

COMANDOS. ESTRUCTURA Y CONTENIDO DE DIRECTORIOS Y FICHEROS

Desde **directorio personal**, lista el nombre de los contenidos del directorio **/dev** que empiecen por **tty**.

```
cd $HOME
```

```
ls/dev/tty*
```

Desde **/dev**.

```
cd /dev/
```

```
ls tty*
```

Desde tu **directorio personal**, lista el nombre de los contenidos del directorio **/dev** que empiecen por **tty** y tengan **5 caracteres** en total.

```
cd $HOME
```

```
ls /dev/tty??
```

Haz lo mismo que antes, desde el directorio **/dev**.

```
cd /dev/
```

```
ls tty??
```

Realiza los puntos 2 y 4, pero uniendo en **una misma línea** (un mismo mandato), los 2 comandos que has utilizado.

```
cd $HOME
```

```
ls /dev/tty* && ls /dev/tty??
```

Desde tu **directorio personal** (/home/tu_usuario), **crea cuatro subdirectorios**.

- ventasdelaempresa
- marketingdelaempresa
- desarrollodelaempresa
- direcciondelaempresa

```
mkdir ventasdelaempresa marketingdelaempresa desarrollodelaempresa direcciondelaempresa
```

En caso de tener **mkdir subdirectorioA/subdirectorioB/subdirectorioC** → El **subdirectorioC** se crea dentro de **subdirectorioB** y este a su vez, dentro de **subdirectorioA**.

Sitúate en el directorio **ventasdelaempresa** y crea **tres ficheros** (enero.txt, febrero.txt y marzo.txt) con algo de contenido.

```
cd ventasdelaempresa/
```

```
cat > enero.txt
```

```
texto del fichero enero (+Intro)
```

```
CTRL+D
```

Sitúate en tu directorio personal y copia en **marketingdelaempresa** todos los ficheros de **ventasdelaempresa**. ¿Puedes hacerlo con sólo un comando?

```
cd $HOME
```

```
cp ventasdelaempresa/* marketingdelaempresa/
```

Con sólo un comando → **cp ventasdelaempresa/* ./marketingdelaempresa/**

Desde **marketingdelaempresa**, crea un directorio **ventasactuales** dentro de **ventasdelaempresa** y mueve los tres ficheros de **ventasdelaempresa** a este directorio.

```
cd marketingdelaempresa/  
mkdir ../ventasdelaempresa/ventasactuales  
mv ../ventasdelaempresa/* ../ventasdelaempresa/ventasactuales
```

Evitar que intente copiar la carpeta **ventasactuales** dentro de sí misma → **mv ../ventasdelaempresa/*.txt ../ventasdelaempresa/ventasactuales)**

Desde **ventasdelaempresa**, crea un directorio **ventasanteriores** dentro de **ventasdelaempresa** y copia los ficheros de **marketingdelaempresa** que terminen por “ero”.

```
cd ../ventasdelaempresa/  
mkdir ventasanteriores  
cp ../marketingdelaempresa/*ero.txt ventasanteriores/
```

O bien → **cp ../marketingdelaempresa/*ero.txt ../ventasanteriores/**
cp ../marketingdelaempresa/*ero* ../ventasanteriores/

Posicionado en **desarrollodelaempresa**, copia el directorio **ventasdelaempresa/ventasanteriores** en el directorio **desarrollodelaempresa/ventasinicio**. Este directorio tendrás que crearlo previamente.

```
cd ../desarrollodelaempresa/  
cp -r ../ventasdelaempresa/ventasanteriores ventasinicio/  
O bien → cp -r ../ventasdelaempresa/ventasanteriores ../ventasinicio/  
Copiar el contenido → cp ../ventasdelaempresa/ventasanteriores/* ../ventasinicio/
```

Posicionado en **desarrollodelaempresa**, borra el directorio **ventasdelaempresa/ventasanteriores**.

```
rm -rf ../ventasdelaempresa/ventasanteriores
```

Posicionado en tu **directorio personal**, renombra el directorio **desarrollodelaempresa/ventasinicio** como **desarrollodelaempresa/ventashistoricas**. Para renombrar usaremos el comando mv, reflexiona porqué.

```
cd $HOME  
mv desarrollodelaempresa/ventasinicio/ desarrollodelaempresa/ventashistoricas/
```

