

## 1. ls

Este comando lista los ficheros y directorios (¡Recuerda que en GNU/Linux todo es un fichero!)

Veamos un ejemplo:

Si queremos que nos muestre la información ampliada y en columnas, utilizaremos el parámetro '**-l**', y si además queremos que nos muestre los elementos ocultos, usaremos '**ls -la /home**'

## 2. cd

Con este comando nos podemos mover entre diferentes directorios. Si nos queremos ir a un directorio el particular:

Si en cambio nos encontramos en un directorio, y queremos pasar al inmediatamente superior:

O bien ir directamente a otro directorio que esta al mismo nivel que el nuestro:

## 3. mkdir

Ya que hablamos de directorios, con este comando podemos crear los que deseemos, como si no hubiera un mañana.

Simplemente:

```
1 mkdir LaLigaDeLaJusticia
2 mkdir LaLigaDeLaJusticia/Batman
```

## 4. rmdir

Como si fuéramos un archi enemigo del universo DC, podemos borrar del mapa un directorio, por ejemplo:

```
1 rmdir /LaLigaDeLaJusticia/GreenLantern
2 #(Total, a quien le importante Linterna Verde)
```

## 5. touch

Con este comando podemos crear ficheros. Su uso es el siguiente:

```
1 touch fichero1 fichero2 fichero3
```

No hace falta que os diga que dichos ficheros están vacíos.

## 6. rm

Al igual que con touch podemos crear nuevos ficheros, con rm, su antítesis (lo siento, a veces me salen estos palabros), podemos eliminarlos de la faz de la tierra.

Es primo hermano del comando comando '**rmdir**', pero en este caso borra ficheros en vez de directorios.

Un ejemplo de uso:

Es un comando un poco preguntón, si quieres que se dedique a eliminar sin mala conciencia, sólo debemos añadir el parámetro '**-f**', y si además quieres que elimine ficheros y subdirectorios, con '**-r**'

**r'**, tendrás más que suficiente. Vamos lo que viene a ser un “**rm -rf**” (**Nota:** ¡Utilizar este comando con esos parámetros son responsabilidad tuya y solo tuya, eh!)

## 7. mv

¿Quieres mover un fichero o un directorio de lugar? ¿Quieres cambiar el nombre de un fichero?  
¡Este es tu comando!

Su uso es muy sencillo:

```
1 mv /facturas/porpagar/factura1.pdf /facturas/yapagadas/
```

O bien:

```
1 mv /Avengers/PeterParker.jpg /Avengers/Spider-Man.jpg
```

## 8. rename

Cambia el nombre de un fichero o conjunto de ficheros. Tiene un buen puñado de parámetros interesantes.

Su uso:

```
1 rename 's/.jpeg/.jpg/' *
```

De esta manera indicamos que en la ubicación se cambiarán todas los ficheros con extensión “jpeg” por “jpg”

## 9. man

Con este comando podemos consultar el manual, de ahí que se llame “man”

Si lo utilizamos seguido del comando a consultar, nos mostrará su entrada en el manual.

Por ejemplo:

## 10. info

Es un comando similar al de ‘man’ con información ampliada sobre el comando a consultar.

Su uso es igual de simple:

## 11. whatis

Un comando poco conocido, realmente muy útil.

Se encarga de buscar el contenido de la palabra indicada, en una base de datos propia, que contiene breves descripciones de los comandos.

