

fedora 



 ubuntu

 Mandriva

Curso de Formação LPIC-1

Exame 101



Curso Linux: formação

- › Processar texto usando filtros
 - › Procurar dentro de ficheiros
 - › The Stream Editor (SED)
 - › Processamento de texto com AWK
 - › Substituir caracteres
 - › Utilitário para remover duplicações
 - › Extrair colunas de texto
 - › Juntar vários ficheiros
 - › Formatar texto

Processamento de texto

Procurar dentro de ficheiros

- Comando *grep* (*Global regular expression print*)
- Pode encontrar quase qualquer string ou frase num ficheiro de texto

```
grep -options pattern ficheiro
```

- Algumas opções
 - -c – Mostra uma contagem das ocorrências, sem texto no écran
 - -H – Mostra o nome do ficheiro onde encontrou uma ocorrência
 - -h – Não mostra o nome do ficheiro onde encontrou a ocorrência
 - -i – Não liga a capitalização
 - -l – Mostra o nome do ficheiro onde encontrou a ocorrência, mas não mostra texto
 - -L – Mostra o nome do ficheiro ONDE NÃO encontrou ocorrências
 - -w – Mostra apenas linhas que contenham a string a procurar como uma palavra, e não como parte de uma palavra
 - -r – Processa todas as directorias especificadas e ficheiros incluídos nelas
 - -v – Mostra as linhas num ficheiro que não tenham nenhuma ocorrência da string. É o comportamento oposto

Processamento de texto

Procurar dentro de ficheiros

```
who | grep aluno
```

```
find / -name readme -exec grep -iw kernel {} \;
```

```
find / -name readme -exec grep -iw kernel {} \;
```

```
grep -v :x: /etc/passwd
```

```
grep -ir idiot /usr/src/linux/Documentation | tail
```

Processamento de texto

Procurar dentro de ficheiros usando expressões regulares

- › Caracteres especiais
 - › . - Um ponto corresponde a qualquer caracter único e força a que o caracter tenha que existir (a.v – 3 caracteres)
 - › ? - Exclamação corresponde a qualquer caracter opcional e só é visto uma única vez
 - › * - asterisco corresponde desde zero a muitos caracteres
 - › + - o item tem que corresponder uma vez e pode corresponder várias vezes
 - › {n} – significa que o item vai corresponder *n* vezes
 - › {n,} - significa que o item vai corresponder *n* ou mais vezes
 - › {n,m} – significa que o item vai corresponder de *n* a *m* vezes

Processamento de texto

Procurar dentro de ficheiros usando expressões regulares

- Procurar todas as palavras começadas com co mas que o terceiro caracter não seja r:
 - `grep "\<co[^r]" watch.txt`
- `\<` significa que a palavra começa com aquelas letras
- `[^r]` significa encontrar todos os caracteres menos aquele
- No vi, `^` significa encontrar no inicio da palavra, mas dentro de `[]` exclui aquele caracter
- Para procurar palavras que terminem com determinadas letras:
 - `grep "ta\>" watch.txt`

- Pesquisas permitidas pelo grep incluem:
 - `conta` – procura pela palavra exacta
 - `^conta` – procura pela palavra conta no inicio de cada linha
 - `conta$` - procura pela palavra conta no final de cada linha
 - `[cC]` – procura pelas palavras conta e Conta
 - `co[nr]ta` – procura pelas palavras conta e corta
 - `^.....$` - procura por palavras que contenham seis caracteres
 - `[cC][oO]nt[aA]` – procura por Conta, COnTa, contA ou combinações entre elas

Processamento de texto

sed – The Streaming Editor

- › Comando *sed*
- › Processa e realiza acções em texto
- › Poderoso, mas ao mesmo tempo complicado de usar

```
sed -option action/regexp/replacement/flag ficheiro
```

- › `sed s/Terra/lua/ ficheiro` – substitui todas as ocorrências de dia com noite numa linha no ficheiro
- › `sed s/Terra/lua/g ficheiro` – substitui todas as ocorrências de dia com noite em todas as linhas