En **ventasdelaempresa**, renombra el directorio anterior **desarrollodelaempresa/historicodeventas**.

```
cd ventasdelaempresa/  
mv ../desarrollodelaempresa/ventashistoricas/ ../desarrollodelaempresa/historicodeventas/
```

Crea un fichero que contendrá una lista de personas llamado **personas.txt**. Visualiza el contenido del fichero mediante comandos.

```
cat > personas.txt
```

```
CTRL+D
```

Añade una persona con tu nombre y otra con nombre Pascual mediante comandos.

```
echo "Tunombre" > personas.txt
```

```
echo "Pascual" >> personas.txt
```

Muestra el contenido del fichero ordenado alfabéticamente en **orden inverso**. Puedes usar el comando **sort** (lo del orden inverso lo resuelve un parámetro de sort...).

```
sort -r personas.txt
```

En el directorio **ventasdelaempresa** crea un fichero llamado **concatenados.txt** que contenga la concatenación del contenido de todos los ficheros de **ventasactuales**, esto es, el contenido de todos los ficheros, uno detrás de otro. Puedes hacerlo con 3 mandatos, ¿podrías hacerlo con 1?

```
cat ventasactuales/* > concatenados.txt
```

Crea en **direcciondelaempresa** un fichero llamado **concatena2.txt** cuyo contenido sea el contenido de todos los ficheros que hay en **desarrollodelaempresa/historicodeventas**.

```
cat ../desarrollodelaempresa/historicodeventas/* > concatena2.txt
```

Guarda una lista de todos los ficheros que empiezan por **"lib"** y estén ubicados en los subdirectorios de la carpeta **/lib**, en un fichero que se denomine **librerias.txt** dentro de **desarrollodelaempresa**.

```
find /lib/ -name "f*" -type f > ../desarrollodelaempresa/librerias.txt
```

Muestra por pantalla el **número total de líneas del fichero librerias.txt**. En función de la versión de Ubuntu, el resultado será diferente.

```
wc -l ../desarrollodelaempresa/librerias.txt
```

Muestra las líneas que tengan la cadena de caracteres **"kernel"** del fichero **librerias.txt**

```
cat librerias.txt | grep "kernel"
```

Muestra las líneas que ocupan **desde la posición 2 a la 5** del fichero **librerias.txt**

```
cat librerias.txt | head -5 | tail -4
```

Reflexiona sobre las **diferencias entre | y &&**.

```
Usando tubería "|", los comandos enlazan la salida de uno con la entrada del siguiente.
```

Muestra las líneas del fichero que contenga la palabra “**Linux**”

```
grep "Linux" ./lineasdetexto.txt
```

```
$ grep "Linux" ./lineasdetexto.txt
```

```
$ cat ./lineasdetexto.txt | grep "Linux"
```

Muestra las líneas del fichero que contenga la cadena “**gnu**”. En este caso, nos da igual si aparece en mayúscula o minúscula

```
grep -i "gnu" lineasdetexto.txt
```

```
$ cat ./linasdetexto.txt | grep -i "gnu"
```

Muestra las líneas del fichero que **no contenga** la cadena “**1994**”

```
grep -v "1994" lineasdetexto.txt
```

```
$ cat ./linasdetexto.txt | grep -v "1994"
```

Muestra las líneas del fichero que **no contenga** la cadena “**1994**” y muestra el **número de línea** dentro del fichero para cada resultado

```
grep -vn "1994" lineasdetexto.txt
```

```
$ cat ./linasdetexto.txt | grep -vn "1994"
```

Muestra las líneas del fichero que contengan la cadena “**gnu**”, da igual si aparece en **mayúscula o minúscula**, pero que sea una **palabra completa** (que no sea una subcadena dentro de una palabra)

```
grep -iw "1994" lineasdetexto.txt
```

```
$ cat ./linasdetexto.txt | grep -iw "gnu"
```

Muestra las líneas del fichero que contengan la cadena “**Esta**”, pero en la **primera posición de la línea**

```
grep "^Esta" lineasdetexto.txt
```

```
$ cat ./linasdetexto.txt | grep "^Esta"
```