Por ejemplo:

```
1 [david@servcentos1 ~]$ whatis man
2 man (1)                - una interfaz de los manuales de referencia electrónicos
```

## 12. clear

Este comando se encarga de borrar la pantalla. No hace mucho más. Para utilizar solo hemos de escribir 'clear'

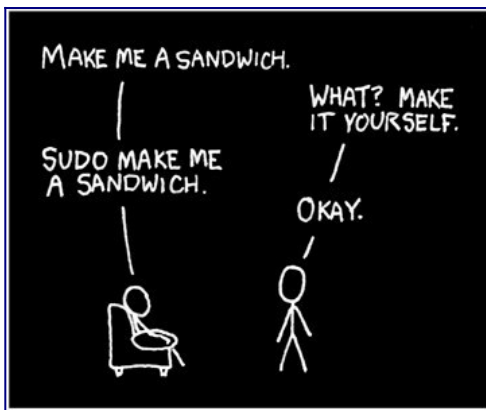
## 13. sudo

Alguno o alguna se puede haber enfadado de que todavía no hayamos hablado de él, pero no os preocupéis, el puesto 11 no es un mal puesto.

Con este comando nos podemos otorgar los poderes de super usuario, siempre que tengamos permisos para ello.

```
1 # make install pastel
2 ¡No tienes permisos!
3 #sudo make install pastel
4 ¡No te olvides apagar el horno!
```

Ya os expliqué en una entrada a trabajar con [sudo](#).



## 14. history

Este es uno de los comandos, que sobre todo al principio, utilizarás más a menudo. Se encarga de mostrar un historial de todos los comandos que has utilizado.

Puedes consultar la entrada que le dediqué al comando [history](#)

## 15. pwd

Nos muestra el nombre del directorio de trabajo actual.

Un ejemplo:

```
1 [david@servcentos1 ~]$ pwd
2 /home/david
```

## 16. cat

Muestra el contenido de un fichero dado. Si se utiliza con varios ficheros a la vez, mostrará su contenido de manera secuencial.

Su uso:

```
1 cat GuiaDelAutoestopistaGalactico-CopiaLegal-eh.txt
```

## 17. **chmod**

Se encarga de cambiar los permisos de acceso a los ficheros.

Le dediqué una entrada a su uso, por si os interesa: “[Permisos de archivos en Linux](#)”

## 18. **chown**

Cambia el usuario y grupo propietarios de ficheros

Un ejemplo de su uso:

```
1 chown david:familia fotos-de-vacaciones.tar.gz
```

## 19. **find**

Busca un directorio o ficheros específico en el sistema de ficheros. Tiene un larga lista de opciones.

También le dedique una entrada, que comparte con el siguiente comand, llamada: “[Búsqueda de ficheros en carpetas en GNU/Linux](#)”

## 20. **locate**

Similar al comando ‘find’, se encarga de buscar en el todo el sistema, ficheros o directorios que coincidan con una consulta. Por forma predeterminada busca únicamente en los ficheros que tiene permisos. A diferencia de ‘find’ tiene su propia base de datos de consulta.

## 21. **wget**

Descarga el fichero o página web dada, indicando la URL

Ejemplo de uso:

```
1 wget https://www.ochobitshacenunbyte.com/
```

Le dediqué un artículo a su uso.

## 22. **grep**

Busca en uno o más ficheros una cadena determinada de texto. Si encuentra la cadena nos indica donde está. Es un comando muy potente, muy utilizado por DevOps y desarrolladores.

Un ejemplo:

```
1 cat /etc/passwd | grep -i davidochobits
```

Utilizando el parámetro ‘-i’ ignoramos la diferencia entre mayúsculas y minúsculas.

## 23. **tail**

Imprime las diez últimas líneas de un fichero.

Es muy utilizado en la consulta de ficheros de registro.

Por ejemplo:

```
1 tail -f -n 20 /var/log/httpd/httpd.log
```

De esta manera lista las últimas líneas de registro del fichero httpd.log, con el parámetro “-f”, indicamos que queremos ver la actividad del registro “en directo”, con “-n” indicamos que queremos ver siempre las 20 últimas líneas.

#### 24. head

Al igual que tail nos mostraba por defecto las diez últimas líneas, con head se nos muestran las diez primeras líneas.

#### 25. dpkg

Es la herramienta habitual para trabajar con los paquetes con extensión **.DEB**, esto es, el gestor de paquetes de Debian. Su fin es instalar, compilar, eliminar y manipular los paquetes de Debian.

