

PRÁCTICA 5: *Awk en bash*

ALUMNO: Jose Angel Albaladejo Sanchez

1. INTRODUCCIÓN

Awk es un verdadero lenguaje de programaci3n, especializado en análisis y procedimiento de patrones de texto. Es una mezcla de *grep* y *sed*, vistos en las anteriores prácticas. La sintaxis básica es la siguiente:

```
awk [-f archivo_programa] [-Fc] ['programa'] [variable=valor ...] [archivo]
```

Entrada	Descripci3n
Archivo_programa	Especifica el archivo fuente del programa a aplicar al archivo
c	Especifica el carácter delimitador de campos. Por defecto se considera el espacio en blanco
Programa	Especifica el conjunto de patrones e instrucciones a ejecutar en el archivo
Variable=valor	Se utiliza para establecer los valores que tomarán las variables que utilice el programa
Archivo	Es el archivo que será procesado por <i>awk</i> . Si se especifica “-” se interpretará como la entrada estándar

Un ejemplo de la aplicaci3n del comando *awk* es el siguiente: podemos utilizar *awk* para imprimir archivos de texto como *cat*. Simplemente escribiendo

```
awk '{print}' archivo.txt
```

Tambi3n funciona si escribimos

```
awk '{print $0}' archivo.txt
```

Si adem3s queremos imprimir una determinada palabra, lo haremos con el comando

```
awk '/palabra/{print $0}' archivo.txt
```

Igual que si queremos imprimir las líneas que empiezan por dicha palabra:

```
awk '/^palabra/{print $0}' archivo.txt
```

Una característica muy interesante de *print* es que si cambiamos el 0 por el 1 imprimirá la primera palabra de cada línea.

2. OBJETIVO DE LA PRÁCTICA

El objetivo de esta práctica es que os familiaricéis con las acciones básicas del comando *awk* en la consola *bash* de linux.

3. EJERCICIOS GUIADOS EN BASH

3.1. Imprime las dos noticias adjuntas a la práctica con el comando *awk*.

```
joseangel@joseangel-virtualbox:~/Desktop$ awk '{print}' nba.txt teleco.txt
Locura en el Madison: pérdidas, tiro imposible... y polémica
En una jugada final muy contravertida, los Pistons sacaron el balón pa
```

3.2. Imprime la primera palabra de cada línea como pone en la introducción de esta práctica.

```
joseangel@joseangel-virtualbox:~/Desktop$ awk '{print $1}' nba.txt teleco.txt
Locura
En
Entonces,
```

3.3. Imprime todas las líneas que contengan la palabra *baloncesto* del archivo *nba.txt* con el comando *awk*.

```
joseangel@joseangel-virtualbox:~/Desktop$ awk '/baloncesto/' nba.txt
esto está alargando en demasía el tiempo de los partidos, ya de por sí con un
a duración excesiva, y convirtiendo el baloncesto
```

3.4. Imprime todas las líneas que contengan la palabra *de* en el archivo *teleco.txt*.

```
joseangel@joseangel-virtualbox:~/Desktop$ awk '/de/' teleco.txt
Cambios en el mercado de las telecom
La fusión entre Orange y MásMóvil, que se produce en un momento de concentrac
ión de empresas europeas y con el reto de la IA en el
```

3.5. Cuenta el número total de líneas de ambas noticias utilizando *awk 'END {print NR}' archivo.txt* donde, con *NR*, especificamos que queremos dicho número.

```
en los derechos de los consumidores.  
joseangel@joseangel-virtualbox:~/Desktop$ awk 'END {print NR}' nba.txt teleco  
.txt  
84  
joseangel@joseangel-virtualbox:~/Desktop$
```

