

## Comandos Linux Básicos de Linux(1)

**Nota:** “[” indica que son opcionales. “...” que puede ir más de una. Las opciones suelen comenzar con “-” o “--”. Si en la explicación del comando se acaba con “...” es porque hay muchas más opciones o información que es mejor consultar en la página del manual.

### ***cd [directorio]***

Cambia de directorio. Sin argumentos lleva al directorio del usuario (HOME). Si el directorio es “..” sube un nivel.

### ***pwd***

Imprime el directorio actual de trabajo.

### ***ls [opciones...] [directorio/fichero ...]***

Lista el contenido del directorio, sin argumentos lista el contenido del directorio actual de trabajo. La opción más habitual es “-l” que muestra información más completa de cada directorio y fichero. La opción “-R” hace un listado recursivo en la jerarquía de directorios.

### ***man [opciones...] página***

Muestra la documentación de un determinado comando (en realidad también de cualquier documentación, no sólo comandos). Por ejemplo “man ls” nos dará todas las opciones del comando ls. Cada documento se denomina “página”, las páginas están divididas por “secciones”. A veces existen páginas en secciones distintas con el mismo nombre, para especificar la sección se usa el número de ella como opción, por ejemplo “man 3 printf”. “man -a printf” mostrará todas las páginas, en sucesión, de printf en todas las secciones. Si se quiere buscar “man -k printf” mostrará un listado resumido de todas las páginas donde aparezca printf en la descripción corta. El número mostrado entre paréntesis es la sección.

Se puede indicar que muestra la documentación en otros idiomas (si están instalados en el sistema). Por ejemplo “man -L ca ...” lo hará en catalán (no hay muchas, hint! hint! administraciones y universidades que quieren hacer algo y tienen poco dinero), “man -L es ...” en castellano, “man -L en ...” en inglés.

### ***cat [opciones...] [ficheros...]***

Muestra el contenido de los ficheros por la “salida estándar”. Si no se especifican ficheros, lee de la “entrada estándar”.

### ***more [opciones...] [ficheros]***

Muestra el contenido de los ficheros o la entrada estándar página a página y espera que el usuario indique las acciones a tomar. Estas acciones se suelen indicar con una tecla, por ejemplo “<ESPACIO>” es para avanzar una página, “<ENTER>” avanza una línea. “h” da la ayuda, “/” sirve para buscar una cadena, “q” para salir...

### ***less [opciones...] [ficheros]***

Viene de la frase “less is more”. Es similar a more pero con la ventaja fundamental que puede ir hacia atrás en el texto. Los comandos son similares y otros nuevos, como “?” para buscar hacia atrás, “b” que retrocede una pantalla, “d” que avanza media pantalla, “u” que retrocede media pantalla...

### ***Redirecciones: > < |***

Todos los comandos de GNU-Linux/Unix tienen asociado una “entrada estándar” (normalmente el teclado), una “salida estándar” (normalmente la consola o pantalla) y una “salida estándar de error” (habitualmente la misma pantalla). Es posible redirigir esas entradas y salidas hacia un fichero o directamente a otro comando.

“**comando ... > fichero**”: redirige la salida del comando hacia el fichero.

“**comando ... < fichero**”: redirige la entrada y lee desde el fichero.

“|”: La redirección “|” (tubería o “pipe” permite “entubar” las salidas a entradas de otros procesos evitando el uso de ficheros temporales y ejecución no sincronizada de comandos.

La concatenación de comandos es muy potente y típica de la filosofía de Unix y Linux. Por ejemplo:

```
cat fichero1 fichero2 | grep palabra | sort | uniq
```

**Envío de salida como argumentos.** Existe otra forma de enviar la salida de un proceso hacia otro, pero en vez de hacerlo a su entrada estándar lo hace como argumentos, se hace así con la comilla simple invertida (la del acento grave en catalán o francés): “`”. Se usa en el lugar donde se especifican los argumentos de comandos:

```
otro_comando `comando `
```

Significa que se ejecutará primero “comando”, su salida será luego los argumentos de “otro\_comando”. Probado: “ls -l `which ls`”.

### ***cp [opciones...] fuente... destino***

Permite copiar un fichero, o varios ficheros a un directorio. Quizás la opción más usada es “-r” que permite copiar recursivamente directorios hacia otros directorios.

### ***mv [opciones...] fuente... destino***

Cambia el nombre de un fichero por otro o mueve una serie de ficheros y directorios a un directorio destino.

### ***mkdir [opciones...] directorio***

Crea el directorio con el nombre indicado.

