ter os seguintes formatos (h=horas m=minutos):hh:mm ou hhmm. Os valores noon, midnight e now são válidos.
data	Um nome de mês (Jan, Feb, ... Dec) seguido de um dia e, possivelmente, um ano precedido por vírgula. Os valores today e tomorrow são válidos.
+ incr	Um número seguido de uma das seguintes palavras: minutes, hours, days, weeks, months ou years (ou singular). O modificador next pode aparecer antes do incremento (equivale a + 1).
-f script	Lê os comandos do arquivo script a serem executados.
-ljobs	Lista os jobs pendentes na fila do at. Se uma lista de identificadores de jobs (job ids) for passada como parâmetro, listará apenas o status daqueles jobs. Essa opção inexiste em alguns sistemas; para estes casos, use o comando atq.
-m	Envia e-mail avisando que o comando foi executado.
-r jobs	Remove todos os jobs especificados pela lista de job ids (jobs). Alguns sistemas não implementam essa opção; para estes casos, utilize o comando atrm.

Obs: Caso não seja fornecido um arquivo com os comandos a serem executados, at lerá os comandos da entrada padrão (stdin) até encontrar o caractere de fim de arquivo (EOF).

Exemplos:

```
at -m now + 1 week

at -mf comandos now + 3 days

at 18:30 Jan 24
```

bg

Põe em background um processo em execução.

Sintaxe:

```
bg [%id]
```

Utilização: Comando interno da shell. Quando um processo está ocupando o terminal e digitamos ^Z (CTRL-Z) o processo recebe um sinal SIGSTOP e a shell interrompe a execução do mesmo. bg irá colocar o processo interrompido no modo de execução em background (em segundo plano) e o processo poderá continuar sua execução.

Parâmetro	Descrição
%id	No caso de haver vários processos interrompidos, indica qual processo será colocado em background.

Obs: O identificador (id) do processo será indicado pela shell assim que executarmos um comando em modo background (com '&') ou interrompermos um processo pressionando ^Z. Uma lista dos processos pode ser obtida com o comando **jobs**. Um processo em background geralmente pode escrever sua saída no terminal (isso pode ser desabilitado com stty) e interromperá sua execução se precisar ler dados da entrada padrão (stdin).

Exemplo:

```
# sequência exemplo de comandos, $ é o prompt
# da shell
$ find / -print
^Z
[1]+ Stopped            find / -print
$ bg %1
[1]+  find / -print
```

env

Executa um comando em um ambiente modificado.

Sintaxe:

```
env [-][-i] [-u nome] [var=valor]... [comando [args]
```

Parâmetro	Descrição
-i	Começa um ambiente do zero, ignorando as variaveis de ambiente atuais.
-	O mesmo que -i.
-u nome	Elimina a variável dada por nome ao ambiente montado por env.
var=valor	Cria a variável de ambiente dada por var e lhe atribui o valor especificado.

Parâmetro	Descrição
comando	Comando a ser executado em ambiente modificado.
args	Argumentos do comando a ser executado por env.

Obs: Caso não seja especificado um comando, env listará todas as variáveis do ambiente montado. As variáveis de ambiente criadas. por env só estarão disponíveis para o comando especificado durante sua execução.

Exemplos:

```
# Executa o comando echo num ambiente onde a
# variável NOME está definida.

env NOME=Eduardo echo SNOME

# Executa o comando echo num ambiente onde nenhuma variável
# está definida.

env -i echo $HOME
```

fg

Põe em foreground um processo em execução.

Sintaxe:

```
fg [%id]
```

Utilização: Comando interno da shell. Quando um processo foi interrompido ou está sendo executado em background (veja bg), o comando fg transferirá o controle do terminal para o processo.

Obs: Uma lista dos processos e seus respectivos identificadores poderá ser obtida com o comando jobs. Se nenhum identificador de job for especificado, o último processo interrompido será colocado em foreground.

Parâmetro	Descrição
id	No caso de haver varios processos interrompidos, indica qual processo sera colocado em foreground.

Exemplo:

#sequência de comandos, $ indica o prompt da shell

```
$ vi &
[1] xxx  (<-- número do processo)
[1]+ Stopped (tty output) vi
$ fg %1
```

finger

Mostra informações sobre usuários locais ou remotos.

Sintaxe:

```
finger [-lmsp] [usuário ...]
```

Parâmetro	Descrição
-l	Saida em formato longo.
-m	Compara apenas o nome de login do usuário e não seu nome completo.
-s	Força a saída em modo simplificado (short).
-p	Não imprime o arquivo .plan do usuário.

Obs: Os usuários podem ser especificados pelos userids (ou login names) para usuários locais, ou usuário@nome.da.maquina para usuários de máquinas remotas.

Exemplos:

```
finger root

finger president@whitehouse.gov
```

groups

Mostra os grupos de um usuario. Se nenhum usuario for especificado, mostra informações sobre o usuário que invocou o comando.

Sintaxe:

```
groups [usuário]
```

id

Mostra o identificador do usuário.

Sintaxe:

```
id [-Ggu]
```

Parametro	Descrição
-G	Mostra apenas os identificadores de grupo real, efetivo e suplementares do usuário.
-g	Mostra apenas o identificador de grupo efetivo do usuário.
-u	Mostra apenas o identificador efetivo do usuário

jobs

Lista processos em execução pela shell.

Sintaxe:

```
jobs [-l]
```

Parâmetro	Descrição
-l	Lista também o número de cada processo.

kill

Envia um sinal a um processo.

Sintaxe:

```
kill [-sinal] <processo> [processo ...]

kill -s sinal [processo ...]

kill -l [sinal]
```

Parâmetro	Descrição
sinal	Pode ser tanto o número do sinal como seu nome.
-s	Em alguns sistemas é necessária a presença da opção -s para se especificar o sinal que se quer enviar.
-l	Lista todos os nomes e números de sinais. Presente em apenas algumas versões de kill. Alguns sinais comuns:
HUP	1
INT	2
QUIT	3
ABRT	6
KILL	9
ALRM	14
TERM	15

Obs: Sua sintaxe varia de sistema para sistema e entre shells diferentes. Em alguns sistemas pode ser encontrado como um comando separado da shell. A lista de processos pode ser uma lista de números ou de nomes de processos. O sinal SIGTERM (15) é enviado por default. Somente o superusuário pode enviar sinais a processos de outros usuários.

Exemplos:

```
kill -HUP 1

kill -9 %2
```

last

Mostra os últimos logins de usuário ou terminal. O tamanho do registro de log é determinado pelo administrador do sistema.

Sintaxe:

```
last [-#] [-t term] [nome ...]
```

Parâmetro	Descrição
-#	Limita a saída do comando em Élinhas.
-t term	Seleciona um terminal (apenas para as implementações que não permitem terminais em linha de comando).

Exemplo:

```
last macan
```

login

Inicia a sessão do usuário. Para encerrar a sessão, usa-se o comando logout.

Sintaxe:

```
login [usuário]
```

Exemplo:

```
login root
```

logout

Encerra a sessão do usuário (função interna da shell). Veja o comando login.

Sintaxe:

```
logout
```

nice

Faz com que um processo seja executado com uma prioridade de escalonamento diferente da padrão.

Sintaxe:

```
nice [-#] <comando> [args]
```

Parâmetro	Descrição
-#	Valor a ser adicionado à prioridade do processo que será executado. Quanto maior este valor, MENOR será a prioridade de escalonamento do processo. Em alguns sistemas este valor representa o valor absoluto da prioridade.
comando	O comando a ser executado com uma prioridade diferente.
args	Os argumentos do comando a ser executado.

