



1º DAM/DAW Sistemas Informáticos

U2. Comandos en Linux.

Comandos. Contenido de ficheros II

Comando AWK

- **AWK** es un comando muy versátil que sirve para procesar patrones en líneas de texto. Es decir, podemos usarlo por ejemplo:

- Para acceder a datos con formato de tabla, que estén en un fichero, o lleguen a través de una tubería.



- Para acceder a datos separados por un delimitador que estén en un fichero, o lleguen a través de una tubería.

root	0	0
daemon	1	1
bin		2 2
sys		3 3
sync	4	65534
games	5	60

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
```

Comando AWK

- **AWK** Ejemplo: vamos a ejecutar `ls -l` en nuestra carpeta de usuario. Y el resultado lo vamos a redirigir, mediante una tubería, para que el comando `awk` muestre el valor de las columnas **3** y **9** por pantalla.
- `$ ls -l | awk '{print $3, $9}'`

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ ls -l | awk '{print $3, $9}'  
  
pmartinez a.txt  
pmartinez codigo  
pmartinez datos  
pmartinez Descargas  
pmartinez Documentos  
pmartinez Escritorio  
pmartinez examples.desktop  
pmartinez Imágenes  
pmartinez Música  
pmartinez noexiste.txt  
pmartinez nombres.txt  
pmartinez no.txt  
pmartinez Plantillas  
pmartinez pruebatexto.txt  
pmartinez Público  
pmartinez salida.txt  
pmartinez texto.txt  
pmartinez U2  
pmartinez Videos  
pmartinez@Ubuntu12:~$
```

```
pmartinez@Ubuntu12: ~  
pmartinez@Ubuntu12:~$ ls -l  
total 84  
-rw-rw-r-- 1 pmartinez pmartinez 9 nov 8 18:55 a.txt  
drwxrwxr-x 4 pmartinez pmartinez 4096 oct 18 18:24 codigo  
drwxrwxr-x 2 pmartinez pmartinez 4096 oct 18 18:27 datos  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Descargas  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Documentos  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Escritorio  
-rw-r--r-- 1 pmartinez pmartinez 8445 oct 1 11:13 examples.desktop  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Imágenes  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Música  
-rw-rw-r-- 1 pmartinez pmartinez 6 nov 9 14:11 noexiste.txt  
-rw-rw-r-- 1 pmartinez pmartinez 71 nov 8 19:13 nombres.txt  
-rw-rw-r-- 1 pmartinez pmartinez 4 nov 9 14:02 no.txt  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Plantillas  
-rw-rw-r-- 1 pmartinez pmartinez 6 nov 9 14:10 pruebatexto.txt  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Público  
-rw-rw-r-- 1 pmartinez pmartinez 1190 nov 9 14:08 salida.txt  
-rw-rw-r-- 1 pmartinez pmartinez 7 nov 9 14:12 texto.txt  
drwxrwxr-x 2 pmartinez pmartinez 4096 nov 9 13:00 U2  
drwxr-xr-x 2 pmartinez pmartinez 4096 oct 1 13:20 Videos
```

Ampliación comando GREP

- Ampliando un poco la utilidad del comando grep, podríamos decir que nos **permite localizar coincidencias en el contenido de ficheros, dado un patrón de búsqueda.**
- El patrón de búsqueda se denomina también **expresión regular**. Una expresión regular es una denominación más genérica utilizada también en otros ámbitos, y en detalle, es una secuencia de caracteres que conforma un patrón de búsqueda.



GREP. Sintaxis

- La sintaxis del comando grep es la siguiente:

\$ grep [opciones] [expresión regular] [archivos]

- Donde:
 - **[opciones]**: nos permite indicar parámetros para personalizar el uso del comando.
 - **[expresión regular]**: nos permite definir el patrón de búsqueda.
 - **[archivos]**: nos permite expresar los archivos (ruta) donde se realizará la búsqueda.



GREP. Buscar en el contenido de un fichero

- Vamos a comenzar con el caso más básico del comando grep, en el que queremos buscar una cadena de texto en un fichero conocido:

\$ grep “texto” fichero.txt

- Otra opción muy habitual para realizar lo mismo sería:

\$ cat fichero.txt | grep “texto”

- El comando devolverá las líneas del fichero que contengan una coincidencia con el patrón de búsqueda, es decir, la cadena “**texto**”.

GREP. Buscar en el contenido de un fichero, **opciones**

- Respecto a las opciones que podemos indicar como parámetros (podemos consultarlas en el Terminal con el comando **man**), unas de las más relevantes son las siguientes:

-i : no tendrá en cuenta mayúsculas y minúsculas a la hora de buscar.

\$ cat fichero.txt | grep -i "texto"

-v : nos devuelve el resultado inverso, es decir, las líneas que no cumplan el patrón.

\$ cat fichero.txt | grep -v "texto"

-n : muestra el número de línea donde sucede la coincidencia.

\$ cat fichero.txt | grep -n "texto"

-w : encuentra coincidencia sólo si el patrón coincide con una palabra completa.

