

LINUX

Sistema de Ficheros.

Implantación de Sistemas Operativos.





1. Para directorios.

1.a. Comando ls -l

1.b. Otras opciones del comando ls -l

1.c. Otras órdenes para directorios.

2. Órdenes para ficheros.

3. Redireccionamiento y tuberías



1. Para directorios.

mkdir [opciones] directorios...

Crea directorios.

Opción:

- **-p** : Crea varios directorios a la vez.

cd

Cambia de un directorio a otro.

pwd

Muestra la ruta completa del directorio actual donde nos encontremos.

ls [opciones] [fichero]

Opciones:

- **-F** : visualiza a la derecha del fichero un (*) si es un ejecutable, (/) si es un directorio y (@) si es un enlace simbólico.
- **-a** : lista TODOS los ficheros, incluso los llamados ocultos, aquellos que empiezan por un punto.
- **-R** : listado recursivo.
- **-l** : formato largo.

Fichero:

- Se puede indicar el directorio que se quiere listar: **ls /dev**
- Se puede indicar el fichero o ficheros que se quieren listar, para cuyo caso se emplean los **METACARACTERES**.



1.a. Comando ls -l.

```
drwxr-xr-x 2 jantpf jantpf 4096 mar 25 12:41 Documents/
```

Tipo:

- d → directorio.
- - → archivo normal.
- b → archivo tipo bloque (dispositivos de almacenamiento masivo)
- e → archivo tipo carácter (dispositivos periféricos de E/S)
- l → enlace simbólico.

Permisos del propietario

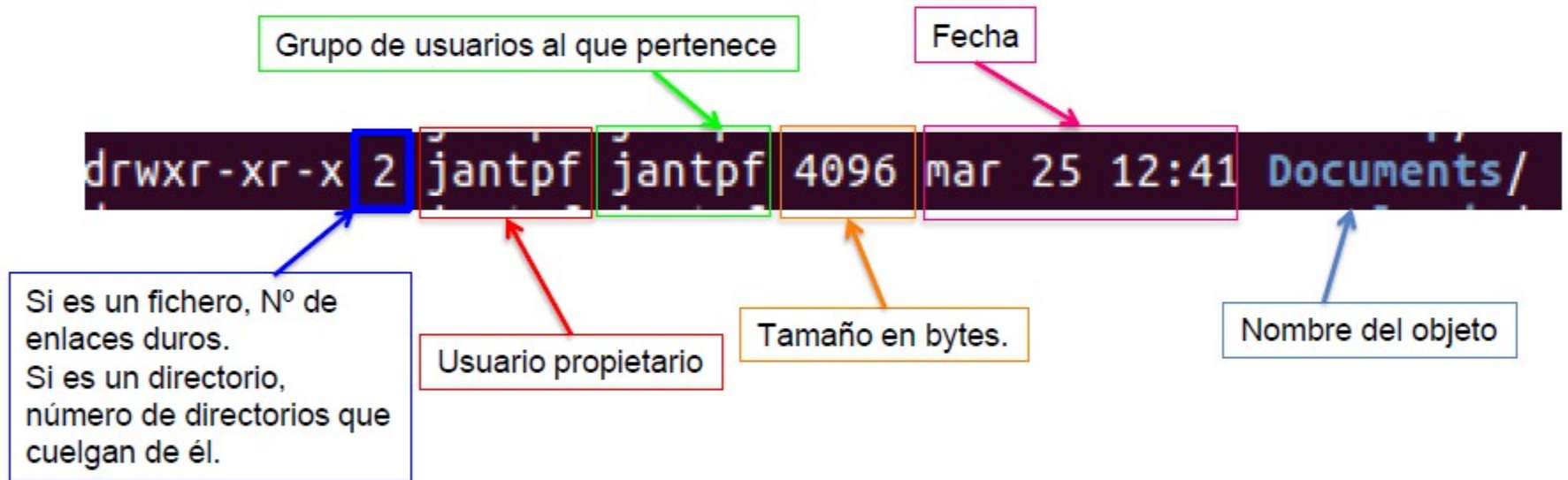
Permisos del grupo

Permisos para otros usuario

```
drwxr-xr-x
```



1.a. Comando ls -l.





1.b. Otras opciones del comando ls -l.