Obs: Os valores padrão de nice e as prioridades limites variam entre as várias versões do Unix, portanto convém consultar os manuais do sistema para a obtenção dos valores corretos. O nome nice vem do fato de alguém estar sendo "bacana" com os outros usuários do sistema, baixando a prioridade de um processo. Apenas o super-usuário pode aumentar a prioridade de escalonamento (especificando valores negativos de nice).

Exemplo:

```
nice -4 find / -name readme.txt -print
```

nohup

Faz com que um comando continue sua execução mesmo que o usuário que o iniciou saia do sistema.

Sintaxe:

```
nohup <comando> [args]
```

Parâmetro	Descrição
comando	O comando a ser executado. Este poderá ser executado em background se após seus argumentos estiver presente o caractere &.
args	Os argumentos a serem passados em linha de comando para o comando cmd.
Obs:	Em algumas versões de nohup, a saída do comando executado é armazenada num arquivo chamado nohup.out, no diretório corrente, ou no diretório HOME do usuário caso o diretório corrente não ofereça permissão de escrita.

Exemplo:

```
nohup find / -print &
```

passwd

Altera a senha do usuario.

Sintaxe:

```
passwd [usuario]
```

Obs: Apenas o superusuário (root) pode mudar a senha de outros usuários.

ps (BSD)

Mostra informações sobre os processos ativos.

Sintaxe:

```
ps [-aCejlSuvwxp] [-txx] pids
```

Obs: Se invocado sem parâmetros, ps mostrará informações suscintas sobre os processos associados ao terminal de controle. Esta seção cobre os parâmetros válidos para os vários sistemas que adotam a versão Berkeley do comando ps.

Parâmetro	Descrição
-a	Mostra também informações de processos de outros usuários.
-C	Mostra o consumo da CPU, ignorando-se o "resident time".
-e	Mostra também variáveis do ambiente (environment) em que o processo estiver sendo executado.
-j	Mostra informações associadas ao sistema de job control.
-l	Mostra a saída no formato longo.
-S	Muda o modo através do qual o tempo do processo é calculado, somando-se o tempo dos processosfilhos que já terminaram sua execução.
-u	Mostra o nome do usuário e hora do início do

Parâmetro	Descrição
	processo.
-v	Mostra informações associadas ao uso da memória virtual pelo processo.
-w	Não trunca as linhas para que caibam no terminal.
-x	Mostra também os processos não associados a um terminal de controle.
-p pid	Mostra o processo cujo número é dado por pid.
-t xx	Mostra todos os processo associados ao terminal especificado por xx.

Exemplo:

```
ps -aux | grep macan
```

ps (SYSV)

Mostra informações sobre os processos ativos.

Sintaxe:

```
ps -[adefl] [-t terms] [-p procs] [-u uids] [-g gids]
```

Obs: Se invocado sem parâmetros, ps mostrará informações suscintas sobre os processos associados ao terminal de controle. Esta seção cobre os parâmetros válidos para os vários sistemas que adotam a versão System V do comando ps.

Parâmetro	Descrição
-a	Mostra informações sobre os processos mais requisitados; ou seja, todos, menos os líderes de grupo de processos e aqueles não associados a um terminal.
-d	Mostra informações sobre todos os processos exceto líderes de grupo de processos (process group leaders).
-e	Mostra informações sobre todos os processos em execução no presente momento.
-f	Gera uma listagem completa (full). _
-l	Gera uma listagem no formato longo (long).
-t term	Mostra apenas os processos associados ao terminal especificado por term.
-p procs	Mostra apenas os processos cujos números (PIDs) estejam presentes na lista procs.
-u uids	Mostra apenas os processos cujos nomes (ou números) dos usuários proprietários (owners) estejam presentes na lista uids.
-g gids	Mostra apenas-os processos cujos grupos estejam presentes na lista uids.

Exemplo:

```
ps -efu macan
```

su

Troca o ID efetivo do usuário. Solicita confirmação através de senha para efetuar a execução da shell, exceto quando executado pelo superusuario (root).

Sintaxe:

```
su [-] [user]
```

Parâmetro	Descrição
-	Executa uma shell de login, carregando todo o ambiente do usuário.
user	O nome do usuário para o qual se quer altemar. Se omitido, su tomará o usuário root como padrão.

Exemplo:

```
su - macan
```

users

Mostra quem está usando o sistema.

Sintaxe:

```
users
```

Obs: Em algumas implementações de users pode-se indicar um arquivo alternativo ao arquivo utmp de onde users retirará a informação.

whoami

Mostra o ID efetivo do usuário.

Sintaxe:

```
whoami
```

who

Mostra quem está usando qual terminal.

Sintaxe:

```
who [arquivo] [am i]
```

Parâmetro	Descrição
arquivo	Indica o nome de um arquivo alternativo a ser usado como fonte de informações para who.
am i	Mostra o nome real do usuário (real user name).

Exemplo:

```
who am i
```

Obs: Por default, who usará os dados do arquivo utmp, a não ser que um arquivo alternativo seja indicado. Se invocado sem argumentos, who mostrará o login de todos os usuários com processos associados a terminais, o nome dos terminais e dados sobre atividade e tempo de login.

w

Mostra os usuários conectados ao sistema e o que estão fazendo.

Sintaxe:

```
w [-hls] [user]
```

Parâmetro	Descrição
-h	Não mostra o cabeçalho na saída.
-l	Saída em formato longo (default).
-s	Saída em formato simplificado.

Obs: Se nenhum usuário for especificado, w retornará informações sobre todos os usuários atualmente ativos.

Miscelânea

apropos

Mostra informações sobre um assunto. Equivalente a man -k.

Sintaxe:

```
apropos [chave ...]
```

Parâmetro	Descrição
chave	Identifica um assunto.

Exemplo:

```
apropos directory
```

biff

Notifica a chegada de mensagem de correio eletrônico.

Sintaxe:

```
biff [yn]
```

Parâmetro	Descrição
y	Liga o aviso de chegada de mensagens.
n	Desliga o aviso de chegada de mensagens.

calendar

Serviço de agenda. Notifica o usuário das atividades do dia.

Sintaxe:

```
calendar [-a]
calendar [-]
```

Parâmetro	Descrição
-a	Notifica as atividades do dia aos usuários do

Parâmetro	Descrição
sistema	via e-mail. É executado automaticamente pelo sistema todos os dias.
-	Em alguns sistemas a opção - substitui -a.

Obs: Os compromissos devem estar armazenados em um arquivo chamado calendar, no diretório HOME do usuário, um por linha, no formato MÉS/DIA DESCRIÇÃO.

cal

Imprime o calendário de um determinado mês/ano.

Sintaxe:

```
cal [[mês] ano]
```

Parâmetro	Descrição
mês	O mês para o qual se quer o calendário.
ano	O ano para o qual se quer o calendário.

Obs: Se invocado sem parâmetros, cal imprimirá o calendário do mês atual (data do sistema).

Exemplo:

```
#Imprime o calendário de julho de 1974.

cal 7 1974
```

crontab

Instala, lista ou remove arquivo crontab de usuário.

