El shell de linux: Comando grep

El comando **grep** nos permite buscar, dentro de los archivos, las líneas que concuerdan con un patrón. Bueno, si no especificamos ningún nombre de archivo, tomará la entrada estándar, con lo que podemos encadenarlo con otros filtros.

Por defecto, grep imprime las líneas encontradas en la salida estándar. Es decir, que podemos verlo directamente la pantalla, o redireccionar la salida estándar a un archivo.

Como tiene muchísimas opciones, vamos a ver tan sólo las más usadas:

-c En lugar de imprimir las líneas que coinciden, muestra el número de líneas que coinciden.
-e PATRON nos permite especificar varios patrones de búsqueda o proteger aquellos patrones de búsqueda que comienzan con el signo -.
-r busca recursivamente dentro de todos los subdirectorios del directorio actual.
-v nos muestra las líneas que no coinciden con el patrón buscado.
-i ignora la distinción entre mayúsculas y minúsculas.
-n Numera las líneas en la salida.
-E nos permite usar expresiones regulares. Equivalente a usar **egrep**.
-o le indica a grep que nos muestre sólo la parte de la línea que coincide con el patrón.
-f ARCHIVO extrae los patrones del archivo que especifiquemos. Los patrones del archivo deben ir uno por línea.
-H nos imprime el nombre del archivo con cada coincidencia.

Veamos algunos ejemplos:

- Buscar todas las líneas que contengan palabras que comiencen por **a** en un archivo:
$ grep '\<a.*\>' archivo

Otra forma de buscar, sería:
$ cat archivo | grep "\<a.*\>"

- Mostrar por pantalla, las líneas que contienen comentarios en el archivo /boot/grub/menu.lst:
$ grep "#" /boot/grub/menu.lst

- Enviar a un fichero las líneas del archivo /boot/grub/menu.lst que no son comentarios:
$ grep -v "#" /boot/grub/menu.lst

- Contar el número de interfaces de red que tenemos definidos en el fichero /etc/network/interfaces:
$ grep -c "iface" /etc/network/interfaces

- Mostrar las líneas de un fichero que contienen la palabra BADAJOZ o HUELVA:
$ grep -e "BADAJOZ" -e "HUELVA" archivo

- Mostrar las líneas de un fichero que contienen la palabra BADAJOZ o HUELVA, numerando las líneas de salida:
$ grep -n -e "BADAJOZ" -e "HUELVA" archivo

- Mostrar los ficheros que contienen la palabra TOLEDO en el directorio actual y todos sus subdirectorios:
$ grep -r "TOLEDO" *

Veamos algunos ejemplos con expresiones regulares:

- Obtener la dirección MAC de la interfaz de red eth0 de nuestra máquina:
$ ifconfig eth0 | grep -oiE '([0-9A-F]{2}:){5}[0-9A-F]{2}'

Sacamos la dirección MAC de la interfaz eth0 de nuestra máquina haciendo un:
ifconfig eth0

Y aplicando el filtro grep:
grep -oiE '([0-9A-F]{2}:){5}[0-9A-F]{2}'

Las opciones que he usado en grep son:

-o Indica que la salida del comando debe contener sólo el texto que coincide con el patrón, en lugar de toda la línea, como es lo habitual.
-i Lo he usado para que ignore la distinción entre mayúsculas y minúsculas.
-E Indica que vamos a usar una expresión regular extendida.

En cuanto a la expresión regular, podemos dividirla en dos partes:
([0-9A-F]{2}:){5} Buscamos 5 conjuntos de 2 carateres seguidos de dos puntos
[0-9A-F]{2} seguido por un conjunto de dos caracteres.

Como las direcciones MAC se representan en hexadecimal, los caracteres que buscamos son los números del 0 al 9 y las letras desde la A a la F.

- Extraer la lista de direcciones de correo electrónico de un archivo:
grep -Eio '[a-z0-9._-]+@[a-z0-9.-]+[a-z]{2,4}' fichero.txt

Utilizo las mismas opciones que en el caso anterior:

-o Indica que la salida del comando debe contener sólo el texto que coincide con el patrón, en lugar de toda la línea, como es lo habitual.
-i Lo he usado para que ignore la distinción entre mayúsculas y minúsculas.
-E Indica que vamos a usar una expresión regular extendida.

Analicemos ahora la expresión regular:
[a-z0-9._-]+@[a-z0-9.-]+[a-z]{2,4}

Al igual que antes, la vamos dividiendo en partes:
[a-z0-9._-]+ Una combinación de letras, números, y/o los símbolos . _ y - de uno o más caracteres
@ seguido de una arroba
[a-z0-9.-]+ seguido de una cadena de letras, números y/o los símbolos . y -
[a-z]{2,4} seguido de una cadena de entre dos y cuatro caracteres.

- Obtener la dirección IP de la interfaz de red eth1 de nuestra máquina:
$ ifconfig eth1 | grep -oiE '([0-9]{1,3}\.){3}[0-9]{1,3}' | grep -v 255

En el ejemplo anterior, hemos tomado la información que nos ofrece ifconfig:
ifconfig eth1

Hemos filtrado dicha información con el comando grep, obteniendo todas las direcciones IP que aparecen:
grep -oiE '([0-9]{1,3}\.){3}[0-9]{1,3}'

Por último, hemos filtrado la salida del comando anterior, para eliminar la dirección de broadcast junto con la máscara de red para quedarnos sólo con la dirección IP de la máquina:
grep -v 255

La línea anterior no mostraría las líneas que no contengan el valor 255, es decir, las direcciones de broadcast y máscara de red.

Analicemos ahora el comando grep:
grep -oiE '([0-9]{1,3}\.){3}[0-9]{1,3}'

Al igual que en los otros dos ejemplos de expresiones regulares uso las opciones -oiE en el comando grep:

-o Indica que la salida del comando debe contener sólo el texto que coincide con el patrón, en lugar de toda la línea, como es lo habitual.
-i Lo he usado para que ignore la distinción entre mayúsculas y minúsculas.
-E Indica que vamos a usar una expresión regular extendida.

En cuanto a la expresión regular:
'([0-9]{1,3}\.){3}[0-9]{1,3}'
([0-9]{1,3}\.){3} Representa 3 bloques de entre uno y tres dígitos separados por puntos. Observemos que como el punto es un metacaracter, tengo que usar el caracter de escape \ para que no sea interpretado como un metacaracter, sino como un caracter normal.
[0-9]{1,3} Representa el último bloque de la dirección IP, que está formado por un número de entre 1 y 3 dígitos.