

2. Comandos básicos de GNU/LINUX

Vamos a ver algunos [comandos](#) básicos de GNU-LINUX que nos permitirán familiarizarnos con el sistema. Para poder utilizar estos comandos ya debemos haber iniciado la sesión. Iniciada la sesión, estaremos delante de un *prompt* (solicitud) que es la línea de comandos de GNU-LINUX. El formato de este *prompt* será similar a

```
user@pablito:~$
```

2.1. Comandos sencillos

date

Obtenemos la fecha y la hora actual.

who

Nos permite ver los usuarios que están utilizando el sistema, incluyendo la fecha en la que se conectaron al sistema.

uptime

Podemos ver hace cuánto tiempo que se ha iniciado el sistema de nuestra computadora.

clear

Nos sirve para limpiar la pantalla.

2.2. Comandos relacionados con archivos y directorios

pwd

Nos muestra la ruta de directorios en la que estamos situados en este momento.

ls

Lista los archivos dentro de un determinado directorio.

touch

touch **archivo** puede tener dos consecuencias: si **archivo** no existe aún, lo crea con tamaño 0 y como propiedad de nuestro usuario. Por otro lado, si **archivo** ya existe, actualiza la fecha de modificación.

cd

Nos permite cambiar de directorio. Por ejemplo, **cd /** nos lleva al directorio raíz, que es

de donde surgen todos los directorios del sistema y `cd ..` nos lleva a la super carpeta.

cp

Se utiliza para copiar archivos. Si escribimos `cp viejo nuevo`, copiaremos el archivo `viejo` con el nombre `nuevo`. Es decir, el archivo origen se escribe primero y a continuación el archivo que se va a crear. Una vez hecha la copia, tendremos dos archivos diferentes, con el mismo contenido. También podemos ejecutar `cp archivo1 archivo2 directorio`, de forma que los archivos `archivo1` y `archivo2` se copiarán dentro de `directorio`.

mv

Se utiliza para mover archivos de un lugar a otro, o para cambiarle el nombre a un archivo. Si ejecutamos, `mv viejo nuevo`, el archivo `viejo` habrá pasado a llamarse `nuevo`. Por otro lado, si ejecutamos `mv archivo1 archivo2 directorio`, los archivos `archivo1` y `archivo2` se moverán dentro de `directorio`.

rm

Sirve para borrar archivos. Hay que usarlo cuidadosamente, porque una vez que los archivos han sido borrados, no pueden recuperarse **de ninguna forma**. `rm r miCarpeta` borra `miCarpeta`.

mkdir

Sirve para crear directorios. Pueden utilizarse rutas absolutas o relativas; es decir, si queremos crear el directorio `/home/user/temp`, y estamos situados dentro del directorio `/home/user`, podremos ejecutar `mkdir temp` o `mkdir /home/user/temp` indistintamente.

rmdir

Sirve para borrar directorios. Solamente funcionará si el directorio está vacío.

cat

Ejecutando `cat archivo` podremos ver el contenido de `archivo`. Este comando puede recibir una serie de archivos, y el resultado será que nos mostrará un archivo a continuación del otro.

find

Nos permite encontrar archivos, utilizando diversas técnicas:

- `find . -name "hola.txt"` encuentra todos los archivos llamados `hola.txt` que se encuentren a partir del directorio actual.

- `find . -size 50k` busca los archivos que ocupan 50 kilobytes a partir del directorio actual. Si se utiliza `find . -size 20c`, buscará los archivos que ocupen 20 bytes.
- `find /home/user -empty` busca todos los archivos que se encuentran vacíos, a partir del directorio `/home/user`.

Ejercicio 2.1.

1. Ejecute el comando **pwd** en el terminal y escriba lo que produce.
2. Ejecute el comando **ls -lsrth** en el terminal y escriba las últimas dos líneas que produce.
3. Cree el directorio CC101. Luego cree una copia de este directorio con el nombre CC101copia.
4. Ejecute el comando **ls -lsrth** en el terminal y escriba las dos últimas líneas que produce.

Ejercicio 2.2.

1. Ejecute el comando **cd** en el terminal. Luego, ejecute el comando **pwd** y escriba lo que produce.
2. Cree el directorio semana10 y dentro de este cree el archivo virus.exe
3. Estando en el directorio semana10 cambie el nombre del archivo virus.exe por virus2.exe
4. Estando en el directorio semana10 ejecute **ls -lsrth** y escriba lo que aparece. Luego, ejecute este mismo comando estando en el directorio que contiene a semana7 y escriba las dos últimas líneas que aparece.

Ejercicio 2.3.

1. Ejecute el comando **ls -lS** en el terminal y escriba las tres primeras líneas que produce.
2. ¿Qué funcionalidad produce la bandera **-S** en el comando anterior?
3. Escriba la versión del comando **ls** de tu computadora.
4. Escriba el(los) nombre(s) del(de los) autor(es) del comando **ls**.

Ejercicio 2.4.

1. Ejecute el comando **date** en el terminal y escriba lo que produce.

2. Escriba la versión del comando date de tu computadora.
3. Escriba el nombre del autor del comando date.
4. Escriba la fecha actual en el formato RFC 2822.