Sintaxe:

```
crontab [arquivo]

crontab [ -! | -r| -e]
```

Parâmetros	Descrição
-l	Mostra o arquivo crontab atual do usuario.
-r	Remove o arquivo crontab atual do usuário.
-e	Edita o arquivo crontab atual do usuário. Presente em algumas versões de crontab.
arquivo	Indica um arquivo de crontab a ser posto em atividade, veja o formato do arquivo crontab abaixo.

Obs: Um arquivo de crontab é um arquivo texto onde cada linha especifica um comando e a periodicidade com queo mesmo será executado (automaticamente) pelo sistema, no formato:

minuto hora dia mês dia da semana comando

Os campos podem ser separados por espaços ou TABs, os cinco primeiros podem assumir um valor numérico, um intervalo (no formato m-n), asterisco ou uma lista de quaisquer destes, separada por

vírgulas. Os valores válidos são O a 59 para minuto, 0 a 23 para hora, 1 a 31 para dia, 1a 12 para mês e 0 a 6 para dia da semana (0=domingo).

Exemplos:

```
# Seguem exemplos de linhas válidas de um arquivo crontab:

# Enviará e-mail ao autor deste guia todo dia 16 de julho
# às 00:00h.
0 0 16 7 * mail -s'Feliz aniversario!' macan

# Executa o script teste.pl de quinze em quinze minutos,
    # de segunda a sexta-feira, no horário comercial.

0,15,30,45 8-11,13-16 * * 1-5 /bin/teste.pl
```

date

Mostra a data e a hora atuais do sistema, com a possibilidade de formatação da saída.

Sintaxe:

```
date [-u] [+formato]
```

Parâmetro	Descrição
-u	Universal time. Mostra o horário GMT.
+formato	formato é uma string que contém informações sobre a formatação da saída do comando date. Os seguintes operandos especiais serão substituídos por seus valores correspondentes quando encontrados dentro da string formato:
%n	Caractere newline (quebra de linha).
%t	Caractere de tabulação (TAB).
%m	Mês do ano (01 a 12).
%d	Dia do més (01 a 31).
%y	Últimos dígitos do ano (00 a 99).
%D	Data no formato MM/DD/AA.
%H	Hora (00 a 23).
%M	Minuto (00 a 59).
%S	Segundo (00 a 59).
%T	Hora no formato HH:MM:SS.
%j	Dia do ano (001 a 366).
%w	Dia da semana (Dom=0 a Sáb=6).
%a	Dia da semana abreviado (Sun ... Sat).
%h	Mês abreviado (Jan, Feb,... Nov,Dec).
%r	Hora no formato am/pm.

Exemplos:

```
date +"Agora são %H horas e %M minutos$n"

date +' Hoje é %d/%m/%y '

date +"\\ Hoje é %d/%m/%y \\"
```

false

Nãofaz nada, apenas retorna status diferente de zero. Muito útilem shell scripts. Veja também o comando true.

Sintaxe:

```
false
```

hostname

Mostra ou configura o nome da máquina em que se está conectado. Apenas o super usuário (root) pode configurar o nome da máquina.

Sintaxe:

```
hostname [nome]
```

Parâmetro	Descrição
nome	Se informado um parâmetro em linha de comando, hostname tentará configurar o nome da máquina.

lpr

Envia arquivos para a fila de impressão.

Sintaxe:

```
lpr [-c] [-d dest] [-n& ] [arquivo ...]
```

Parâmetro	Descrição
-c	Envia cópias dos arquivos para a fila de impressão. Modificações feitas nos arquivos durante aimpressão não se refletirão no resultado impresso.
-d dest	Envia um arquivo para impressão na impressora cujo nome seja dest.
-n#	Imprime # copias de cada arquivo.

Exemplo:

```
lpr -dlaserl readme.txt install.txt
```

man

Consulta os manuais on-line do sistema.

Sintaxe:

```
man [-fk] [seção] [chave ...]

man [-fk] [-s seção] [chave ...]
```

Parâmetro	Descrição
-f	Mostra descrições de uma linha sempre que chave coincidir com uma entrada do manual.
-k	Pesquisa informações sobre palavras-chave nas descrições das páginas dos manuais on-line.

Parâmetro	Descrição
-s seção	Algumas versões de man necessitam do parâmetro -s caso se queira especificar uma seção dos manuais onde buscar a informação.

Obs: As páginas do manual são tradicionalmente divididas em 8 seções principais, a saber:

Seção	Descrição
1	Comandos de usuário.
2	Chamadas de sistema.
3	Subrotinas (programação).
4	Dispositivos.
5	Formatos de arquivos.
6	Jogos.
7	Miscelânea.
8	Administração do sistema.

Exemplos:

```
man man

man ls

man -k tape
```

sleep

Suspende a execução por um tempo determinado.

Sintaxe:

```
sleep <segundos>
```

tee

Mostra a saída de um programa e a escreve em um arquivo simultaneamente.

Sintaxe:

```
tee [-ai] [arquivo ...]
```

Parâmetro	Descrição
-a	Append. Concatena a saída ao arquivo, em vez de sobrescrevê-lo.
-i	Ignora interrupções.

Exemplo:

```
# Guarda uma cópia de sua seção de ftp em ftp.out.

ftp ftp.cdrom.com | tee ftp.out
```

test

Checa tipos de arquivos e compara valores. Resultado do teste como valor de retorno do comando.

Sintaxe:

```
test [expr]
```

Parâmetro	Descrição
-b arq	Verdadeiro se arq existir e for um block device.
-c arq	Verdadeiro se arq existir e for um character device.
-d arq	Verdadeiro se arq existir e for um diretório.
-f arq	Verdadeiro searq existir e for um arquivo regular.
-g arq	Verdadeiro se arq existir e seu bit setgid estiver ligado.
-h arq	Verdadeiro se arq existir e for um link simbólico.
-k arq	Verdadeiro searqexistir e seu bitsticky estiver ligado.
-n str	Verdadeiro se a string str tiver comprimento diferente de zero.
-p arq	Verdadeiro se arq existir e for um FIFO.
-r arq	Verdadeiro seargexistiretiver permissão de leitura.
-s arq	Verdadeiro seargexistire tivertamanho diferente de zero.
-t [n]	Verdadeiro se o arquivo aberto com descritor de arquivo n estiver associado a um terminal (defaut n=1).
-u arq	Verdadeiro se arq existir e seu bit suid estiver ligado.
-w arq	Verdadeiro se arq existir e tiver permissão de escrita.
-x arg	Verdadeiro se arq existir e for executável.
-z str	Verdadeiro se a string str tiver comprimento zero.
s1	Verdadeiro se s1 tiver comprimento maior do que zero.
s1!=s2	Verdadeiro se as strings s1 e s2 não forem idênticas.
s1=s2	Verdadeiro se as strings s1 e s2 foremidénticas.
n1 op n2	Verdadeiro, de acordo comos valores numéricos de n1 en2 eo operador utilizado. Operadores podem ser:
-eq	Igual a.
-ne	Diferente de.
-gt	Maior do que.
-ge	Maior ou igual a.
-It	Menor do que.
-le	Menor ou igual a.

Operador	Significado
!	Operador de negação.
-a	Operador “e” binário.
-o	Operador “ou” binario (-a tem maior precedéncia do que -o).