- Opción **-h** : muestra el tamaño en formato de KB, MB, GB, ... en lugar de en bytes.
- Opción **-a** : muestra todos los archivos y directorios, incluidos los archivos ocultos.

Probad:

```
ls -l -h
```

```
ls -lh
```




1.c. Otras órdenes para directorios.

rmdir [opciones]... [directorio]...

- Borra uno o más directorios siempre y cuando no estén vacíos.

Opción: **-p**: permite borrar un directorio, y después intenta borrar su directorio padre si está vacío.

EJERCICIOS

1. Crea el directorio **uno** e intenta borrarlo.
2. Crea los directorios **3 1 4 9** y elimínalos en una sola sentencia.
3. Intenta borrar el directorio **."** de cualquier directorio. ¿Qué ocurre?
4. Crea los directorios **temporal/A/B** y **temporal/A/C**
5. Intenta remover el directorio **C** junto con su directorio padre utilizando la opción **-p** ¿Qué ocurre?
6. Intenta borrar la carpeta **B** junto con su directorio superior utilizando la opción **-p** ¿Qué ocurre?
7. Crea los directorios **6/7** y **8/9**. Intenta borrar los cuatro directorios en una sola sentencia, utilizando la opción **-p**.



1.c. Otras órdenes para directorios.

mv [opciones] [origen]... [destino]

- Sirve para modificar el nombre de archivos y directorios o para trasladarlos a alguna otra ubicación dentro del árbol de directorios.
- Se parece mucho al comando cp, excepto que borra el archivo original después de copiarlo.
- Sirve para renombrar o trasladar.
- Si se utiliza para trasladar, el destino SOLO puede ser un directorio.

Ej: **mv** quijote sancho dulcinea /home/micarpeta/laMancha

¡¡IMPORTANTE!!: Hay que tener cuidado a la hora de usar el comando, pues NO controla si el archivo que queremos renombrar o trasladar existe en el destino. Se puede utilizar la opción **-i** para que nos pregunte antes de sobrescribirlo.



2. Órdenes para ficheros.

touch [opciones] <nombre_fichero>

- Crea un fichero en blanco.

cat [opciones]... <nombre_fichero>...

- Visualiza el contenido de uno o más ficheros por pantalla.
- Ejemplos de uso:
 - **cat** file1.txt > Visualiza el contenido de un fichero.
 - **cat** >file1.txt > Crea un nuevo archivo. Presiona Ctrl+d para finalizar la creación.
 - **cat** >>file1.txt > Añade información a un archivo.



2. Órdenes para ficheros.

nano [opciones] <nombre_fichero>

- Editor para la creación/modificación de ficheros.

cp [opciones]... <fichero_origen>... <destino>

- Permite copiar archivos y directorios dentro de la jerarquía de directorios del sistema. NO borra ningún fichero.
- Puede copiar múltiples archivos entre directorios.
- Con la opción **-i**, preguntará si se desea sobrescribir un archivo ya existente.
- Con la opción **-R**, copia directorios recursivamente.
- También se puede copiar un archivo y cambiar su nombre en una sola sentencia.

```
cp dir1/fichero1.txt nombreNuevo.txt
```



2. Órdenes para ficheros.

cp [opciones]... <fichero_origen>... <destino>

- Formato1: **cp** fichero1 fichero2

Hace una copia del fichero1 en fichero2. Si fichero2 no existe, lo crea.

- Formato2: **cp** fichero1 fichero2 dir_destino/

Copia todos los ficheros en el directorio destino.

- Formato3: **cp** directorio1 directorio2

Copia los ficheros del *directorio1* al *directorio2*.

Con la opción **-R** se copia recursivamente TODO el contenido de *directorio1*.



2. Órdenes para ficheros.

mv

- Formato1: **mv** fichero1 fichero2

Cambia el nombre a un fichero.

- Formato2: **mv** fichero... dir_destino/

Mueve todos los ficheros al directorio destino.

rm [opciones]... <ficheros>

- Elimina uno o varios ficheros.
- Opción **-i** : Pide confirmación para borrar.
- Opción **-f**: Omite la confirmación.
- Opción **-R**: Borra los archivos de un directorio de forma recursiva.



2. Órdenes para ficheros.

Ejercicios-1.