Muestra las líneas del fichero que contenga la subcadena “**ta**”, precedida de 2 caracteres cualesquiera

```
grep "..ta" lineasdetexto.txt
```

```
$ cat ./linasdetexto.txt | grep "..ta"
```

Muestra las líneas del fichero que contenga la **letra “q” o la “f” seguidas de “ue”** (es decir, una expresión regular que nos permitirá construir las cadenas “**que**” y “**fue**”)

```
grep "[qf]ue" lineasdetexto.txt
```

```
$ cat ./linasdetexto.txt | grep "[qf]ue"
```

Muestra las líneas del fichero que contengan **cualquier letra seguida de “ue”** (es decir, una expresión regular que nos permitirá construir, entre otras, las cadenas “**que**”, “**fue**”, “**cue**”, ...)

```
grep "[a-z]ue" lineasdetexto.txt
```

```
$ cat ./linasdetexto.txt | grep "[a-z]ue"
```

Muestra las líneas del fichero que contengan un **número indeterminado** (de 0 a n veces) de **letras de la “a” a la “z”**, seguidas de **1 o 2 números del “0” al “9”**.

```
grep "[a-z]*[0-9]\{1,2\}" lineasdetexto.txt
```

```
$ cat ./lineasdetexto.txt | grep "[a-z]*[0-9]\{1,2\}"
```

Crea un subdirectorio en tu carpeta personal llamado **“BusquedaRekursiva”**. **Copia** en él, el fichero **lineasdetexto.txt**, le puedes llamar **“copialineasdetexto.txt”** para identificarlo mejor. Situado en tu carpeta personal **busca todos los ficheros**, incluyendo subdirectorios, que contengan la palabra **“controversia”**. Haz la búsqueda mediante un comando.

```
grep -r "controversia" ./*
```

```
$ mkdir BusquedaRekursiva
```

```
$ cp ./lineasdetexto.txt ./BusquedaRekursiva/copialineasdetexto.txt
```

```
$ grep -r "controversia" ./*
```

Muestra las columnas nombre, apellido, código postal y numero (columnas 5, 6, 2 y 1)

```
awk '{print $5,$6,$2,$1}' formatotabla.txt
```

```
$ cat ./formatotabla.txt | awk '{print $5, $6, $2, $1}'
```

Recupera el nombre y apellido de las personas que tengan la cadena **“46022”**. En este caso, serán aquellas cuyo código postal sea el 46022

```
awk '$2 == 46022 {print $5,$6}' formatotabla.txt
```

```
$ cat ./formatotabla.txt | grep "46022" | awk '{print $5, $6}'
```

Recupera la **matricula** (columna 4) de aquellas líneas que tengan, cualquier **numero de 1 cifra**, seguido de **“41”**, **después de dos caracteres cualesquiera** y, a **continuación, la letra “D”**

```
awk '$4 == [0-9]41..D/ {print $4}' formatotabla.txt
```

```
$ cat ./formatotabla.txt | grep "[0-9]41..[D]" | awk '{print $4}'
```

Muestra el **número de las personas** (1 columna), **nombre, y apellidos** (5 y 6) y el **género** (3 columna), teniendo en cuenta que el género (H hombre, M mujer) viene concatenado con un **“-”** y 3 números, primero de aquellas personas con género **“M”** y después de aquellas con género **“H”**

```
$ grep "M-..." ./formatotabla.txt | awk '{print $1, $5, $6, $3}' && grep "H-..." ./formatotabla.txt | awk '{print $1, $5, $6, $3}'
```

```
Solución alternativa 1er mandato: $ cat ./formatotabla.txt | grep "[M]-" | awk '{print $1, $5, $6, $3}'
```

```
Solución alternativa 2º mandato: $ cat ./formatotabla.txt | grep "[H]-" | awk '{print $1, $5, $6, $3}'
```

Muestra el nombre y apellido de aquellas personas cuya **matricula** contine un **“7”** en la **primera posición** de los números de la matricula, pero en su **nombre y apellidos no aparece una “e”**

```
awk '$1 ~ /^7/ && $5 !~ /e/ && $6 !~ /e/ {print $5, $6}'
```

```
$ cat ./formatotabla.txt | grep "[7]..." | grep -v "[e]" | awk '{print $5, $6}'
```