Exemplos (shell scripts):

```
#!/bin/sh
# Script para testar tipo de um arquivo.
# Exemplo 1 do comando test.
# Uma implementação simplista do comando "file"
# do Unix, em shell script.
# Eduardo Macan - 1997

if "tests
  then
    for FILE In S*
    do
      if test -h SFILE
      then
        echo "SFILE e' um link simbólico."
      elif test -b SFILE
      then
        echo "SFILE e' um block device."
      elif test -c $FILE
      then
        echo "SFILE e' um character device."
      elif test -d $FILE
      then
        echo "SFILE e' um diretorio."
      elif test -p $FILE
      then
        echo "SFILE e' um FIFO."
      elif test -f $FILE
      then
        echo "SFILE e' um arquivo comum."
      fi
    done
  else
    echo "uso: testa [arquivo... ]"
  fi

#!/bin/sh
#Exemplo 2 do comando test
#Eduardo Marcel Macan - 1997
echo "Quantos anos voce tem?"
read AGE
if test SAGE CIC
then
  echo "Voce ainda nao nasceu?"
elif test SAGE -eq 18
then
  echo "Parabens, voce ja pode ser preso!"
elif test SAGE -lt 10
then
  echo "Seu pai sabe que voce esta aqui?"
elif test SAGE -gt 18 -a SAGE -lt 80
then
  echo "Lembre-se do Imposto de renda...
elif test SAGE -gt 79
then
  echo "Vida longa e prospera."
else
```

```
  echo "Volte para seu videogame!"
fi
```

```
#!/bin/sh
#Exemplo 3 do comando test
echo "Voce gosta de batatas [s|n]?"
read RESP
if test SRESP
  then
    if test SRESP = s -o SRESP = S
       then echo "Eu tambem gosto! :)"
       exit 0
    elif test SRESP ='n -o SRESP = N
      then echo "Eu gosto. :p"
      exit 1
    fi
fi
echo "Responda s ou n da proxima vez! :("
exit 1
```

time

Mede o tempo de execução de um comando.

Sintaxe:

```
time <comando>
```

tr

Substitui caracteres de entrada presentes em str1 por seus correspondentes em str2.

Sintaxe:

```
tr [-cds] [strf [str2]]
```

Parâmetro	Descrição
-c	Efetua a troca em todos os caracteres que não estejam especificados em str1 (complemento).
-d	Elimina ocorrências de caracteres de str1 na entrada.
-s	Elimina repetições de caracteres de str2 na saída.

Exemplo:

```
# Mostra o nome de todos os arquivos do diretório
# corrente em maiúsculas.
ls | tr 'a-z' 'A-Z'
```

true

Não faz nada, apenas retorna status igual a zero. Muito útil em shell scripts. Veja também o comando false.

Sintaxe:

```
true
```

uname

Mostra informações sobre o sistema operacional e o hardware. Se não forem fornecidos parâmetros, a ação default é imprimir o nome do sistema operacional.

Sintaxe:

```
uname [-amnrsv]
```

Parâmetro	Descrição
-a	Equivale a especificar todas as opções.
-m	Mostra o nome da plataforma na qual o sistema está sendo executado.
-n	Mostra o nome da maquina (hostname).
-r	Mostra o "release level" do sistema operacional.
-s	Mostra o nome do sistema operacional.
-v	Mostra a versão (version level) do sistema operacional.

uptime

Mostra a data atual, o tempo desde o último boot do sistema, o número de usuários on-line e a média de carga nos últimos 5, 10 e 15 minutos.

Sintaxe:

```
uptime
```

Obs: Por carga entenda-se o número de processos esperando para serem executados em um determinado momento no sistema.

whatis

Consulta manuais on-line do sistema e mostra um sumário de uma linha sobre palavras-chave. Equivale a man -f

Sintaxe:

```
whatis [chave ...]
```

Parâmetro	Descrição
chave	A palavra a ser procurada nos manuais on-line do sistema.

xargs

Constrói e executa linhas de comando a partir de stdin.

Sintaxe:

```
xargs [-pt] [-e [str }] [-i [str ]] 1 [n]] [-n n]
      [-s n] [comando [arg ...]]
```

Parâmetro	Descrição
-t	Modo detalhado. Imprime a linha de comando em stderr antes de executá-la.
-e str	Usa str como string de fim de arquivo. Ignora

Parâmetro	Descrição
	toda a entrada que vier após uma linha contendo str.
-i str	Substitui ocorrências de str nos argumentos iniciais (veja args) por nomes lidos de stdin.
-l n	Usa no máximo n linhas de texto por linha de comando; default=1.
-n n	Usa no máximo n argumentos por linha de comando.
-p	Modo interativo; pede confirmação antes de executar cada linha de comando.
-s n	Usa no máximo n caracteres por linha de comando. O defaulté o maior possível, variando de sistema para sistema.
-x	Termina se o tamanho máximo da linha de comando (veja -s) for excedido.
comando	O comando a ser executado em cada linha de comando montada.
arg	Argumento inicial a ser informado ao comando.

Exemplos:

```
# Comprime todos os arquivos com extensão txt
  # abaixo do diretório corrente, pedindo confirmação,
  # dois arquivos de cada vez.

find . -name N*. txt | xargs -pn 2 compress

# Comprime arquivos um a um até encontrar a string
  # gif em um nome de arquivo ou diretório (saida de
  # find) e encerra a leitura de stdin.

find .| xargs -pn 1 -e gif compress -v
```

yes

Imprime uma resposta afirmativa indefinidamente. Se invocado sem parametros, yes escreve y.

Sintaxe:

```
yes [mensagem]
```

Parâmetro	Descrição
mensagem	mensagem a ser impressa.

Exemplos:

```
yes | rm -i \*.gif

# Para comandos que pedem confirmação em português.
yes s
```

ftp

Transfere arquivos entre máquinas da rede.

Sintaxe:

```
ftp [-ginv] [maquina]
```

Parâmetro	Descrição
-g	Desabilita o “globbing” dos nomes de arquivos, habilitado por default. Veja o comando glob abaixo.
-i	Não pede confirmação antes da transferência de múltiplos arquivos. Veja o comandoprompt abaixo.
-n	Desabilita o “auto-login”. Veja o comando open abaixo.
-v	Habilita a saida descritiva. Default se a entrada padrão estiver associada a um terminal.

Comandos:

Comando	Descrição
![cmd [args]]	Invoca uma shell na máquina local e executa o comando cmd.
append loc[rem]	Concatena o arquivo *loc* na máquina local ao arquivo remoto *rem*.
ascii	Ativa o modo de transferência adequado para a transferência de arquivos texto.
bell	Soa um alarme a cada transferência de arquivo completada.
binary	Ativa o modo de transferência adequado a arquivos binários.
bye	Encerra a sessão como servidor remoto e saido programa ftp. Um caractere EOF (geralmente ˆD) tem o mesmo efeito.
case	Converte o nome dos arquivos transferidos para minúsculas.
cd dir	Muda diretório de trabalho na máquina remota para dir.
cdup	Muda para o diretório pai do diretório atual no servidor.
chmod modo arq	Muda o modo de acesso do arquivo remoto arq para aquele especificado por modo.
close	Termina a sessão com o servidor remoto e retorna ao interpretador de comandos.
cr	Liga/desliga a filtragem de caracteres CR durante transferências do tipo ascii. Ativada é o default.
delete arq	Remove o arquivo arq da máquina remota.
dir [dir] [arq]	Mostra o conteúdo do diretório remoto dir, opcionalmente colocando a saída no arquivo local arq.
disconnect	O mesmo que close.
get arq [loc]	Transfere o arquivo arq da máquina remota para a máquina local, renomeando-o para loc caso esse parâmetro seja fornecido.
glob	Liga/desliga a expansão de nomes de arquivos para os comandos mget , mput e mdelete. O default é a expansão de nomes ativada.
hash	Mostra um caractere # (hash) para cada bloco de dados recebido (1Kbyte).