1. Copia los archivos `/etc/hosts` `/etc/passwd` `/etc/shells` a la carpeta `/tmp`
2. Borra todos los archivos de la carpeta `/tmp` en una sola instrucción.
3. Trata de borrar el archivo `/etc/shells` utilizando la opción `-i`. ¿Qué sucede?
4. Trata de borrar el archivo `/etc/passwd` utilizando la opción `-f`. ¿Qué sucede?
5. Crea el directorio `/tmp/directorio` y a continuación crea varios directorios dentro de él.
6. Intenta eliminar el directorio `/tmp/directorio` con las opciones `-i` y `-r` al mismo tiempo ¿Qué sucede?



2. Órdenes para ficheros.

file [opciones] nombre_fichero

- Determina de qué tipo es el fichero (texto, directorio, ...)

tail [num_lineas] nombre_fichero

- Visualiza las últimas líneas de un fichero. Si no se especifica el número, la opción por defecto muestra las 10 últimas líneas.

```
tail -3 fichero.txt
```

head [num_lineas] nombre_fichero

- Visualiza las primeras líneas de un fichero. Si no se especifica el número, la opción por defecto muestra las 10 primeras líneas.

```
head -3 fichero.txt
```



2. Órdenes para ficheros.

sort [opciones] [nombre_fichero]

- Muestra el contenido de un fichero pero mostrando sus líneas en orden alfabético.
- Si no se indica nombre de fichero, lee la entrada estándar



2. Órdenes para ficheros.

Ejercicios-2

1. Copia el archivo `/etc/shells` `etc/hosts` `etc/passwd` a la carpeta `/tmp`.
2. Crea la carpeta `dir` en `/tmp`.
3. Mueve el archivo `/tmp/shells` a la carpeta `dir` del ejercicio anterior.
4. Mueve el archivo `/tmp/hosts` a la carpeta `dir` y cambie su nombre en una sola instrucción.
5. Utiliza la opción `-i` para mover el archivo `passwd` a la carpeta `dir` especificando que su nuevo nombre debe ser `shells`.
6. Mueve todos los archivos que se encuentran dentro del directorio `dir` a la carpeta `/tmp` en una sola instrucción.
7. Intenta mover la carpeta `/tmp/dir` a su directorio inicial. ¿Qué sucede?



2. Órdenes para ficheros.

Ejercicios-3

1. Crea la carpeta **prueba**. Intenta copiarla a otra ruta.
2. Copia la carpeta **prueba** con el nombre **test**.
3. Crea las carpetas **A** y **B** dentro de la carpeta **test**. Copia la carpeta **test**, la nueva carpeta debe tener el nombre **test2**.
4. Intenta copiar las carpetas **A** y **B** dentro de la carpeta **prueba** en una sola sentencia.
5. Copia todas las carpetas creadas al directorio **/tmp** en una sola instrucción y utilizando la opción **-i**.
6. Intenta sobrescribir la carpeta **prueba** por la carpeta **test** usando la opción **-f**.
7. Intenta utilizar ambas opciones **-i** y **-f** al mismo tiempo. ¿Qué ocurre?
8. Repite el ejercicio 3 con las opciones **-r** y **-i** al mismo tiempo. ¿Cuántas veces has tenido que confirmar el reemplazo de carpetas?
9. Repite el ejercicio 3 con las opciones **-r** y **-f** al mismo tiempo.



2. Órdenes para ficheros.

Ejercicios-4

1. Crea el directorio `directorio1`.
2. ¿Qué se mostraría con el comando `ls -a` sobre esta carpeta?
3. Intenta volver a crear el `directorio1` ¿Qué ocurre?
4. Intenta crear el directorio `uno/dos` sin especificar la opción `-p`.
5. Crea los directorios `1 2 3 4` y `5` en una sola sentencia.
6. Crea los directorios `6/7` y `7/8` dentro del directorio `1` en una sola sentencia para ambos.
7. ¿Cuántos niveles de directorio existen en la trayectoria `7/8/`?



2. Órdenes para ficheros.

Realizar la actividad 1 del campus.



2. Órdenes para ficheros.

cmp [opcion]... archivo1 [archivo2]

- Compara dos ficheros cualesquiera.
- Devuelve 0 si son iguales, 1 si son diferentes y 2 si hay algún problema.
- Si como primer archivo usamos “-” en lugar de fichero, se compara la entrada estándar con el fichero2.
- La comparación se realiza hasta el final de los archivos, siempre y cuando la cantidad de bytes sea la misma. En caso contrario, finaliza cuando encuentra la primera marca de fin de archivo, indicando con qué archivo ocurrió.

