

Linux Commands - Ejemplos

Command: cd

Cambia de directorio, podemos poner ruta absoluta, da igual donde nos encontremos, se moverá a esa carpeta.

```
profesor@cliente-ubuntu:~$ cd /home/profesor
```

o relativa, nos moveremos desde la ruta actual hasta el directorio introducido.

```
profesor@cliente-ubuntu:~/home/ cd profesor
```

Command: mkdir

También permite crear directorios en rutas relativas o absolutas.

```
profesor@cliente-ubuntu:~$ mkdir /home/profesor2
```

```
profesor@cliente-ubuntu:~$ /home/mkdir profesor2
```

Command: rm

Sirve para borrar directorios, con la opción `-r` y archivos.

```
profesor@cliente-ubuntu:~$ rm -r /home/profesor2
```

```
profesor@cliente-ubuntu:~$ rm /home/profesor2/fichero.txt
```

Command: touch

Crea un archivo si no existe.

```
profesor@cliente-ubuntu:~$ touch /home/profesor2/fichero.txt
```

Command: cat

Muestra el contenido de un archivo por pantalla

```
profesor@cliente-ubuntu:~$ cat /home/profesor2/fichero.txt
```

Contenido del fichero

Contenido del fichero

Command: ls

Lista los archivos de un directorio. Con `-l` indica formato lista, y `-a` todos los ficheros, incluso ocultos.

```
profesor@cliente-ubuntu:~$ mkdir /home/profesor2
```

```
-rw-r--r-- 1 root root 100 jan 6 21:19 archivo.txt
drwxr-xr-x 3 root root 4096 jan 7 01:44 carpeta2
drwxr-xr-x 2 root root 4096 jan 6 22:19 carpeta3
```

Command: cp

Permite copiar archivos. Es decir crear un archivo a partir de otro.

```
profesor@cliente-ubuntu:~$ cp /home/profesor2/fichero.txt  
/home/profesor2/fichero.copia
```

Command: mv

Le cambia el nombre a un fichero, en nuestro caso fichero.txt dejará de existir para llamarse fichero2.txt

```
profesor@cliente-ubuntu:~$ mv /home/profesor2/fichero.txt  
/home/profesor2/fichero2.txt
```

Command: chmod

Permite cambiar los permisos de un fichero o directorio. Con - le quitamos permisos y con + le añadimos. Además podemos especificar a qué usuarios les añadimos o quitamos permisos. También se pueden modificar los permisos convirtiendo los permisos de cada fichero a un código octal

```
-rw-r--r-- 1 root root 100 jan 6 21:19 archivo.txt
```

```
profesor@cliente-ubuntu:~$ chmod a+x archivo.txt
```

Le cambiamos los permisos al archivo para todos los usuarios (a) y añadiendo los de ejecución(x).

```
-rwxr-xr-x 1 root root 100 jan 6 21:19 archivo.txt
```

Command: chown

Le cambiamos los propietarios al archivo

```
-rw-r--r-- 1 root root 100 jan 6 21:19 archivo.txt
```

```
profesor@cliente-ubuntu:~$ chown user:user archivo.txt
```

```
-rw-r--r-- 1 user user 100 jan 6 21:19 archivo.txt
```

Command: history

Muestra el histórico de comandos ejecutados. Le podemos especificar el número de comandos.

```
profesor@cliente-ubuntu:~$ history 6
```

Muestra los últimos 6 comandos.

Command: sudo

Permite ejecutar comandos con los privilegios de otro usuario, normalmente como root.

```
profesor@cliente-ubuntu:~$ sudo su
```

Nos pedirá contraseña y podremos entrar realizar acciones como superusuario.

Command: uname

Muestra información del sistema.

```
profesor@cliente-ubuntu:~$ uname -a
```

Muestra toda la información del sistema.

Command: apt

Es un comando para instalar paquetes, es decir programas adicionales para nuestro sistema operativo. Hace referencia a unos servidores desde los cuales realiza la descarga.

```
profesor@cliente-ubuntu:~$ apt-get update
```

Actualiza la lista de servidores.

```
profesor@cliente-ubuntu:~$ apt-get install programa
```

Instala el programa siempre y cuando exista en el repositorio con ese nombre.