Comando	Descrição
help [cmd]	Mostra uma mensagem informativa sobre o comando cmd. Se nenhum comando for especificado, lista todos os comandos disponiveis.
idle[seg]	Ajusta o relógio de inatividade no servidor remoto para seg segundos. Se seg não for especificado, mostra o valor atual do contador.
lcd [dir]	Muda o diretório de trabalho na máquina local. Se dir não for especificado, o diretório HOME do usuário será usado.
ls [dir] [arq]	Similar a dir, porém inclui informações dependentes de sistema fornecidas pelo servidor.
mdelete [arqs]	Similar a delete para múltiplos arquivos.
mdir arqs loc	Gera uma listagem de arqs no arquivo local loc.
mget arqs	Similar a get para multiplos arquivos.
mkdir dir	Cria um diretório na máquina remota.
mls arqs loc	Gera listagem similar à de ls para os arquivos dados por arqs e grava sua saída no arquivo local loc.
modtime arq	Mostra a data de modificação do arquivo remoto arq.
mput arqs	Copia os arquivos dados por arqs para o diretório de trabalho corrente na máquina remota.
newer arq	Copia um arquivo somente se a versão remota for mais recente do que a versão local, cujo nome será dado por arq.
open host[porta]	Abre uma conexão com o host na porta especificada. Se a opção auto-login estiver habilitada, ftp tentará logar no host remoto.
prompt	Liga ou desliga o modo interativo, onde ftp pede confirmação antes de transferir ou eliminar múltiplos arquivos.
put arq [rem]	Armazena um arquivo na máquina remota, opcionalmente com o nome dado por rem.
pwd	Imprime o diretório corrente da máquina remota.
quit	O mesmo que bye.
recv rem [loc]	O mesmo que get.
remotehelp [cmd]	Pede informações sobre os comandos do servidor ftp remoto.
rename old new	Altera o nome do arquivo remoto chamado old, chamando-o de new.
reset	Sincroniza o cliente com o servidor de ftp.
rmdir dir	Remove do servidor o diretório especificado.
runique	Evita a sobreposição de arquivos, concatenando a seu nome um sufixo numérico.
sendarq loc [rem]	O mesmo que put.
status	Mostra o estado atual do ftp.
sunique	O equivalente a runique para o sistema remoto.
type [tipo]	Muda o "tipo de representação" para tipo. Os tipos ascii e binary (ou image) são válidos, sendo ascii o default.
user nome	Identifica o usuário nome com o servidor ftp. Se uma senha de acesso for necessária o servidor irá pedi-la.
verbose	Aciona o modo detalhado de apresentação. Default se os comandos estiverem sendo digitados em um terminal.

Comando	Descrição
? [comando]	O mesmo que help.

Exemplos:

```
ftp sunsite.unc. edu

ftp -niv ftp.linux.org < comandos.txt &
```

lpr

Envia arquivos para afila de impressão remota. Caso nenhuma impressora seja especificada, Ipr usará a impressora padrão do sistema.

Sintaxe:

```
lpr [-hmrs][-J jobJ[-Pprinter][ -n& ][arquivo ...]
```

Parâmetro	Descrição
-h	Não imprime a página de identificação da impressão.
-m	Envia mail ao final da impressão.
-r	Remove o arquivo após a impressão (com a opção -s).
-s	Usa links simbólicos em vez de copiar os arquivos para o diretório de spool.
-J job	Nome a ser impresso na página de identificação. O nome do primeiro arquivo é o default.
-Pprinter	Envia um arquivo para impressão na impressora cujo nome seja printer.
-n#	Imprime # cópias de cada arquivo.

Exemplo:

```
lpr -Plaser3 -J game readme.txt install.txt
```

Comandos de Rede

ftp

Transfere arquivos entre máquinas da rede.

Sintaxe:

```
ftp [-ginv] [maquina]
```

Parâmetro	Descrição
-g	Desabilita o “globbing” dos nomes de arquivos, habilitado por default. Veja o comando glob abaixo.
-i	Não pede confirmação antes da transferência de múltiplos arquivos. Veja o comandoprompt abaixo.
-n	Desabilita o “auto-login”. Veja o comando open abaixo.
-v	Habilita a saida descritiva. Default se a entrada padrão estiver associada a um terminal.

Comandos:

Comando	Descrição
![cmd [args]]	Invoca uma shell na máquina local e executa o comando cmd.
append loc[rem]	Concatena o arquivo *loc* na máquina local ao arquivo remoto *rem*.
ascii	Ativa o modo de transferência adequado para a transferência de arquivos texto.
bell	Soa um alarme a cada transferência de arquivo completada.
binary	Ativa o modo de transferência adequado a arquivos binários.
bye	Encerra a sessão como servidor remoto e saido programa ftp. Um caractere EOF (geralmente ˆD) tem o mesmo efeito.
case	Converte o nome dos arquivos transferidos para minúsculas.
cd dir	Muda diretório de trabalho na máquina remota para dir.
cdup	Muda para o diretório pai do diretório atual no servidor.
chmod modo arq	Muda o modo de acesso do arquivo remoto arq para aquele especificado por modo.
close	Termina a sessão com o servidor remoto e retorna ao interpretador de comandos.
cr	Liga/desliga a filtragem de caracteres CR durante transferências do tipo ascii. Ativada é o default.
delete arq	Remove o arquivo arq da máquina remota.
dir [dir] [arq]	Mostra o conteúdo do diretório remoto dir, opcionalmente colocando a saída no arquivo local arq.
disconnect	O mesmo que close.
get arq [loc]	Transfere o arquivo arq da máquina remota para a máquina local, renomeando-o para loc caso esse parâmetro seja fornecido.
glob	Liga/desliga a expansão de nomes de arquivos para os comandos mget , mput e mdelete. O default é a expansão de nomes ativada.
hash	Mostra um caractere # (hash) para cada bloco de dados recebido (1Kbyte).
help [cmd]	Mostra uma mensagem informativa sobre o comando cmd. Se nenhum comando for especificado, lista todos os comandos disponiveis.
idle[seg]	Ajusta o relógio de inatividade no servidor remoto para seg segundos. Se seg não for especificado, mostra o valor atual do contador.
lcd [dir]	Muda o diretório de trabalho na máquina local. Se dir não for especificado, o diretório HOME do usuário será usado.
ls [dir] [arq]	Similar a dir, porém inclui informações dependentes de sistema fornecidas pelo servidor.
mdelete [arqs]	Similar a delete para múltiplos arquivos.
mdir arqs loc	Gera uma listagem de arqs no arquivo local loc.
mget arqs	Similar a get para multiplos arquivos.