Opciones:

- l : lista el número de carácter y los valores diferentes para cada una de las diferencias de los ficheros.
- b : imprime los bytes como texto.



2. Órdenes para ficheros.

diff [opcion]... archivo1 [archivo2]

- Realiza la comparación entre dos archivos, generando las diferencias o los cambios realizados en un archivo determinado comparándolo con una versión anterior del mismo archivo.
- El funcionamiento de “-” es el mismo que en la orden anterior.

Opciones:

-w : ignora las líneas que solo se diferencian en espacios en blanco.

-i : ignora la diferencia entre mayúsculas y minúsculas.

Ejemplos de uso:

```
diff [-wi] fichero1 fichero2
```

```
diff [-wi] – fichero2
```



2. Órdenes para ficheros.

WC [opcion]... archivos

- Devuelve el número de líneas, palabras y caracteres que contiene un fichero.

Ejemplo:

```
wc [-lwc] nombre_archivo
```

- Si no se indica ningún parámetro, se mostrarán 4 columnas que indicarán el número de líneas, de palabras y de caracteres, además del nombre del archivo procesado.
- Si se procesan varios archivos al mismo tiempo, al final se mostrará la cuenta total.

Opciones:

- **-w** : Indica el número de palabras.
- **-l** : Indica el número de líneas.
- **-c** : Indica el número de caracteres.

Ejercicios:

1. Muestra el número de palabras, líneas y caracteres del archivo `/etc/hosts`
2. Muestra el total de palabras, líneas y caracteres de los archivos:

```
/etc/hosts /etc/passwd /etc/shells.
```




find lista_de_directorios [expresion_de_busqueda]

- Busca ficheros.
- Le tenemos que indicar el directorio donde vamos a realizar la búsqueda y una expresión que se refiera a lo que queremos buscar.

grep [opciones] expresión [archivos]

- Busca las líneas de un fichero que contienen una determinada cadena o expresión. También puede buscar entre los resultados de un **find** o de un **ls**.

Opciones:

- n : muestra el número de línea.
- i : ignora mayúsculas de minúsculas.
- v : muestra las líneas que no coinciden.
- w : obliga a que coincida el patrón buscado.

Ejemplo: `ls -l | grep 2017`



ACTIVIDADES



3. Redireccionamiento y tuberías.

- Uno de los principios de la filosofía [Unix](#) consiste en tener aplicaciones muy pequeñas que hagan tareas muy puntuales y que usadas en conjunto puedan realizar tareas complejas. Siguiendo este principio debe existir una forma para que varias aplicaciones puedan interactuar entre sí, aquí es donde entran las tuberías.
- Una tubería en Linux no es más que una forma práctica de **redireccionar la salida estándar de un programa** hacia la entrada estándar de otro. Esto se logra usando el símbolo **|** (pipe).
- Por lo general, las órdenes en Linux envían la salida hacia la pantalla. Por ejemplo, si hacemos **ls**, se presentan en pantalla un listado de todos los ficheros y directorios.
- La entrada por defecto viene del teclado, aunque también se puede toma como entrada el contenido de un fichero o el resultado/salida de un programa.
- Tanto la entrada como la salida, se pueden redirigir.



3. Redireccionamiento y tuberías.

Uso	Resultado
orden < fichero	La orden toma su entrada estándar del fichero. <code>\$ cat < /etc/passwd</code> <code>\$ wc < /usr/share/dict/spanish</code>
orden > fichero	La orden deposita su salida estándar en el fichero, al principio de éste y borrándolo si ya existía. <code>\$ ls -l /etc > lista</code> <code>\$ ls -la /etc > lista</code>
orden >> fichero	La orden añade lo que escribe por su salida estándar al final del fichero, creándolo si no existía. <code>\$ echo "Esta línea se añade al final" >> lista</code>
orden << etiqueta	La orden toma su entrada estándar del shell, que nos va pidiendo líneas hasta que llegue una que solo contenga la etiqueta. Ej: <code>\$ cat << fin > mi_texto</code> <code>> Los ficheros de mi directorio son:</code> <code>> \$(ls -l)</code> <code>> fin</code>
orden1 orden2	La segunda orden toma su entrada estándar de la salida estándar de la primera.. <code>\$ ls -l /etc grep '^d' wc -l</code>



4. Filtros.

- Los **filtros** nos permiten personalizar o modificar la salida de la ejecución de un comando de diversas maneras.

more [opciones] archivo ...

