

Tema 2 - Actividad 3: Comandos básicos

Recuerda:

Sintaxis grep

- `grep -r` busca recursiva
- `grep -n` muestra numeros de línea
- `grep -i` case insensitive (mayusculas o minusculas)
- `grep -v` excluir una expresión de búsqueda
- `grep -e` búsqueda de varias expresiones
- `grep "^cadena"` buscar expresiones que empiecen por cadena
- `grep "cadena$"` buscar expresiones que acaben en cadena

Sintaxis find.

Por defecto el directorio de búsqueda es el directorio actual "." La búsqueda es recursiva (incluye subdirectorios). Se pueden usar búsqueda con comodines y expresiones regulares

- `find [dir] -name "nombre_de fichero"`
- `find [dir] -iname "nombre_de fichero"` [sin importar mayusculas o minusculas]
- `find [dir] -not -name "nombre_de fichero"` [buscar todos los ficheros, excepto "nombre de fichero"]
- `find [dir] -name "nombre_de fichero" -delete` [buscar y borrar Cuidado !]
- `find [dir] -type [d(irectory), f(ile), etc]` por tipo de fichero
- `find [dir] -[mtime,atime]` por tiempo
- `find [dir] -size [c:byte, k:kilobytes, M:megas, G:giga, b:bloques de 512 bytes]` por tamaño
- `find [dir] -perm [perm]` por permisos
- `find [dir] -executable -readable`

Ejercicio 1. (grep)

`ip neigh` nos da la tabla arp de la interfaz Ubuntu. Seleccionar las interfaces que están en estado STALE, y mostrar solo las dos últimas líneas

(NOTA: Si al ejecutar el comando `ip neigh` solo sale una línea, entonces utilizar el comando `ip a`, seleccionar en este caso las líneas que contengan la cadena "forever" y mostrar sólo las dos últimas).

```
ip neigh | grep "STALE" | tail -n 2
```

Ejercicio 2. (find)

Buscar los ficheros de tu directorio y tu subdirectorio de trabajo, que tienen un tamaño de 4096 bytes. NOTA: Los directorios tienen 4096 bytes

```
find . -type f -size 4096c
```

Ejercicio 3. (find)

- a) Buscar en el directorio de trabajo todos los ficheros ordinarios (no directorios ni enlaces) que tengan exclusivamente permiso de ejecución para el usuario.
- b) Busca después los ficheros cuyo tipo son exclusivamente enlaces simbólicos.

```
a) find . -type f -perm 700
b) find . -type l
```

Ejercicio 4. (awk)

Usando el comando `ip a` y usando la herramienta `awk`, imprime tu dirección MAC, de tu interfaz de red activo

```
ip a | awk '/ether/ {print $2}'
```

Ejercicio 5 (sed)

```
echo "1) John      Maths      6.54" > notes.txt
echo "2) Paul      Physics     8.23" >> notes.txt
echo "3) Michael   Biology     7.98" > notes.txt
echo "4) Steve     Physics     5.10" >> notes.txt
echo "5) Jack      History     4.68" >> notes.txt
echo "6) Richard   Programming 9.05" >> notes.txt
echo "7) John      Programming 8.42 - this note should be reviewed" >> notes.txt
```

- a) Crear un fichero nuevasnotas.txt. Cambiar 9.05 por "sobresaliente"

```
sed 's/9.05/sobresaliente/' < notes.txt > nuevasnotas.txt
```

Ejercicio 6. (awk)

Ejecutar un ls -l y obtener los datos que se indican:

- a) Mostrar solo los nombres y los tamaños de los ficheros
- b) Mostrar solo los nombres y los tamaños de los ficheros con extensión *.txt

```
a) ls -l | awk '{print $5 "\t" $9}'  
b) ls -l | awk '/.txt/ {print $5 "\t" $9}'
```

Ejercicio 7. (awk)