\$ cat fichero.txt | grep -w "texto"

GREP. Buscar en el contenido de un fichero, **opciones**

-A num : nos muestra las líneas posteriores a una coincidencia. El número de líneas a mostrar dependerá del valor **num**.

```
$ cat fichero.txt | grep -A 1 "texto"
```

Nos mostrará cada línea del fichero donde aparezca la cadena "texto" y la posterior.

-B num : nos muestra las líneas anteriores a una coincidencia. El número de líneas a mostrar dependerá del valor **num**.

```
$ cat fichero.txt | grep -B 2 "texto"
```

Nos mostrará cada línea del fichero donde aparezca la cadena "texto" y las 2 anteriores.

-b : nos muestra la posición de la coincidencia en el fichero

```
$ cat fichero.txt | grep -b "texto"
```

- Las opciones se pueden combinar, por ejemplo:

```
$ cat fichero.txt | grep -iw "texto"
```


GREP. Buscar en el contenido de un fichero, **expresión regular**

- Hasta este momento, hemos visto cómo localizar cadenas de texto. Vamos a centrarnos ahora en cómo generar una **expresión regular** que nos sirva como **patrón** para realizar la búsqueda.
- Las expresiones regulares, nos **permiten definir una serie de aspectos con mayor nivel de detalle**, para concretar mejor la búsqueda a realizar.
- Por ejemplo, si queremos buscar una cadena de texto, pero queremos que sea la primera palabra de una línea, o la última. O bien sabemos que la cadena está formada por 3 letras y 4 números, pero no sabemos qué cadena es.....

GREP. Buscar en el contenido de un fichero, **expresión regular**

- Las expresiones regulares utilizan una serie de caracteres para poder indicar este tipo de cuestiones:

- ^ (acento circunflejo): este carácter utilizado junto a una cadena de texto, indica que dicha cadena debe aparecer, al comienzo de una línea del fichero.

`$ cat fichero.txt | grep "^Texto"`

- \$: lo mismo que el anterior pero al final de una línea del fichero.

`$ cat fichero.txt | grep "texto$"`

- . (punto): sustituye a cualquier carácter (similar al carácter comodín ? usado en el sistema de archivos).

`$ cat fichero.txt | grep "t...o"`

GREP. Buscar en el contenido de un fichero, **expresión regular**

- **[]** (corchetes): lo que va entre corchetes se entiende como un conjunto de posibilidades que debe cumplir el resultado.

- **\$ cat fichero.txt | grep "[a-z]texto" :**

En este ejemplo, buscaría una cadena formada por una letra desde la **a** hasta la **z**, seguida de la palabra "texto". Por ejemplo, encontraría una línea del fichero, si contiene la palabra "**contexto**" o "**pretexto**".

- **\$ cat fichero.txt | grep "U[0-9]" :**

En este ejemplo, buscaría una cadena formada por la letra "U" seguida de un número de **0** a **9**. Por ejemplo, encontraría la línea del fichero si contiene la palabra "**U1**" o "**U2**".

GREP. Buscar en el contenido de un fichero, **expresión regular**

- **[]** (corchetes): podemos indicar el número de **repeticiones** de una letra o un número. Y también podemos **acotar el conjunto** de caracteres posible para una letra o un número:

- **\$ cat fichero.txt | grep "[a-t]*texto" :**

En este ejemplo, el **"*"** indica que la posible letra se repite de **0** a **n veces**, en este caso acotando el conjunto desde la **a** hasta la **t** (en orden alfabético), y después va seguida la palabra **"texto"**. Por ejemplo, encontraría una línea del fichero, si contiene la palabra **"contexto"** o **"pretexto"** o **"texto"** o **"htmltexto"**...

- **\$ cat fichero.txt | grep "U[0-1]\{1,2\}" :**

En este ejemplo, buscaría una cadena formada por la letra **"U"**, en este caso seguida de **1** o **2** números acotados entre **0** y **1**. Por ejemplo, encontraría la línea del fichero si contiene la palabra **"U0"** o **"U1"** o **"U10"** o **"U01"**.

GREP. Buscar en múltiples ficheros

- Cuando la necesidad se amplía y en lugar de buscar coincidencias en 1 único fichero, lo que necesito es localizar los ficheros de una determinada ruta cuyo contenido contiene un patrón de búsqueda, lo podemos hacer del siguiente modo:

\$ grep [opciones] “patrón” ruta_archivos

Donde **ruta_archivos** puede ser, por ejemplo:

- Todos los archivos de la carpeta actual (ruta relativa): **./***
- Los archivos que empiezan por lib, de la carpeta padre (ruta relativa): **../lib***
- Los archivos de texto de la carpeta de usuario (ruta absoluta): **/home/usuario/*.txt**
- Cualquier otra ruta absoluta o relativa...

GREP. Buscar en múltiples ficheros, **opciones**

Donde las [opciones] podrían ser:

-r : búsqueda recursiva, para que busque también en los subdirectorios.

\$ grep -r "texto" ./*

Buscará la cadena "texto" en todos los ficheros de la carpeta actual y en todos los subdirectorios de la carpeta actual también.

Nos dará como resultado la ruta de los ficheros en los que encuentre la cadena "texto" y, dentro de cada fichero, las líneas en las que ha localizado la coincidencia.