Command: tar

Comprime y descomprime archivos. Es útil para realizar copias de seguridad.

```
profesor@cliente-ubuntu:~$ tar -cvf copia.tar  
/home/profesor2
```

Comprime en copia.tar todo el directorio.

```
profesor@cliente-ubuntu:~$ tar -xvf copia.tar
```

Descomprime copia.tar

Command: pwd

Nos muestra la ruta actual, es decir en qué directorio nos encontramos.

```
profesor@cliente-ubuntu:~$ /home/profesor2/pwd
```

El comando devolverá /home/profesor2

Command: find

Busca ficheros en un árbol de directorios.

```
profesor@cliente-ubuntu:~$ find /home -type f -name "*.txt"
```

El comando anterior buscará ficheros (type -f) en la carpeta /home con la extensión .txt

Command: grep

Busca cadenas dentro de ficheros, es decir, un determinado texto en un fichero, o en el resultado de la ejecución de una instrucción.

```
profesor@cliente-ubuntu:~$ grep "usuario" /etc/passwd
```

El comando anterior buscará al "usuario" en el archivo /etc/passwd, no devolviendo nada si no lo encuentra y mostrándolo si lo encontrara.

Command: man

Sirve para mostrar la página de ayuda de distintos comandos, si no existiera siempre podemos mostrar la ayuda con el comando que buscamos añadiendo la opción -h.

```
profesor@cliente-ubuntu:~$ man grep
```

```
profesor@cliente-ubuntu:~$ grep -h
```

Command: ps

Muestra los procesos que se están ejecutando en el sistema. Es interesante combinarlo con grep para así saber si un determinado proceso se está ejecutando.

```
profesor@cliente-ubuntu:~$ ps -edf | grep "proceso"
```

El símbolo | (pipe) Alr Gr+1, nos sirve para encadenar la salida de un comando con la entrada de otro.

Command: kill

Sirve para eliminar un determinado proceso

```
profesor@cliente-ubuntu:~$ kill "proceso"
```

Command: top

Nos da la información de los procesos de nuestro sistema.

```
profesor@cliente-ubuntu:~$ top
```

Command: service

Activa y desactiva servicios que se están ejecutando en nuestro sistema. Linux tiene servicios ejecutando cuando los activamos, es decir, cuando queremos que actúe como servidor. Si quisiera activar Linux como Web Server.

```
profesor@cliente-ubuntu:~$ service apache2 start
```

Si quisiera parar el servidor web..

```
profesor@cliente-ubuntu:~$ service apache2 stop
```

Necesitaré ejecutar estos comandos como superusuario (root), por lo que anteriormente pondré **sudo su**.

Command: df

Informa del espacio de disco usado.

```
profesor@cliente-ubuntu:~$ df
```

Command: du

Estima el espacio ocupado por el directorio.

```
profesor@cliente-ubuntu:~$ du /home/profesor2
```

Command: mount

Monta un dispositivo, esto quiere decir que va a “instalar” ese dispositivo y ponerlo disponible para el usuario. En los sistemas actuales ya no hace falta realizar esta operación.

```
profesor@cliente-ubuntu:~$ mount /dev/usb1 /mnt/pendrive
```

Instalará el pendrive en la carpeta /mnt/pendrive

Command: ifconfig

Nos da la información de la ip de nuestras conexiones.

```
profesor@cliente-ubuntu:~$ ifconfig
```

Command: netstat

Muestra conexiones de red, tablas de rutas, estadísticas..

```
profesor@cliente-ubuntu:~$ netstat
```

Command: nslookup

Muestra el nombre dns de una ip y viceversa

```
profesor@cliente-ubuntu:~$ nslookup www.iessanvicente.com
```

Command: rsync

Más que un comando es un programa para enviar y recibir archivos entre dos servidores u ordenadores de manera remota. Hay que instalarlo adicionalmente para poder utilizarlo.

```
profesor@cliente-ubuntu:~$ apt-get install rsync
```

La siguiente instrucción copiará el archivo.txt en un servidor remote y dentro de la carpeta /home/profesor2

```
profesor@cliente-ubuntu:~$ rsync archivo.txt usuario@  
servidor_destino:/home/profesor2
```

Command: free

Nos dice como se está utilizando los recursos de memoria en nuestro sistema

```
profesor@cliente-ubuntu:~$ free
```

Command: reboot, shutdown

Tenemos que ejecutar todos estos comandos como root es decir como superusuario.

Reboot reinicia el sistema y shutdown lo apaga.

```
profesor@cliente-ubuntu:~$ sudo su reboot
```

```
profesor@cliente-ubuntu:~$ sudo su shutdown -h now
```