Processamento de texto

sed – The Streaming Editor

- › Permite várias operações numa linha, em que todas são executadas antes de passar à próxima linha.

```
sed 's/Terra/lua/g; s/lua/Linux/g' ficheiro
```

- › O uso do ; é semelhante ao poder da *bash* para executar vários comandos independentemente.
- › Todo o comando entre ' ' (peliculas) é executado no sed e não na *bash*.
- › Para usar vários comandos, usar ; ou várias vezes -e

```
sed -e s/Terra/lua/g -e s/lua/Linux/g ficheiro
```


Processamento de texto

sed – The Streaming Editor

- Permite automação, colocando os comandos mais usados dentro de um ficheiro.
- Colocar em script:
 - s/Terra/lua/g
 - s/lua/Linux/g

```
sed -f script ficheiro
```

- Suprimir linhas que não tenham sido alteradas
 - sed -n s/Terra/lua/pg ficheiro

- Substituir apenas um determinado número de linhas
 - sed -n 1s/Terra/lua/pg ficheiro
 - sed -n 4,5s/Terra/lua/pg ficheiro

Processamento de texto

Ordenar ficheiros

- Comando *sort*
- Uso de campos para ordenar ficheiros

```
sort ficheiro
```

```
sort numeros.txt
```

```
1  
11  
14  
2  
21  
34  
4  
56  
57  
6  
63  
69  
77  
96
```

```
sort -n numeros.txt
```

```
1  
2  
4  
6  
11  
14  
21  
34  
56  
57  
63  
69  
77  
96
```

```
sort nomes.txt
```

```
Augusto Martins  
Carlos Botelho  
Conceicao Sousa  
Helder Martins  
Helio Lameiras  
Ivo Goncalves  
Joao Paulo  
Luis Gaspar  
Luis Sergio  
Miguel Antunes  
Miguel Capinha  
Miguel Mendes  
Paco Fernandes  
Paulo Torres  
Pedro Antonio  
Ricardo Fontes  
Ricardo Mostardinha  
Rui Nunes  
Rui Toscano
```

Processamento de texto

Numeração de linhas

- Comando *nl*
- Útil para numerar todas as linhas de um ficheiro
- Directamente da consola

`nl ficheiro`

`nl -s <string> ficheiro`

`nl -ba -nrz nomes.txt`

`nl nomes.txt`

```
1 Pedro Antonio
2 Miguel Antunes
3 Joao Paulo

4 Carlos Botelho
5 Miguel Capinha

6 Paco Fernandes
7 Ricardo Fontes
8 Luis Gaspar
9 Ivo Goncalves

10 Helio Lameiras
11 Augusto Martins
12 Helder Martins
13 Miguel Mendes

14 Ricardo Mostardinha
15 Rui Nunes
16 Luis Sergio
17 Conceicao Sousa
18 Paulo Torres
19 Rui Toscano
```

Processamento de texto

Contagem de linhas e palavras

- Comando `wc`
 - `-l` : conta linhas de saída
 - `-w` : conta palavras
 - `-c` : conta caracteres ou bytes de saída

`who | wc -l` : forma de ver o numero de utilizadores no sistema a dado instante

Processamento de texto

Cortar colunas

- Comando *cut*
- Corta colunas de caracteres
- Permite filtrar também campos

-c : especificar quais as colunas a cortar
-d : separador
-f : campos a extrair

```
cut -d: -f 1,5,6 /etc/passwd | tail -n 5
```

```
dhcp:added by portage for dhcp:/var/lib/dhcp  
gdm:added by portage for gdm:/var/lib/gdm  
sabayon-admin:added by portage for  
sabayon:/var/lib/sabayon  
games:added by portage for gnome-games:/usr/games  
tcpdump:added by portage for tcpdump:/dev/null
```

```
cut -c 1-10 nomes2.txt  
Pedro An  
Miguel  
Joao Pau  
Carlos B  
Miguel C  
Paco Fer  
Ricardo  
Luis Ga  
Ivo Gonca  
Helio L  
Augusto  
Helder M  
Miguel  
Ricardo  
Rui Nunes  
Luis S  
Conceicao  
Paulo T  
Rui Tosc
```