Podéis consultar su [página](#) dedicada en el proyecto **Debian**

#### 26. rpm

Es un potente administrador de paquetes con extensión **.RPM**, como bien dice su nombre “Red Hat Package Manager”, sirve para gestionar los paquetes de RHEL y similares, en distribuciones como Centos, Fedora o ScientificLinux

En la [Wiki de Centos](#) hay un buen artículo al respecto

#### 27. free

Muestra la memoria utilizada y disponible en el sistema, incluyendo la swap.

Sus parámetros más habituales, por ejemplo, “-m” que nos muestra la información en megabytes, o “-k” que nos la muestra en kilobytes.

Un ejemplo:

#### 28. df

Nos informa de la utilización de disco en un sistema de ficheros. Este comando se encarga de mostrar el espacio usado y del disponible en todos los sistemas de ficheros montados.

Mi parámetro favorito es ‘-h’ que muestra información para humanos, vamos más entendible.

Su uso habitual:

#### 29. du

Muestra el espacio estimado ocupado por los ficheros y directorios.

Mis parámetros favoritos para este comando, son ‘-s’, que nos muestra únicamente el espacio utilizado real, no el de sus subdirectorios, en el caso de un directorio. Y ‘-h’, que muestra el tamaño en megabytes

#### 30. lsblk

Muestra información de los dispositivos de bloques, como son los discos duros, volúmenes y grupos de volúmenes

Su uso habitual:

Con “-f” se muestra información ampliada y con “-m”, muestra información de su propietario y los permisos de lectura.

### 31. fdisk

Muestra información de los discos duros y los discos lógicos, llamados particiones.

El uso más habitual es:

De estos dos últimos comandos os hablé en el artículo: “[Ubuntu: Cómo ver particiones](#)”

### 32. top

Sí, es verdad, este comando es de los más utilizados, el puesto 32 no le hace justicia, ¡pero es que ahora le tocan a los procesos!

Top se encarga de mostrar una lista de los principales procesos y su uso de CPU entre otros valores.

Su uso es bien simple:

Podéis ampliar información de su uso en este [enlace](#).

### 33. ps

Informa del estado de los procesos. Nos muestra una instantánea de los procesos actuales.

Los parámetros habituales, por lo menos lo que yo utilizo son:

### 34. kill

Se encarga de liquidar un proceso dado, su uso es:

```
1 kill -9 pid-del-proceso
```

Su utilizaremos el parámetro “-9”, se encarga de eliminar el proceso y sus procesos hijos.

Si queremos ser como [Thanos](#), aunque a lo bruto, podemos utilizar “[killall](#)”

### 35. mount y umount

Se encarga de montar o desmontar un dispositivo dado sobre el sistema de ficheros.

Un ejemplo de uso es:

```
1 mount recursonfs:/carpeta /carpeta
```

O bien:

### 36. uname

Imprime información del sistema, incluyendo la versión del kernel o núcleo y del sistema operativo.

Si queremos que nos muestra una información ampliada:

### 37. uptime

Indica el tiempo que el sistema encendido

Un ejemplo:

```
1 [david@servcentos1 ~]$ uptime
2 20:04:58 up 0 min,  2 users,  load average: 4,26, 1,19, 0,41
```

### **38. who**

Nos muestra quien está conectando en el sistema. Importante si lo queremos reiniciar y no queremos que nadie se encuentre con la sorpresa.

Un ejemplo de uso:

```
1 [david@servcentos1 ~]$ who
2 root      tty1          2018-12-04 20:04
3 david     pts/0          2018-12-04 20:04
```

### **39. whereis**

Nos indica la ubicación de los binarios de un comando.

Por ejemplo:

### **40. sort**

Ordena el resultado de un texto dado

### **41. more**

Se encarga de paginar texto, mostrando una pantalla cada vez.

### **42. less**

Se trata de un programa similar a “more”, pero más avanzado, ya que permite consultar páginas hacia atrás y hacia delante.