Rellena el fichero agenda.txt con la siguiente información:

```
echo "Lorenzo Gonzalez lorenzo@gmail.com 678333222 Madrid" > agenda.txt  
echo "Federico Martin fm@tef.net 698444555 Madrid" >> agenda.txt  
echo "Clara García clara@tef.com 972666555 Barcelona" >> agenda.txt  
echo "Luis Dominguez ld@gmail.com 625999888 Burgos" >> agenda.txt  
echo "Carlos Velez carlosv@gmail.com 625000111 Madrid" >> agenda.txt  
echo "Luisa Fernández luisa@gmail.com 975222777 Barcelona" >> agenda.txt
```

- a) Mostrar los correos electrónicos de todos los contactos del fichero agenda.txt
- b) Mostrar los contactos que residan en Barcelona
- c) Mostrar cuantas personas viven en Madrid
- d) Mostrar las líneas que tengan más de 50 caracteres
- e) Mostrar el nombre y el teléfono de cada uno de los contactos con el siguiente formato: Nombre: xxxx, Teléfono:xxxxx

```
a) awk '{print $3}' agenda.txt  
b) awk '/Barcelona/ {print}' agenda.txt  
c) awk 'BEGIN {} /Madrid/ {++counter} END {print "Viven en Madrid: ", counter}' agenda.txt  
d) awk 'length ($0) > 50' agenda.txt  
e) awk '{print "Nombre: " $1 $2 "\t" "Telefono: " $4}' agenda.txt
```

Ejercicio 8. (awk y grep)

Usando el comando ip addr, obtener la dirección IP del interfaz enp0s3, usando al menos los comandos grep y awk. Hay que excluir de la dirección IP cualquier otro texto asociado. El script obtenido tiene que ser universal, es decir tiene que servir para cualquier resultado del comando ip addr en cualquier ordenador (no debe contener el nombre de la interfaz, se sabe que la IP es dinámica).

```
ip addr | grep "enp0s3" | awk '/net/ {print $2}'
```

Ejercicio 9. (cut, tr y sort)

```
echo "2010:Junio:23:Carla:Martínez:Equipo-de-basquet" > extraescolares.txt  
echo "2011:Abril:22:Yoel:Alonso:Clases-de-danza" >> extraescolares.txt  
echo "2009:Febrero:21:Bernat:Molina:Equipo-de-basquet" >> extraescolares.txt  
echo "2010:Mayo:11:Arnau:Silvestre:Clases-de-judo" >> extraescolares.txt  
echo "2011:Febrero:23:Lucia:Pérez:Clases-de-danza" >> extraescolares.txt  
echo "2013:Mayo:20:Gerard:Reverte:Clases-de-judo" >> extraescolares.txt  
echo "2010:Marzo:02:Isaac:Meler:Club-de-ajedrez" >> extraescolares.txt  
echo "2009:Abril:25:Carmen:Grandes:Clases-de-natación" >> extraescolares.txt  
echo "2011:Noviembre:22:Isabel:Valle:Equipo-de-fútbol-sala" >> extraescolares.txt  
echo "2010:Mayo:14:María:Segovia:Equipo-de-basquet" >> extraescolares.txt
```

- a) Extraer únicamente los nombres y apellidos de los alumnos
- b) Sustituir los ":" de separador por un espacio en blanco o un tabulador.
- c) Ordenar el resultado alfabéticamente
- d) Mostrar información de los alumnos del Club de Basquet (nombre, apellidos y nombre de la actividad) ordenados alfabéticamente

```
a) cut -d ":" -f 4,5 extraescolares.txt
b) cat extraescolares.txt | tr ':' '\t'
c) cat extraescolares.txt | tr ':' '\t' | sort -k4
d) cat extraescolares.txt | tr ':' '\t' | grep "basquet" | sort -k4
```

Ejercicio 10. (awk)

Utilizando el siguiente fichero ejemplo imprime el tercer campo de la secuencia y envia el resultado a un nuevo fichero

```
echo "1;2;3;4;5;6" > ejemplo1.txt

cat ejemplo1.txt | tr ';' '\t' | awk '{print $3}' > ejemplo2.txt
```