Filtros y herramientas

Joaquín Seoane Pascual
joaquin@dit.upm.es

Departamento de Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid

21 de octubre de 2001

Índice General

Búsqueda de ficheros

- `find / -name core -print`
- `find / -user joaquin -print`
- `find / -atime +10 print`
- `find / -name core -exec rm {} \;`
- `find ~ -name '*.o' -exec rm {} \;`
- ```
  find ~ \( -name '*.o' -o -name core \) \
         -exec rm {} \;
  ```
  ```

Filtros unix

- Toman datos de la entrada estándar o de argumentos encadenados,
- Producen resultados en la *salida estándar*.
- Mensajes de error separados en la *salida de error*.
- Concebidos para componer.
- Opciones.

Filtros simples

- `cat` *[ficheros]*
- `od` *[ficheros]*
- `strings` *[ficheros]*
- `head` *[-cuenta] [ficheros]*
- `tail` *[-cuenta] [ficheros]*
- `compress` *[ficheros]*
- `gzip -9` *[ficheros]*
- `crypt` *[clave]*

```
ls -lrt | tail -3
od -x datos | tail -200 | pr | lpr
strings datos | grep version
tar -cvf - * | crypt XeKr | compress > /dev/fd0
```

Trasliteraciones

- `tr a-z A-Z`
- `tr A-Z a-z`
- `tr A-Za-z -`
- `tr -c A-Za-z -`
- `tr -sc A-Za-z '\012'`
- `tr -sc A-Za-z '\012' <poema | grep -i amor | wc -l`
- `dd conv=ascii`
- `dd conv=ebcdic`

Ordenaciones

- `sort /etc/passwd`
- `sort -r /etc/passwd`
- `du -a . | sort -n`
- `sort -t: +2n -3 /etc/passwd`
- `sort -t: +3n -4 +0r -1 /etc/passwd`
- ```
  cat *.c | tr -sc a-zA-Z '\012' |  \
  sort | \
  uniq -c | \
  sort -n | \
  tail
  ```
  ```

Búsquedas

`fgrep` *[opciones] cadena [ficheros]*
`grep` *[opciones] expresión regular [ficheros]*
`egrep` *[opciones] expresión regular [ficheros]*

Opción	*Descripción*
`-i`	Considera iguales mayúsculas y minúsculas
`-v`	Imprime líneas que **no** casan
`-c`	Imprime el número de líneas que casan
`-n`	Acompaña de número de línea en fichero
`-s`	Silencioso
`-f` *fichero*	Expresiones en fichero

Expresiones regulares en grep/egrep

Comunes	
`x`	El carácter `x` si no es especial
`\x`	El carácter `x`
`.`	Cualquier carácter
expr`*`	Repetición (0 ó más veces)
`.*`	Cualquier cadena
`[xyz]`	Enumeración de caracteres
`[a-z]`	Rango de caracteres
`[a-zA-Z0-9]`	Rangos de caracteres
`[abcA-Z]`	Enumeraciones y rangos
`[^A-Z]`	Excepto
`^`	Principio de línea
`$`	Fin de línea

Sólo de `egrep`	
expr`+`	Repetición (1 ó más veces)
expr`?`	0 ó 1 vez
`(`*expr*`)`	agrupamiento
expr1`|` *expr2*	alternativas

Ejemplos de grep y egrep

- `grep '^a.*n$' /usr/dict/words`
- `grep '^[^:]*::' /etc/passwd`
- `egrep 'love|amor' poema`
- `egrep '[Ll]ove|[Aa]mor' poema`
- `egrep '[aeiou][aeiou]+' poema`
- `egrep '([aeiou][aeiou])+' poema`

Edición de flujos

Invocaciones más frecuentes

`sed` *guión [ficheros]*
`sed -e` *guión* `-e` *guión ... [ficheros]*
`sed -f` *fichero_de_guión [ficheros]*
`sed -n` ...

