

Comandos

- cut
- sed
- grep
- tr
- sort
- paste

cut

Finalidade:

O comando cut pode ser usado para mostrar apenas seções específicas de um arquivo de texto ou da saída de outros comandos. Ele lê o



Sintaxe Padrão:

```
$ cut OPÇÃO ... [ ARQUIVO ] ...
```

Parametros(se existir):

- -b , -bytes = LIST Selecione apenas os bytes de cada linha, conforme especificado em LIST . LIST especifica um byte, um conjunto de bytes ou um intervalo de bytes; consulte Especificando a lista abaixo.
- -c , -characters = LIST Selecione apenas os caracteres de cada linha, conforme especificado em LIST . LIST especifica um caractere, um conjunto de * caracteres ou um intervalo de caracteres; consulte Especificando a lista abaixo.
- -d , -delimiter = DELIM use o caractere DELIM em vez de uma guia para o delimitador de campo .
- -f , -fields = LIST selecione apenas esses campos em cada linha; imprima também qualquer linha que não contenha caracteres delimitadores, a menos que * a opção -s seja especificada. LIST especifica um campo, um conjunto de campos ou um intervalo de campos; consulte Especificando a lista abaixo.
- -n Esta opção é ignorada, mas está incluída por motivos de compatibilidade.
- -complement complementa o conjunto de bytes, caracteres ou campos selecionados.
- -s , -only-delimited não imprima linhas que não contenham delimitadores.
- -output-delimiter=STRING use STRING como a string delimitadora de saída. O padrão é usar o delimitador de entrada.
- -help Exiba uma mensagem de ajuda e saia.
- -version informações da versão de saída e saia.

Script:

```
echo "Executando script do cut - Mostrar primeira coluna da passwd";
echo;
cut -d: -f1 /etc/passwd
echo "Executando script do cut - Pega a segunda coluna do arquivo anotações.md";
echo;
cut -f 2 -d ' ' ./anotações.md
echo;
```

sed

Finalidade:

O Sed é um editor de textos não interativo. Vem do inglês [S]tream [E]ditor, ou seja, editor de fluxos de texto.

Sintaxe Padrão:

```
$ sed [OPÇÃO]... {script-apenas-se-for-único} [arquivo-entrada]...
```

Parametros(se existir):

- -n , --quiet , --silent suprime a impressão automática do buffer padrão --debug anota a execução do programa
- -e script , --expression=script adiciona o script aos comandos a serem executados
- -f arquivo-script , --file=arquivo-script adiciona o conteúdo do arquivo-script aos comandos a serem executados --follow-symlinks segue links simbólicos ao

- editar o próprio arquivo original
- -i[SUFIXO], --in-place[=SUFIXO] edita o arquivo original (faz backup se usado SUFIXO)
- -l N, --line-length=N determina comprimento da quebra de linha para comando 'l'
- --posix desativa todas as extensões GNU.
- -E, -r, --regext-extended usa sintaxe moderna de expressões regulares, sem escapes. (para portabilidade, use POSIX -E).
- -s, --separate considera arquivos como entidades separadas, e não como um longo e único fluxo de dados.
- --sandbox opera em modo sandbox (desabilita comandos e/r/w).
- -u, --unbuffered carrega uma quantidade mínima de dados dos arquivos de entrada e descarrega os buffers de saída com mais frequência
- -Z, --null-data separa linhas por caracteres NULOs
- --help mostra esta ajuda e sai
- --version mostra informações sobre a versão e sai

Se nenhuma opção -e, --expression, -f, ou --file é dada, o primeiro argumento que não seja uma opção é considerado como o script sed a ser interpretado. Todos os argumentos restantes são considerados como nomes de arquivos de entrada. Caso nenhum arquivo de entrada seja especificado, então a entrada padrão será lida.

Script:

```
echo;
echo "Executando script do sed";
sed -n '16p' anotações.md
```

grep

Finalidade:

Imprimir linhas que correspondem a uma expressão regular.