Processamento de texto

Colar e juntar

- Comando *paste*
 - *Paste* não remove quaisquer dados da saída
 - Concatena os dados

```
paste ficheiro1.txt ficheiro2.txt  
Linha um do ficheiro1  Linha um do ficheiro2  
Linha dois do ficheiro1 Linha dois do ficheiro2
```

- Comando *join*
 - Remove campos redundantes dos dados
 - Usa o primeiro campo do primeiro ficheiro como uma chave
 - Tenta estabelecer uma relação com todos os ficheiros e campos dos mesmos
 - Campos separados com um separador

Processamento de texto

Colar e juntar

- Comando *join*
- Remove campos redundantes dos dados
- Usa o primeiro campo do primeiro ficheiro como uma chave
- Tenta estabelecer uma relação com todos os ficheiros e campos dos mesmos
- Campos separados com um separador

```
join utilizadores.txt localizacao.txt  
002:São Paulo:Enorme Paulo:SP  
003:Rio de Janeiro:Grande de janeiro:RJ
```

```
join -t: -o 1,2,2,3 utilizadores.txt localizacao.txt  
Belo Horizonte:MG  
São Paulo:SP  
Rio de Janeiro:RJ
```

Processamento de texto

Dados únicos

- Comando *uniq*
 - Mostra ou oculta linhas repetidas
 - -u : apenas mostra linhas unicas
 - -d : apenas mostra linhas repetidas

```
cat nomes3.txt  
Kioshi  
Hugo  
Hugo  
Adriana  
Eitch  
Danilo  
Kioshi  
Kioshi
```

```
uniq nomes3.txt  
Kioshi  
Hugo  
Adriana  
Eitch  
Danilo  
Kioshi
```

```
uniq -u nomes3.txt  
Kioshi  
Adriana  
Eitch  
Danilo
```

```
uniq -d nomes3.txt  
Hugo  
Kioshi
```


Processamento de texto

Separar ficheiros

- Comando *split*
- Separa um ficheiro em pedaços mais pequenos
 - -a # : usa o sufixo com # caracteres
 - -b # : separa os ficheiros em pedaços de # bytes
 - -C # : separa # de bytes em cada linha
 - -l # : separa com # linhas em cada parte

```
ls -lh
total 52K
-rw-r--r-- 1 feiticeir0 feiticeir0 842 2009-06-02 22:49 Dia_e_noite2.txt
-rw-r--r-- 1 feiticeir0 feiticeir0 2.7K 2009-06-02 22:33 Dia_e_noite.txt
```

```
split Dia_e_noite.txt -b 1k
ls -l
total 64
-rw-r--r-- 1 feiticeir0 feiticeir0 842 2009-06-02 22:49 Dia_e_noite2.txt
-rw-r--r-- 1 feiticeir0 feiticeir0 2665 2009-06-02 22:33 Dia_e_noite.txt
-rw-r--r-- 1 feiticeir0 feiticeir0 1024 2009-06-08 00:13 xaa
-rw-r--r-- 1 feiticeir0 feiticeir0 1024 2009-06-08 00:13 xab
-rw-r--r-- 1 feiticeir0 feiticeir0 617 2009-06-08 00:13 xac
```