Guiones

[dirección [,dirección]] función [argumentos]

Direcciones:

	todas las líneas
`55`	*línea 55*
`$`	*última línea*
`/`*expresión*`/`	*líneas que casan*
`\x`*expresión*`x`	*líneas que casan*

Rangos:

`1,9`	*líneas 1 a 9*
`6,$`	*de la 6 a la última*
`1,/printf/`	*de la 1 a la primera con* `printf`

Ejemplos de sed

- `sed 3q`
- `sed '/[Ee]jemplo/q'`
- `sed '/[Ee]jemplo/d'`
- `sed -n '/[Ee]jemplo/p'`
- `sed 's/patata/tomate/'`
- `sed 's/patata/tomate/g'`
- `sed '/solanacea/s/patata/tomate/g'`
- `who | sed 's/ .* / /'`
- `ls -t | sed '/^cosa$/q'`

Examen y proceso de patrones

awk *[-Fseparador] 'programa' [ficheros]*

patrón { acción }
patrón { acción }
...

Ejemplos simples

- `awk '/patata/ { print }'`
- `awk '/patata/'`
- `who | awk '{ print $1, $5}'`
- `who | awk '{ print $5, $1}' | sort`
- ```
  who | awk '{
  printf("%s entro a las %s\n", $1, $5)
  }'
  ```
- `awk -F: '{ print $1 }' /etc/passwd | sort`
- `awk -F: '$2 == ""' /etc/passwd`
- `awk '{ print NR, $0 }' texto`
  ```

awk como lenguaje de programación

- Variables no tipadas (números o cadenas)
- Arrays asociativos
- Expresiones
- Estructuras de control
- Funciones predefinidas
- Estructura de bloques
- Similar a C, pero *¡distinto!*
- Interpretado

Operadores awk

Operadores

`= += -= *= /= %=`	asignaciones
`||`	ó
`&&`	y
`!`	no
`> >= < <= == != ~ !~`	relacionales
nada	concatenación
`+ -`	aditivos
`* / %`	multiplicativos
`++ --`	incremento y decremento

Patrones awk

Patrones	
	todas las líneas
/expr/	las líneas que casan con *expr*
`$3 == "hola"`	el campo 3 es `hola`
`$3 != "hola"`	el campo 3 no es `hola`
`$3 ~ /h.*a/"`	el campo 3 casa con `h.*a`
`!($3 ~ /h.*a/)"`	el campo 3 no casa con `h.*a`
`$3 == $5`	los campos 3 y 5 son iguales
`length($4) < 5`	el campo tiene menos de 5 caracteres
`BEGIN`	línea anterior a la primera
`END`	línea posterior a la última

Variables awk

Predefinidas

`FILENAME`	fichero de entrada
`FS`	separador de campos de entrada
`RS`	separador de registros de entrada
`NF`	número de campos
`NR`	número de registro

Contador de palabras

```
awk 'BEGIN {total=0}
     {total+=NF}
     END {print total}'
```

Control y arrays awk

```
if ( expr )
    sentencia
else
    sentencia
```

```
for ( expr ; cond ; expr )
    sentencia
```

```
for ( var in array )
    sentencia
```

```
while ( expr )
    sentencia
```

Contador de frecuencias

```
awk '{ for (i=1; i<=NF; i++)
          cuenta[$i]++
}
END { for (x in cuenta)
         printf("%5d %s\n", cuenta[x], x)
}' texto | sort -n
```

Funciones predefinidas en awk

`sin(`*expr*`)`
`cos(`*expr*`)`
`log(`*expr*`)`
`exp(`*expr*`)`
`int(`*expr*`)`
`index(`*cad1*`, `*cad2*`)`
`length(`*cad*`)`
`split(`*cad*`,`*arr*`,`*car*`)`
`substr(`*cad*`,`*m*`,`*n*`)`
`sprintf(`*fmt*`,...)`