Comando	Descrição
mkdir dir	Cria um diretório na máquina remota.
mls arqs loc	Gera listagem similar à de ls para os arquivos dados por arqs e grava sua saída no arquivo local loc.
modtime arq	Mostra a data de modificação do arquivo remoto arq.
mput arqs	Copia os arquivos dados por arqs para o diretório de trabalho corrente na máquina remota.
newer arq	Copia um arquivo somente se a versão remota for mais recente do que a versão local, cujo nome será dado por arq.
open host[porta]	Abre uma conexão com o host na porta especificada. Se a opção auto-login estiver habilitada, ftp tentará logar no host remoto.
prompt	Liga ou desliga o modo interativo, onde ftp pede confirmação antes de transferir ou eliminar múltiplos arquivos.
put arq [rem]	Armazena um arquivo na máquina remota, opcionalmente com o nome dado por rem.
pwd	Imprime o diretório corrente da máquina remota.
quit	O mesmo que bye.
recv rem [loc]	O mesmo que get.
remotehelp [cmd]	Pede informações sobre os comandos do servidor ftp remoto.
rename old new	Altera o nome do arquivo remoto chamado old, chamando-o de new.
reset	Sincroniza o cliente com o servidor de ftp.
rmdir dir	Remove do servidor o diretório especificado.
runique	Evita a sobreposição de arquivos, concatenando a seu nome um sufixo numérico.
sendarq loc [rem]	O mesmo que put.
status	Mostra o estado atual do ftp.
sunique	O equivalente a runique para o sistema remoto.
type [tipo]	Muda o "tipo de representação" para tipo. Os tipos ascii e binary (ou image) são válidos, sendo ascii o default.
user nome	Identifica o usuário nome com o servidor ftp. Se uma senha de acesso for necessária o servidor irá pedi-la.
verbose	Aciona o modo detalhado de apresentação. Default se os comandos estiverem sendo digitados em um terminal.
? [comando]	O mesmo que help.

Exemplos:

```
ftp sunsite.unc. edu

ftp -niv ftp.linux.org < comandos.txt &
```

mail

Sistema de processamento de correspondência eletrônica.

Sintaxe:

```
mail [-ilnv] [-s ass] [-c lista] [-b lista] end...
```

```
mail [-ilnNv] -f [nome]

mail [-ilnNv] [-u user]
```

Parâmetro	Descrição
-v	Modo detalhado (verbose). Os detalhes da entrega de mail serão mostrados no terminal do usuário.
-i	Ignora sinais de interrupção do terminal. Útilem linhas com ruido.
-l	Força o modo interativo mesmo quando a entrada não for o terminal.
-n	Inibe a leitura do arquivo de inicialização global do sistema de mail.
-N	Inibe a exibição inicial dos cabeçalhos das mensagens.
-s ass	Especifica o assunto (subject) em linha de comando. Apenas o argumento seguinte à opção -S será considerado. Assuntos contendo espaços deverão aparecer entre aspas.
-c lista	Envia uma copia da mensagem para os endereços especificados em lista.
-b lista	Envia uma cópia da mensagem para os endereços especificados em lista, sem que os outros destinatários saibam.
-f[nome]	Lê o conteúdo do arquivo nome para processamento (ou de mbox, caso nenhum nome de arquivo tenha sido especificado).
-u nome	Lê o mailbox do usuário especificado por nome.

Utilização: Se invocado sem parâmetros, mail irá mostrar o conteúdo do mailbox e esperar por comandos do usuário, que podem ser:

Comando	Descrição
?	Lista os comandos disponíveis de forma suscinta.
! cmd	Abre uma shell e executa o comando cmd.
R	Reply. Responde ao remetente da mensagem corrente.
a[args]	Sem argumentos, mostra todos os aliases definidos; com um argumento, mostra apenas um alias; com mais de um argumento, define o alias com o nome dado pelo primeiro argumento.
c [dir]	Muda o diretório corrente para aquele especificado pordir ou para o diretório HOME do usuário caso não haja argumentos.
d[msgs]	Marca as mensagens cujos identificadores estejam na lista msgs para eliminação. Se nenhuma mensagem tiver sido especificada, a mensagem corrente será marcada.
e[msgs]	Edita todas as mensagens dadas por msgs, passando uma a uma para o editor.
x	Sai abandonando alteragées feitas no mailbox ou no arquivo que estiver sendo processado.
m addr	Envia mail para todos os enderegos especificados

Comando	Descrição
	na lista addr.
n	Passa para a próxima mensagem; equivale a "+" ou CR (carriage return).
q	Termina a execução do programa e efetiva todas as alterações feitas pelo usuário (eliminando mensagens marcadas, etc.).
r	Reply. Envia mensagem ao remetente e demais destinatários de uma mensagem.
s list f	ile Toma uma lista de mensagens e as concatena ao arquivo cujo nome seja o dado por file, na ordem especificada.
sh	Invoca uma shell interativa.
u	Undelete. Remove a marca de eliminação de uma mensagem.
w list f	ile Idem a s, exceto pelo fato dos cabeçalhos das mensagens serem ignorados.
z	Avança uma página na lista de mensagens.
z-	Retrocede uma pagina na lista de mensagens.
Edição	Quando usamos os comandosm, rou R, ou especificamos um destinatário em linha de comando, entramos no modo de edição de mensagens. Aqui utilizamos os chamados "tilde escapes", comandos especiais que devem ser digitados no início de uma linha para serem reconhecidos. Se o caractere "=" for necessário no início da linha da mensagem, este deve ser digitado duas vezes.
~!cmd	Abre uma shell e executa o comando cmd, retornando à mensagem em edição.
~cnomes	Acrescenta os nomes da lista nomes à lista de receptores das cópias da mensagem.
~d	Insere o arquivo dead.letter na mensagem corrente.
~e	Invoca o editor de textos e edita a mensagem que estiver sendo composta. Ao final da edição pode-se continuar acrescentando texto à mensagem.
~f msgs	Insere as mensagens especificadas na mensagem em edição.
~h	Edita os campos do cabeçalho (header) da mensagem.
~m msgs	Insere as mensagens da listamsgs endentadas na mensagem em edição.
~q	Aborta a mensagem em edição, guardando seu conteúdo no arquivo dead.letter.
~r file	Insere o arquivo file na mensagem em edição.
~s strin	g Muda o assunto (subject) da mensagem para string.
-w nome	Grava o conteúdo da mensagem em edição no arquivo nome.
~|cmd	Passa a mensagem em edição para o programa cmd através de um pipe.

Obs: Dentre todas as ferramentas que compõe um sistema UNIX, os agentes de mail são, sem dúvida, os mais populares eos que apresentam maior variedade de implementação. O agente documentado aqui é o popular mailx, presente na maioria dos sistemas, tiver sido

escolhido pela generalidade almejada por este guia. As listas de mensagens requeridas por alguns comandos são listas de números ou intervalos separados por vírgulas. Intervalos têm a forma X-Y (leia de X a Y).

Exemplos:

```
cat arq1.txt arq2.txt | mail -s textos macan

mail macan -c wada, tulio, mauronr -b flexa

mail -s "falha no backup" root
```

rcp

Copia arquivos de uma maquina remota.

Sintaxe:

```
rcp [-pr] <fonte> <destino>
```

Parâmetro	Descrição
-p	Mantém os atributos do arquivo original, como, por exemplo, datas de acesso e modificação e permissões de acesso.
-r	Copia também o conteúdo dos subdiretórios abaixo do diretório especificado.
fonte	Especifica um arquivo ou diretório em máquina remota, podendo assumir duas formas: maquina:path ou ainda usuario@maquina:path (o ultimo, no caso do arquivo pertencer a outro usuario).
destino	Similar a fonte. Quando multiplos arquivos são especificados por fonte, o destino precisa necessariamente ser um diretório.

Obs: O usuário e a máquina destino precisam ter permissão de acesso via rede nas máquinas fonte. Consulte o manual do sistema para maiores informações.

