

ls
muestra el contenido de un directorio

echo
hace eco en pantalla

Ejemplo:

```
$ echo hola mundo!
```

cat
muestra el contenido de un fichero

more
muestra el contenido de un fichero haciendo pausas entre pantallas si el fichero es largo

man
muestra la página del manual de un comando

Ejemplo:

```
$ man ls
```

clear
borra la pantalla

cp
copia ficheros y directorios

Ejemplo:

```
$ cp fichero_original fichero_copia
```

mv
mueve ficheros

Ejemplo:

```
$ mv fichero fichero2
```

rm
borra ficheros

Ejemplo:

```
$ rm fichero
```

ln
enlazar (referenciar) ficheros

Ejemplo de enlace "duro" (hardlink):

```
$ ln fichero enlace
```

Ejemplo de enlace "suave" (softlink):

```
$ ln -s fichero enlace_simbólico
```

cd
cambia de directorio de trabajo si no se indica directorio, nos traslada a \$HOME

Ejemplo:

```
$ cd directorio
```

pwd
muestra el directorio de trabajo actual

mkdir
crea directorios

Ejemplo:

```
$ mkdir directorio
```

rmdir
borra directorios (vacíos)

Ejemplo:

```
$ rmdir directorio
```

env

muestra las variables de entorno del programa

head

muestra las n primeras líneas de un fichero (10 por defecto)

Ejemplo:

```
$ head fichero
```

tail

muestra las n últimas líneas de un fichero (10 por defecto)

Ejemplo:

```
$ tail fichero
```

grep

busca ocurrencias de una cadena en un fichero

Ejemplo:

```
$ grep cadena fichero
```

ps

muestra los procesos en el sistema

kill

Envía una señal a un proceso indicando su PID (Process IDentifier, o número único que identifica a cada proceso)

Ejemplo:

```
$ kill 1002
```

export

Exporta una variable al entorno del programa

Ejemplo:

```
$ export VARIABLE=valor
```

read

Lee una linea de la entrada estándar y la almacena en una variable

Ejemplo:

```
$ read linea
```

\$

Delante de una variable permite acceder a su contenido

Ejemplo:

```
$ echo $SHELL
```

;

Separa dos comandos en una misma linea

Ejemplo:

```
$ read linea ; echo se ha leido: $linea
```

file

indica de qué tipo es un fichero

cal

muestra el calendario del mes actual

wc

cuenta líneas, palabras o bytes en ficheros

Ejemplo:

```
$ echo hola que tal | wc
```

date

muestra hora y fecha actuales

Ejemplo:

```
$ date
```

Ejemplo de fecha en formato yyyy-mm-dd:

```
$ date "+%Y-%m-%d"
```

passwd

cambia la contraseña de un usuario

chmod

cambia los permisos de un fichero

chown

cambia el propietario de un fichero

chgrp

cambia el grupo propietario de un fichero

reset

restaura la terminal de texto

whereis

indica donde se puede encontrar un fuente, binario o manual

Ejemplo:

```
$ whereis ls
```

which

indica donde está un comando

Ejemplo:

```
$ which ls
```

locate

busca ficheros

find

búsqueda avanzada de ficheros

who

quién tiene sesión abierta en la máquina

tac

concatena ficheros y los muestra a la inversa

touch

actualiza la fecha y hora de un fichero, si no existe lo crea

Ejemplo:

```
$ touch fichero_inexistente
```

less

una versión más elaborada de **more** que permite desplazarnos por el texto, hacer búsquedas, etc.

df

muestra el espacio libre y ocupados de los discos

du

calcula el espacio de disco usado

mail

programa simple para enviar y leer correo

tar

empaquetar ficheros

Ejemplo empaquetar:

```
$ tar cvf fichero.tar directorio
```

Ejemplo desempaquetar:

```
$ tar xvf fichero.tar
```

gzip

comprimir un fichero

gunzip

descomprimir un fichero comprimido con **gzip**

zcat

muestra el contenido de un fichero comprimido con **gzip**

ldd

muestra las librerías que usa un programa

halt

apaga la máquina

reboot

reinicia la máquina

shutdown

apaga o reinicia la máquina

true

cierto, o uno

false

falso, o cero

exit

termina la sesión y muestra el *login* del sistema

logout

termina la sesión y muestra el *login* del sistema

seq

genera una secuencia de números

Ejemplo:

```
$ seq 1 10
```

cut

elimina partes de ficheros

Ejemplo:

```
$ echo hola que tal | cut -d " " -f 2
```

awk

escáner de patrones y lenguaje de programación para procesar textos

Ejemplo:

```
$ echo hola que tal | awk '{ print $1 "!", $2, $3 "?" }'
```

tr

elimina o traduce caracteres

Ejemplo:

```
$ echo hola que tal | tr a A
```

sed

realiza transformaciones en flujos de bytes

Ejemplo:

```
$ echo hola que tal | sed 's/a/A/g'
```

(substituye las 'a' por 'A' en todo el flujo)

fmt

da formato a cada párrafo de un fichero

sort

ordena ficheros de texto

sleep

detiene el proceso durante n segundos

Ejemplo:

```
$ sleep 5 ; echo Han pasado 5 segundos
```

uniq

lee de **stdin** y compara líneas adyacentes escribiendo las líneas únicas a **stdout**