Processamento de texto

Transformar ficheiros

- Comando *tr*
 - Traduz ou apaga caracteres
 - Não é usado para palavras ou frases completas – *sed*
 - O uso do `<` é obrigatório. Não consegue “alimentar-se” a si mesmo

```
tr 'a-z' 'A-Z' < nomes2.txt  
PEDRO ANTONIO  
MIGUEL ANTUNES  
JOAO PAULO  
CARLOS BOTELHO  
MIGUEL CAPINHA  
PACO FERNANDES  
RICARDO FONTES  
(...)
```

```
tr -d 'A-Z' < nomes2.txt  
edro ntonio  
iguel ntunes  
oao aulo  
arlos otelho  
iguel apinha  
aco ernandes  
icardo ontes  
uis aspar  
vo oncalves  
(...)
```

```
tr [:lower:] [:upper:] < nomes2.txt
```

Processamento de texto

Awk

- › Comando awk
 - › Pequena linguagem, estilo C, para processar texto formatado
 - › Expressões regulares
 - › Manuseamento de padrões de texto
 - › Similar ao perl

```
awk '{ print }' /etc/passwd  
awk '{ print $0 }' /etc/passwd
```

- › {} são usados para agrupar conjuntos de blocos.
- › Quando aparece *print* sozinho, todo o conteúdo é mostrado
- › \$0 representa toda a linha → *print* e *print \$0* são iguais

```
awk '{ print "" }' /etc/passwd
```

- › "" imprime uma linha em branco
- › Imprime uma linha em branco por cada linha do ficheiro.

```
awk '{ print "Linux é o maior" }' /etc/passwd
```

Processamento de texto

Awk: Múltiplos campos

- awk é poderoso no manuseamento de texto que está separado por campos
- Permite referenciar cada campo individualmente

```
awk -F" " '{ print $1 }' nomes.txt  
Pedro  
Miguel  
Joao  
Carlos  
Miguel  
Paco  
Ricardo
```

- -F é usado para especificar o separador do campo
- *print \$1* imprime o primeiro campo

```
awk -F":" '{ print $1 $3 }' /etc/passwd  
awk -F":" '{ print $1 " " $3 }' /etc/passwd
```

```
awk -F":" '{ print "username: " $1 "\t\tuid: " $3 }' /etc/passwd
```

Processamento de texto

Awk: scripts externos

- › awk pode também processar scripts externos
- › -f script.awk

```
awk -f script.awk ficheiro
```

```
BEGIN {  
    FS=":"  
}  
{ print $1 }
```

```
awk -f script.awk /etc/passwd
```

- › Separador é indicado com a variável FS
- › BEGIN é processado antes do ficheiro
- › END é processado após todas as linhas do ficheiro serem processadas

Processamento de texto

Formatar texto

- Comando *pr*
- Formata ficheiros de texto
- Adiciona:
 - Data
 - Nome do ficheiro (se existir)
 - Paginação (página de 66 linhas)

```
pr LoremIpsum.txt
```

```
2009-05-31 22:59
```

```
Lorem1.txt
```

```
Page 1
```

- `--columns=X` : formata o texto em X colunas

```
pr --columns=2 LoremIpsum.txt
```

Processamento de texto

Formatar texto

- Comando *fmt*
- Também usado para formatar texto
- Limitado a reduzir linhas longas para caber em páginas mais pequenas
- Excelente comando quando usado em conjunto com o *pr*

```
fmt -35 LoremIpsum.txt
```

- *-XX* : tamanho máximo da linha (em colunas). 75 colunas por defeito

```
fmt -35 LoremIpsum.txt | pr --columns=2
```

Processamento de texto

Tabulações e espaços

- › Comando *expand*
- › Permite converter tabulações em espaços

```
cat nomes3.txt -T
```

```
expand -t 16 nomes3.txt >> nomes4.txt
```

```
cat nomes4.txt -T
```

- › Comando *unexpand*
- › Permite converter espaços em tabulações

```
unexpand -t 16 nomes4.txt
```


Curso Linux

bibliografia

- › LPIC I, Exam Cram 2, Brunson - QUE Certification
- › LPI Linux Certification In a Nutshell, Pritchard, Pessanha, Langfeldt, Stranger & Dean – O REILLY
- › Linux Administration Handbook, Second edition, Nemeth Snyder Hein – Prentice Hall