Exemplo:

```
# Copia o arquivo teste.pl, residente em /bin na máquina
# chamada cradle para o diretório corrente.

rcp -p leonardo@cradle:/bin/teste.pl .

# Copia todo o diretório /home/macan/bin da maquina chamada
# dracula para o diretorio corrente.

rcp -r dracula:/home/macan/bin/ .
```

rsh

Executa um comando no host especificado. Se invocado sem a especificação de um comando, executa uma shell interativa no host remoto.

Sintaxe:

```
rsh [-Iuser] host [comando]
```

Parâmetro	Descrição
-l user	Permite executar a shell como outro usuario no host remoto (por default, o nome do usuário remoto será o mesmo do local).

Exemplo:

```
#outro.host.exemplo e dracula sao nomes
#ficticios de hosts.

rsh -l macan outro.host.exemplo
rsh dracula
```

ruptime

Mostra o status de cada máquina da rede local, obtendo seus dados de pacotes difundidos pelas máquinas de tempos em tempos (geralmente entre 1 e 3 minutos).

Sintaxe:

```
ruptime [-alrtu]
```

Parâmetro	Descrição
-a	Conta até mesmo os usuários que estejam inativos há mais de uma hora.
-l	Ordena a saída pela carga média de cada host.
-r	Inverte a ordenação da saída.
-t	Ordena a saída por tempo de atividade do sistema (uptime).
-u	Ordena a saida pelo numero de usuarios.

Obs: Uma máquina é considerada inativa (down) quando a mesma não difunde informações sobre seu estado há algum tempo (geralmente entre 5 e 11 minutos, dependendo do sistema utilizado).

Exemplo:

```
ruptime -1
```

rup

Mostra informações sobre o status de uma máquina da rede. Similar a uptime, para máquinas da rede local.

Sintaxe:

```
rup [máquina ...]
```

Obs: Se não forem fornecidos parâmetros, rup fará um pedido do status por difusão (broadcast) a todas as máquinas da rede local, mostrando os resultados na ordem em que asrespostas forem recebidas.

rusers

Mostra quem está usando as máquinas da rede.

Sintaxe:

```
rusers [-al] [máquina ...]
```

Parâmetro	Descrição
-a	Mostra informação de todas as máquinas que responderam, mesmo que não haja nenhum usuário conectado.
-l	Seleciona o formato longo de saída.

Obs: Se nenhuma máquina tiver sido especificada na linha de comando, rusers mostrará informações de todas as máquinas da rede local.

rwall

Envia mensagem a todos usuarios de uma maquina. Sintaxe:

```
rwall [máquina]
```

Parâmetro	Descrição
máquina	O nome da máquina para a qual se deseja enviar a mensagem.

Obs: rwall lerá a mensagem da entrada padrão (stdin) até encontrar o caractere EOF (geralmente CTRL-D).

rwho

Mostra quem está usando as máquinas da rede local e o que estão fazendo.

Sintaxe:

```
rwho [-a]
```

Parâmetro	Descrição
-a	Mostra inclusive usuários com tempo de inatividade maior do que uma hora.

telnet

Abre um canal de comunicação entre duas máquinas através do protocolo TELNET.

Sintaxe:

```
telnet [ host [ port] ]
```

Parâmetro	Descrição
host	Identifica a máquina remota com a qual se quer estabelecer a conexão.

Parâmetro	Descrição
port	Identifica a porta através da qual se estabelecerá a conexão entre as duas máquinas.

Obs: Se não forem fornecidos parâmetros, telnet entrará no modo de comando interativo. Para alternar para o modo interativo durante uma sessão usa-se o caractere de escape CTRL-] (^D).

Comandos:

Comando	Descrição
open host [port]	Abre uma conexão com host através da porta especificada.
close	Encerra uma conexão TELNET e retorna ao modo de comando.
quit	Fecha qualquer sessão aberta e termina a execução de telnet. Um EOF (geralmente ^D) no modo de comando tem o mesmo efeito.
z	Suspende telnet. Este comando só funciona se a shell do usuário implementar mecanismos de controle de processos.
mode modo	Seleciona atransmissão dos dados por caractere ou linha; character ou line, respectivamente.
status	Mostra o estado atual de telnet.
display [arg...]	Mostra o valor de um (ou todos) argumento(s) dos comandos set e toggle (veja adiante).
?[comando]	Mostra informações de ajuda sobre comando. Se invocado sem argumentos, listaos comandos válidos.
![comando]	Invoca uma shelle executacomando caso este tenha sido especificado. Disponível em algumas versões de telnet.
send argumentos	Envia uma sequência especial de caracteres para o host remoto. Os seguintes argumentos podem ser especificados:
escape	Envia o caractere de escape (inicialmente “]).
synch	Envia a sequência de sincronismo do protocolo TELNET, o host remoto descarta todos os caracteres recebidos que ainda não foram interpretados.
brk	Envia a sequência de BRK (break).
ip	Envia a sequência IP (Interrupt Process), que faz com que o host remoto interrompa a execução do processo que estiver sendo executado.
ao	Envia a sequência AO (Abort Output), que faz o host remoto descartar toda a saída do host remoto para o terminal do usuário.
ayt	Envia a sequência AYT (Are You There?) para o host remoto, que pode escolher responder ou não.
ec	Envia a sequência EC (Erase Character), que faz com que o host remoto cancele o último caractere recebido.
el	Envia a sequência EL (Erase Line), que faz

Comando	Descrição
	com que o host remoto descarte a linha que estiver sendo digitada.
ga	Envia a sequência GA (GO AHEAD). Sem significado para o host remoto.
nop	Envia a sequência NOP (No Operation).
?	Mostra informações sobre o comando send.
set variável valor	Ajusta o valor de uma ou mais variáveis telnet. Os valores especiaison e off ligam ou desligam a função associada a uma variável (veja toggle abaixo).
echo	Caractere para ligar ou desligar o eco local dos caracteres digitados (Inicialmente ^E).
escape	Caractere de escape para o modo de comando (inicialmente ^]).
interrupt	Caractere para provocar interrupção do processo remoto. O default dependerá da configuração do terminal (veja stty).
quit	Caractere para provocar aborto de processo. O default dependerá da configuração do terminal (veja stty).
flushoutput	Caractere para provocar um Abort Output.
erase	Caractere usado para eliminar um caractere. O valor default dependerá da configuração do terminal (veja stty).
kill	Caractere usado para cancelar a entrada de uma linha. O default dependerá da configuração do terminal (veja stty).
eof	Caractere para demarcar o fim da entrada. O default dependerá da configuração do terminal (veja stty).
toggle args...	Inverte o valor associado a uma (ou várias) variável(eis) booleana(s) (veja set acima).
localchars	Liga/desliga o reconhecimento local de certos caracteres de controle.
autoflush	Liga/desliga o descarte da saída quando enviando caracteres de interrupção.
autosynch	Liga/desliga o envio automático de caracteres de interrupção no modo “urgente” de transmissão.
crmod	Liga/desligao mapeamento local de caracteres de retorno de carro (CR) recebidos.
options	Liga/desliga a visualização do processamento de opções (debugging).
netdata	Liga/desliga a impressão em hexadecimal de dados provenientes da rede (debugging).
?	Mostra informações sobre o comandotoggle.

Exemplo:

```
telnet localhost
```

whois

Pesquisa um identificador, tal como nomes, handles ou organizações em um banco de registros na internet.

Sintaxe:

```
whois [-h host] identificador
```

Parâmetro	Descrição
-h host	Indica o host ao qual se deve fazer a consulta.