- Permite visualizar uno o más archivos pantalla a pantalla.
- Para pasar de una pagina a otra presionaremos la barra espaciadora.
- Para abandonar la visualización del fichero que se está presentando, presionaremos “q” para salir.
- Formato para utilizarlo en tuberías: comando | **more** [opciones]

Opciones:

+num_linea Comienza a mostrar el archivo a partir de la línea indicada.

+cadena Comienza a mostrar el archivo dos líneas antes de encontrar la cadena buscada.

Ejemplo:

```
more /etc/passwd
```

```
ls -la | more
```




4. Filtros.

tail [opciones] archivo ...

- Permite visualizar las diez últimas líneas del fichero especificado o de la entrada estándar.
- Sintaxis como filtro: **comando** | **tail** [opciones]

Opciones:

- +numero** Comienza a mostrar desde la línea cuyo número se ha indicado hasta el final del archivo.
- numero** Mostrará tantas líneas del archivo como se indique, empezando a contar desde el final. El número predeterminado es 10.

Ejemplo:

tail /etc/passwd Muestra las últimas 10 líneas.

ls -l | **tail** -3



4. Filtros.

head [-numero] archivo ...

- Permite visualizar las diez primeras líneas del fichero especificado o de la entrada estándar.
- Sintaxis como filtro: **comando** | **head** [-numero]

Opciones:

-numero Indica el número de líneas que se mostrarán desde el principio del archivo..

Ejemplo:

head /etc/passwd Muestra las primeras 10 líneas.

ls -l | **head** -3



4. Filtros.

grep [opciones] expresion [archivos]

- Busca en los archivos especificados las líneas que contengan el patrón de datos especificado. Cada línea encontrada, será copiada en la salida estándar.
- Cuando se empleen los caracteres \$, *, [, ^, |, (,), \ en expresiones de búsqueda, se encerrará entre comillas dobles o simples, pues estos caracteres tienen un significado especial para el shell.
- Sintaxis como filtro: **comando** | **grep** [opciones] expresion

Opciones:

- v Se imprimen todas las líneas excepto las que coinciden con la expresión.
- i Ignora la diferencia entre minúsculas y mayúsculas.
- l Lista los nombres de los archivos donde existen líneas coincidentes.

Ejemplo:

grep "root" < /etc/passwd Busca el patrón "root" y muestra todas las líneas que lo contiene.

ls / | **grep** cd Muestra todos los archivos de / que contengan las letras cd.



EJEMPLOS DE EXPRESIONES REGULARES

<https://drive.google.com/file/d/0B0KzNxqMNZEbmg4TVZ3ZWxFWWM/view>

<code>a.b</code>	<code>axb aab abb aSb a#b ...</code>
<code>a..b</code>	<code>axxb aaab abbb a4\$b ...</code>
<code>[abc]</code>	<code>a b c (cadenas de un caracter)</code>
<code>[aA]</code>	<code>a A (cadenas de un caracter)</code>
<code>[aA][bB]</code>	<code>ab Ab aB AB (cadenas de dos caracteres)</code>
<code>[0123456789]</code>	<code>uno de 0 1 2 3 4 5 6 7 8 9</code>
<code>[0-9]</code>	<code>uno de 0 1 2 3 4 5 6 7 8 9</code>
<code>[A-Za-z]</code>	<code>uno de A B C ... Z a b c ... z</code>
<code>[0-9][0-9][0-9]</code>	<code>tres dígitos 000 001 .. 009 010 .. 019 100 .. 999</code>
<code>[0-9]*</code>	<code>cadena_vacía 0 1 9 00 99 123 456 999 9999 ...</code>
<code>[0-9][0-9]*</code>	<code>0 1 9 00 99 123 456 999 9999 99999 99999999 ...</code>
<code>[0-9][0-9]+</code>	<code>00 01 29 474 99 123 456 999 9999 99999 99999999 ...</code>
<code>^.*\$</code>	<code>cualquier línea completa</code>
<code>^\$</code>	<code>una línea vacía</code>