3.6. Cuenta ahora el número de palabras de cada archivo. Para realizar esto debes utilizar

`awk '{total += NF} END {print total}' archivo.txt`

```
joseangel@joseangel-virtualbox:~/Desktop$ awk '{total += NF} END {print total  
}' nba.txt  
922  
joseangel@joseangel-virtualbox:~/Desktop$ awk '{total += NF} END {print total  
}' teleco.txt  
524  
joseangel@joseangel-virtualbox:~/Desktop$
```

3.7. Muestra las líneas que contengan más de 30 caracteres de cada fichero. Para ello, tras `awk` especifica entre comillas `'length($0) > 30'`.

```
joseangel@joseangel-virtualbox:~/Desktop$ awk 'length($0) > 30' teleco.txt  
Cambios en el mercado de las telecom  
La fusión entre Orange y MásMóvil, que se produce en un momento de concentrac
```

```
joseangel@joseangel-virtualbox:~/Desktop$ awk 'length($0) > 30' nba.txt  
Locura en el Madison: pérdidas, tiro imposible... y polémica  
En una jugada final muy controvertida, los Pistons recuperaron el balón para  
volver a perderlo.
```

3.8. También podemos contar el número de veces que aparece una determinada palabra en un fichero con `awk`. Para ello, especifica `'BEGIN {count=0} /palabra/ {count++} END {print count}'`.

```
awk: cannot open nba.txt (No such file or directory)  
joseangel@joseangel-virtualbox:~/Desktop$ awk 'BEGIN {count=0} /palabra/ {cou  
nt++} END {print count}' nba.txt  
0  
joseangel@joseangel-virtualbox:~/Desktop$
```

3.9. Cuenta el número de veces que aparece la palabra *baloncesto* en *nba.txt*.

```
joseangel@joseangel-virtualbox:~/Desktop$ awk 'BEGIN {count=0} /baloncesto/ {  
count++} END {print count}' nba.txt  
1  
joseangel@joseangel-virtualbox:~/Desktop$
```

3.10. Cuenta el número de veces que aparece la palabra *España* en *teleco.txt*.

```
joseangel@joseangel-virtualbox:~/Desktop$ awk 'BEGIN {count=0} /España/ {count++} END {print count}' teleco.txt
3
```

3.11. También podemos imprimir las líneas que están en posición impar con

`awk 'NR % 2 != 0' archivo.txt`

Lo que hace esta sintaxis es especificar que si al dividir el número de línea (NR) entre dos el resto da distinto (! =) de cero, que imprima la línea.

Imprime las líneas impares de *nba.txt*.

```
joseangel@joseangel-virtualbox:~/Desktop$ awk 'NR % 2 != 0' nba.txt
Locura en el Madison: pérdidas, tiro imposible... y polémica
En una jugada final muy controvertida, los Pistons recuperaron el balón para volver a perderlo.
```

3.12. Piensa cómo imprimir las líneas pares de *teleco.txt* (para poner igual se deben utilizar dos signos ==).

```
joseangel@joseangel-virtualbox:~/Desktop$ awk 'NR % 2 == 0' teleco.txt
horizonte, debe garantizar los derechos de los usuarios.
Casi dos años después de que Orange y MásMóvil anunciaran su decisión de fusionarse, la Comisión Europea ha dado el visto bueno a la
```

3.13. Imprime las líneas cuyo número de línea sea múltiplo de tres en *nba.txt*.

```
joseangel@joseangel-virtualbox:~/Desktop$ awk 'NR % 3 == 0' nba.txt
En una jugada final muy controvertida, los Pistons recuperaron el balón para volver a perderlo.
Lo de los árbitros está empezando a ser una constante demasiado grande en la NBA últimamente. Siempre se habla de ellos, pidiéndolo permanentemente, los entrenadores con dos challenges cada uno y permanecer la posibilidad de pedirlo intacta si
```

Además, el comando *awk* también nos permite reemplazar palabras por otras, tal y como hemos visto con otros comandos en *bash*. Esto es posible con *gsub*:

