

# 1 Awk

O **awk** é unha linguaxe de programación empregada para traballar con texto. Así permite modificar caracteres de texto sendo comunmente empregado para presentar ou gardar en variables o texto que nos interesa.

## **awk -> Exemplo 1: Imprimir a primeira columna da saída da execución do comando ls -l**

*NOTA: Por defecto o carácter espazo determina a separación entre columnas*

```
alumno@gnu-linux:~/temporal$ ls -l
total 8
-rw-r--r-- 1 alumno alumno 24 2010-12-27 23:29 texto.txt
-rw-r--r-- 1 alumno alumno 36 2010-12-23 20:39 usuarios.txt
alumno@gnu-linux:~/temporal$ ls -l | awk '{print $1}'
total
-rw-r--r--
-rw-r--r--
```

## **awk -> Exemplo 2: Imprimir a primeira e oitava columna da saída da execución do comando ls -l**

```
alumno@gnu-linux:~/temporal$ ls -l | awk '{print $1 " " $8}'
total
-rw-r--r-- texto.txt
-rw-r--r-- usuarios.txt
```

## **awk -> Exemplo 3: Concatenar awk. Imprimir a primeira e oitava columna da saída da execución do comando ls -l, e logo dividir en columnas mediante o carácter punto**

```
alumno@gnu-linux:~/temporal$ ls -l | awk '{print $1 " " $8}' | awk -F. '{print $1 " " $2 " " $3}'
total
-rw-r--r-- texto txt
-rw-r--r-- usuarios txt
```

## **awk -> Exemplo 4: Imprimir por pantalla o número total de bytes de todos os ficheiros do directorio actual que foron modificados por última vez en Febreiro de 2011:**

```
alumno@gnu-linux:~/temporal$ ls -l | awk '$6 ~ /2011-02-/ { sum += $5 } END { print sum }'
36022
```

Tanto **sed** como **awk** son parte fundamental na execución dos scripts, pois permiten execucións dinámicas co mesmo código fonte, é dicir, o mesmo código pode xerar unha saída executable totalmente distinta. Por exemplo supoñamos o seguinte escenario:

1. Temos un *firewall*, digamos *iptables*, que impide acceder a un servizo tcp no porto 22 (ssh)
2. Creamos un script que permite abrir ese porto, por exemplo contendo a seguinte liña de código:

```
iptables -I INPUT -p tcp --dport 22 -j ACCEPT
```

Neste caso non fai falla empregar *sed* e *awk* posto que o porto sempre é o mesmo. Agora ben, supoñamos este outro escenario:

1. Temos un servizo que non ten claro que portos empregar, por exemplo **nfs**, podemos averiguar en cada momento que portos está empregando mediante o comando *rpcinfo -p*, e co mesmo podemos ver que se reiniciamos o servizo nfs na próxima execución os portos empregados non son os mesmos, entón agora como podemos facer que o noso script de forma estática abra eses portos?
2. Xa que cambian poderíamos coller eses portos no script como variables e facer que a sintaxe anterior difira mais ou menos na seguinte:

```
iptables -I INPUT -p tcp --dport $PORTO_DINAMICO -j ACCEPT
```

Agora ben, como facemos para coller eses portos, pois si, con *sed* e *awk*. Mediante **sed** podemos variar as expresións á nosa vontade, por exemplo: que desaparezan os espazos, que un patrón de texto cambie... e con **awk** podemos separar por caracteres calquera saída en columnas *-considerando a separación de columnas, por defecto, mediante o carácter espazo ou ben por outro carácter que indiquemos-*, co cal, se nunha saída de execución de comandos temos digamos 5 columnas, podemos recoller a columna desexada na variable pertinente.

A potencia do *sed* e *awk* é abismal xa que permite dinamizar os scripts.