Exemplos:

```
whois pigs.com
```

```
whois -h whois.internic.net Silva
```

Obs: O servidor central de whois da internet, para os "top level domains" (.com .org .mil .br , etc) é a máquina whois.internic.net.

Correspondência DOS-UNIX

Esta tabela demonstra a correspondéncia entre alguns comandos do DOS e os comandos do Unix e nao uma equivaléncia. Os comandos do Unix sao, em geral, mais complexos e poderosos do que os correspondentes no MS-DOS.

Table 110: Tabela simplificada de Correspondência DOS-UNIX

Comando do DOS	Correspondente UNIX
ATTRIB	chmod
CD	cd
CLS	clear
COMP	diff
COPY	cp
DATE	date
DEL	rm
DELTREE	rm -rf
DIR	ls
ECHO	echo
EDIT	vi
EDLIN	ed
HELP	man
MD	mkdir
MORE	more
MOVE	mv
PRINT	lp , lpr
RD	rmdir
REN	mv
SORT	sort
TIME	date
TYPE	cat
VER	uname -a
XCOPY /S	cp -r

Leitura Recomendada

- UNIX System Programming, Keith Haviland e Ben Salama, Addison- Wesley Publishing.

- The UNIX System, S. R. Bourne, Addison-Wesley Publishing.
- The UNIX System V Environment, S. R. Bourne, Addison-Wesley Publishing.
- Modern Operating Systems, Andrew S. Tanembaum, Prentice Hall.
- Operating systems: Design and Implementation, Andrew S. Tanembaum, Prentice Hall.
- Computer Networks, Andrew S. Tanembaum, Prentice Hall.
- The C Programming Language, B. W. Kernighan e D. M. Ritchie, Prentice Hall .
- Internetworking with TCP/IP: Principles, Protocols and Architecture, D. Comer, Prentice Hall.

Obs: Parte destes titulos esta disponivel em portugués, consulte a livraria mais proxima.

WWW

Pagina oficial da lista de discussão linux-br, a maior e mais antiga do http://www.openline.com.br/linux-br

Lista de Usuários Avançados de Linux (lista fechada, inscrição median http://www.mondotech.com/lual

Site oficial da linux international. http://www.linux.org

Site oficial da freebsd.org no Brasil. http://www.br.freebsd.org

O Site do Servidor X Free. http://www.xfree86.org

Free Software Foundation. http://www.gnu.org

Debian GNU/LINUX. http://www.debian.org

Home page da lista Linux-BR. http://www.openline.com.br/linux-br

Unix World on-line Magazine. http://unixworld.com/unixworld

Unix Guru Universe. http://www.ugu.com

Mirror brasileiro oficial da Sunsite. http://sunsite.unicamp.br

Sunsite Archives. http://sunsite.unc.edu

FTP Sites

Mirror dos pacotes GNU (Free Software Foundation). ftp://ftp.unicamp.br/pub/software/gnu

GNU Software. ftp://ftp.unicamp.br/pub/gnu

Arquivos da DICAS-L. ftp://ftp.unicamp.br/pub/dicas-1

Internet RFC (Request For Comments). ftp://ftp.unicamp.br/pub/documents/rfc/

Listas de Discussão

listproc@netway.unicamp.br Dicas-l, uma lista moderada com dicas periódicas valiosas para administradores. Para se in-

screver, envie para o endereço acima a mensagem: subscribe dicas-l nome completo.

listproc@listas.ansp.br Este servidor disponibiliza duas listas nacionais de grande importância, linux-br e redes-l. Para se inscrever envie uma mensagem com os seguintes comandos: subscribe linux-br nome completo. subscribe redes-| nome completo.

petidomo@igm.unicamp.br Este serrvidor distribui a lista FreeBSD-I. Para se inscrever, envie o comando add FreeBSD no corpo do e-mail.

Software Livre

Há alguns anos a Free Software Foundation (FSF) vem produzindo software com qualidade no minimo equivalente à dos similares comerciais; o software é desenvolvido por voluntários de várias partes do mundo e garante a liberdade do usuário de redistribui-lo e modificá-lo, pois o código fonte está sempre disponível, o que facilita a adaptação do código a várias plataformas.

Você pode encontrar seu programa "free" preferido, compilado para várias arquiteturas e se não houver versão disponível para seu sistema, você poderá tomar a iniciativa de portá-lo, sem depender do interesse comercial de uma grande companhia.

A iniciativa de produção de software livre*, patrocinada ou não pela FSF, trouxe grandes contribuições ao público, entre elas os sistemas Unix free, destacando-se Linux e FreeBSD.

Para saber mais sobre "free software" consulte as páginas da seção Internet deste guia.

* A tradução adequada de "free" para este caso é "livre" e não "grátis", pois o usuário tem todaliberdade com o sofware. Os direitos de cópia são reservados ao autor, que os cede ao interesse público. O software é pago mediante doação para o autor ou instituição que detiver os direitos do software.

A Respeito do Autor

Eduardo Marcel Maçan, natural de Santa Mariana - PR, ingressou no curso de Engenharia de Computação da Universidade Estadual de Campinas (UNICAMP) em 1992, tendo trabalhado desde então com administração e desenvolvimento para o sistema Unix.

Fundador da lista de discussão linux-br, atualmente presta consultoria em projeto, implantação e integração de redes, Intranet e Internetworking, suporte Unix, desenvolvimento de material didático e cursos para a Mondo Technologies.

Sobre Este Guia

Escrever um guia de consulta rápida para UNIX é uma tarefa no mínimo desafiadora. Há inúmeras versões de UNIX, comerciais ou

não, cada uma sendo constantemente modificada de forma independente, sem muita preocupação de se adotar (ou respeitar) padrões de desenvolvimento.

Esta desordem aliada à quantidade assombrosa de informação envolvida nos comandos do sistema resulta na necessidade de se estabelecer critérios quanto ao que é realmente útil em um guia de consulta rápida e a quem o mesmo se destina.

Este guia foi escrito baseando-se na experiência do autor com vários "sabores" de UNIX, a saber: Linux, FreeBSD, SunOS, Solaris, HP-UX, AIX e OSF/1 e nos manuais dos quatro primeiros. Visamos sempre a documentação do que era comum a todas as versões de cada comando, delimitando desta forma um núcleo de informação genérica o suficiente para que pudéssemos afirmar que este guia será útil em qualquer versão de sistema e para toda classe de usuários.

Os iniciantes têm nesta seleção de comandos um volume de informações bem além do essencial, a nível de usuário, e uma tabela de correspondência de comandos que facilita muito a migração a partir de um ambiente DOS / Windows, tornando possível o ingresso imediato no mundo de um dos mais completos e fascinantes sistemas operacionais já desenvolvidos.

O texto deste guia foi inteiramente composto e editado usando free software. O autor utilizou o sistema operacional Linux (Debian/GNU), XFree86 (sistema de janelas) e o editor de textos GNU Emacs (ainda estou tentando convencer a editora a usar LaTeX para a formatação. :-)).

O autor deste guia pode ser contatado pela Internet, através do endereço:

E-Mail: macan@novatec1.com

Sugestões e comentários sobre este guia são sempre bem-vindos.

Convenção Utilizada neste Guia

Convenção	Significado
[texto]	Texto é opcional.
<texto>	Texto é obrigatório.
texto...	Texto pode ocorrer mais de uma vez.
texto1 | texto2	Texto1 etexto2 são alternativas mutuamente exclusivas.

Licença