`awk '{gsub("palabra1", "palabra2"); print}' archivo.txt`

3.14. Reemplaza la palabra *baloncesto* por *basket* en el archivo *nba.txt*.

```
joseangel@joseangel-virtualbox:~/Desktop$ awk '{gsub(/baloncesto/, "basket"); print}' nba.txt > nba_modificado.txt
joseangel@joseangel-virtualbox:~/Desktop$
```

3.15. Reemplaza la palabra *España* por *Nación* en *teleco.txt*.

```
joseangel@joseangel-virtualbox:~/Desktop$ awk '{gsub(/España/, "Nacion"); print}' teleco.txt > teleco_modificado.txt
joseangel@joseangel-virtualbox:~/Desktop$
```

3.16. Comprueba que lo que hace el anterior comando es sustituir a la hora de imprimir en la consola, pero que si volvemos a imprimir el archivo ya no aparece el reemplazo.

```
joseangel@joseangel-virtualbox:~/Desktop$ sed -i 's/baloncesto/basket/g' nba.txt
joseangel@joseangel-virtualbox:~/Desktop$
```

3.17. Imprime las líneas que empiezan con la palabra *protagonistas* en el archivo *nba.txt*.

```
joseangel@joseangel-virtualbox:~/Desktop$ awk '/^protagonistas/' nba.txt
protagonistas no fueron los jugadores. Por lo que sea.
joseangel@joseangel-virtualbox:~/Desktop$
```

3.18. Imprime las líneas que empiezan por la palabra *protagonistas* o por la palabra *de* en el archivo *nba.txt*. Para ello, tras escribir la referencia a *awk*, entre comillas escribe `'/^palabra1|^palabra2/` (importante dejar los espacios que se muestran).

```
joseangel@joseangel-virtualbox:~/Desktop$ awk '/^protagonistas|^de/' nba.txt
de Ausar Thompson, que se disponía a superar el centro del campo cuando era a
rrollado por el propio DiVincenzo. Monty Williams aireaba
de Hart apenas quedaban 2,8. Es decir, también se perdió un tiempo precioso q
ue los Pistons podrían haber utilizado para atacar. Un
de OG Anunoby y Julius Randle para los neoyorquinos y que se resolvió de la p
eor forma posible. Entre otras cosas, porque los
protagonistas no fueron los jugadores. Por lo que sea.
joseangel@joseangel-virtualbox:~/Desktop$
```

3.19. Piensa en cómo imprimir las líneas que tengan la palabra *zona* o la palabra *cosas* en *nba.txt* (es decir, no hace falta que empiecen por esa palabra, sino que la contengan).

```
joseangel@joseangel-virtualbox:~/Desktop$ awk '/zona|cosas/' nba.txt
y 8 asistencias. Jaden Ivey y Quentin Grimes sumaron 14 por barba y Jalen Dur
en se hizo fuerte en la zona y llegó hasta 11, con 18
de OG Anunoby y Julius Randle para los neoyorquinos y que se resolvió de la p
eor forma posible. Entre otras cosas, porque los
joseangel@joseangel-virtualbox:~/Desktop$
```

3.20. Busca información sobre otras opciones que proporciona el comando *awk*, aparte de las vistas en esta práctica.

Campos y Delimitadores:

Puedes especificar campos y delimitadores utilizando las opciones *-F* para establecer el delimitador. Por ejemplo, *awk -F',' '{print \$1}' file.csv* imprimirá el primer campo de un archivo CSV.

Formato de Salida:

Puedes personalizar el formato de salida utilizando la función *printf*. Por ejemplo, *awk '{printf "%s\t%s\n", \$1, \$2}' file.txt* imprimirá los dos primeros campos separados por tabulaciones.

Condiciones Múltiples:

Puedes combinar condiciones usando operadores lógicos (*&&*, *||*). Por ejemplo, *awk '/patron1/ && /patron2/' file.txt* imprimirá las líneas que contienen tanto "patron1" como "patron2".