Sintaxe Padrão:

```
$ grep [ OPÇÕES ] PADRÃO [ ARQUIVO ... ]
```

Parametros(se existir):

- -c Conta quantas vezes apareceu a string que esta pesquisando
- -v Mostra na tela "tudo" menos onde houver a ocorrência da string pesquisada
- -I Realiza uma busca pela string ignorando o case, sendo case-insensitive
- -o Ira mostrar na tela apenas as ocorrências da string pesquisada ignorando o resto
- -n Ira mostrar na tela na primeira coluna a linha onde encontrou a string pesquisada
- -B Numero de linhas a serem impressas antes da linha que contem a string pesquisada [BEFORE]
- -A Numero de linhas a serem impressas na tela depois da encontrar a linha com a string [AFTER]
- -C Quantidade de linhas antes e depois da linha que contem a string [CONTEXT]
- -q Ira procurar pela string informada, porém estará em modo silencioso, nada sera impresso na tela, porém caso encontre o comando encerra com 0, caso não encontre nada será 1
- -E Estende o uso de Regex no padrão e combinação, usando logica AND e OR por exemplo
- -f Um arquivo com combinações de padrões com Regex, podendo usar varias combinações
- -I Mostra somente o nome do arquivo onde foi encontrado a string pesquisada
- -L Semelhante ao -v, porque mostra apenas os arquivo que não contem a string informada
- -h Pesquisa varios arquivos, diretórios se com -r mas não mostra o nome dos arquivos
- -r Ira realizar uma pesquisa recursiva em todos os diretórios a partir do informado
- --color Deve-se passar o parâmetro 'never' caso não queira que a saída marque com cor a string ou 'auto' e 'always' para operar conforme necessite. Pode mudar a cor alterando GREP_COLOR, GREP_COLORS no environment

Script:

```
echo "Executando scripts do grep";
echo;
echo "Anotações.md";
grep --color "diferente" ./anotações.md
echo;
echo "/etc/passwd";
grep --color "gnome" /etc/passwd
```

tr

Finalidade:

Sua função principal é substituir (traduzir) o conteúdo de uma string (texto) recebido via entrada padrão (STDIN) de um formato para outro, ou ainda excluir caracteres.

Sintaxe Padrão:

```
$ tr [opções] < conjunto1 > [conjunto2]
```

Parametros(se existir):

- -c, -complement Usar o complemento do conjunto1 (conjunto inverso)
- -d, -delete Excluir caracteres do conjunto1, sem traduzi-los
- -s, -squeeze-repeats Substituir cada sequência de um caractere repetido listado no conjunto1 com uma única ocorrência do caractere
- -t, -truncate-set1 Truncar o conjunto1 no mesmo comprimento do conjunto2
- -help Mostrar a ajuda do comando e sair
- -version Mostrar informações de versão e sair.

Script:

```
echo "Executando scripts do tr";  
echo;  
echo "Anotações.md";  
tr 'é' '=' < ./anotações.md > teste2  
echo;
```

sort

Finalidade:

sort classifica o conteúdo de um arquivo de texto, linha por linha.

Sintaxe Padrão:

```
$ sort [ OPÇÃO ] ... [ ARQUIVO ] ...
```

Parametros(se existir):

- -b, -ignore-líderes-em branco Ignore os espaços em branco à esquerda.
- -d, - ordem do dicionário Considere apenas espaços em branco e caracteres alfanuméricos .
- -f, -ignore-case Dobre letras minúsculas para caracteres maiúsculos.
- -g, -geral-numérica geral Compare de acordo com o valor numérico geral.
- -i, -ignore-nonprinting Considere apenas caracteres imprimíveis.
- -M, -month-sort Compare (desconhecido) < JAN ' <... < DEC '.
- -h, -human-numeric-sort Compare números legíveis por humanos (por exemplo, " 2K ", " 1G ").
- -n, -numeric-sort Compare de acordo com o valor numérico da string .
- -R, - tipo aleatório Classifique por hash aleatório de chaves.
- -random-source = ARQUIVO Obter bytes aleatórios de FILE .
- -r, -reverse Inverta o resultado das comparações.
- -sort = PALAVRA Classifique de acordo com WORD : numérico geral -g , numérico humano -h , mês -M , numérico -n , aleatório -R , versão -V .
- -V, -version-sort Tipo natural de (versão) números dentro do texto.

Script:

```
echo "Executando scripts do sort";  
touch frutas.txt  
echo 'maças  
laranjas  
peras  
kiwis  
bananas' > frutas.txt  
sort frutas.txt > saida.txt  
echo;  
echo "Mostrando arquivos ordenados por ordem alfabetica";  
echo "----";  
cat saida.txt  
echo;
```

paste

Finalidade:

Sintaxe Padrão:

```
$ paste [ OPÇÃO ] ... [ ARQUIVO ] ...
```

Parametros(se existir):

- -d , -delimitadores = LIST Reutilize caracteres da lista em vez de guias.
- -s , -serial Cole um arquivo de cada vez em vez de em paralelo.
- -Socorro Exiba uma mensagem de ajuda e saia.
- -versão Exiba informações da versão e saia.

Script:

```
echo "Executando script do paste";  
echo "Comparação dos arquivos classificados anteriormente";  
echo;  
paste frutas.txt saida.txt
```