### ***rmdir [opciones...] directorio...***

Borra un o varios directorios si están vacíos

### ***rm [opciones...] ficheros***

Borra ficheros. Las opciones más habituales son “-f” para forzar el borrado sin preguntar al usuario (la opción contraria es “-i”). “-r” borra recursivamente todos los subdirectorios (¡ALERTA!, típico error de principiantes).

### ***ln [opciones] destino [nuevo\_alias]***

Crea un enlace a un fichero, apuntará a los mismos datos que el fichero “destino”, siempre. La opción “-s” hace que se cree un enlace simbólico, es lo que se conoce como “Acceso Directo” en Windows, o “Enlace” en la interfaz gráfica de Macintosh.

### ***sort [opciones...] [ficheros...]***

Imprime la concatenación ordenada lexicográficamente de los ficheros o entrada estándar. La opción “-n” ordena numéricamente, “-r” en orden inverso. “--field-separator=SEP” hace que SEP sea el separador de campos...

### ***uniq [opciones...] [ficheros...]***

Elimina las líneas consecutivas duplicadas.

### ***head [opciones...] [ficheros...]***

Muestra las 10 primeras líneas de ficheros o la entrada estándar. Si en las opciones se indica “-50” (por ejemplo) mostrará las primeras 50 líneas.

### ***tail [opciones...] [ficheros...]***

Como el head pero las últimas líneas del fichero. Otra opción muy habitual es “-f” que hace que el tail quede verificando si el fichero crece y muestra las nuevas líneas.

### ***wc [opciones...] [ficheros]***

Indica la cantidad de caracteres, palabras y líneas que tienen los ficheros. “-l” indica sólo número de líneas, “-w” palabras y “-c” los bytes, “-m” caracteres.

### ***file [opciones...] [ficheros...]***

Muestra información más detallada del tipo de contenido de los ficheros.

### ***which [-a] comandos...***

Muestra el nombre completo (“pathname”) de los comandos especificados. La opción “-a” muestra todos los que existen para cada uno, no solamente el primero (que es el que se ejecuta).

## Información del sistema y usuarios

### ***date [opciones] [formato]***

Muestra (o cambia) la fecha y hora del sistema.

### ***w [opciones] [usuario]***

Muestra que usuarios están conectados o con sesión abierta. Si se especifica usuario, sólo muestra las conexiones de ese usuario.

### ***whoami***

Parece una pregunta tonta... ¿pero qué usuario eres ahora mismo? (tened en cuenta que es muy fácil cambiar... con el “su”).

### ***su [opciones] [usuario]***

Permite establecer una nueva sesión con otro usuario. Si no se especifica usuario lo hará con el “root”.

### ***last [opciones] [usuarios] [terminales]***

Muestra las últimas sesiones de usuarios.

### ***ps [opciones]***

Linux es un sistema de multiprogramación con muchos procesos ejecutándose “simultáneamente” (probad “ps axl | wc -l”). Las opciones más habituales son: “ax” o “axu” para ver todos los procesos, “l” para ver en formato extendido. Por ejemplo “ps axl”.

### ***pstree [opciones] [usuarios]***

Muestra la jerarquía (de árbol) de los procesos.

### ***top [opciones]***

Muestra los procesos en forma interactiva y refrescada continuamente (la tecla “h” da una ayuda resumida).

### ***mount [...]***

Sin argumentos muestra los sistemas de ficheros montados, su dispositivo correspondiente en /dev y en que directorio se han montado

### ***passwd [opciones...] [usuario]***

Permite cambiar la clave de acceso de un usuario. Si es el root, puede especificar el usuario del que desea cambiar la contraseña.

## Buscar

### ***grep [opciones...] patrón [ficheros]***

Busca la ocurrencia de “patrón” en los ficheros (o salida estándar). Las opciones más habituales son “-l” para imprimir sólo el nombre de los ficheros donde aparece patrón, “-i” para que sea insensible a mayúsculas y minúsculas, “-r” para hacer recursivamente en los directorios...

### ***locate [opciones...] patrón***

Busca nombres de ficheros que cumplan con el patrón en una base de datos que normalmente se actualiza cada día.

### ***find [opciones] expresiones***

Busca ficheros que cumplan determinadas condiciones basados no en los datos, sino en la información que mantiene el sistema operativo, por ejemplo nombre, fechas de creación, modificación, etc... Puede simplemente imprimir por pantalla o ejecutar comandos por cada fichero (opción -exec).

```
find . -name "datos*"
```

## Bucle for

El **for** es una estructura muy usada para el tratamiento individual de muchos ficheros (aunque tiene muchas posibilidades más). Su estructura típica es:

```
for variable in lista
do
    comando1
    comando2
    ...
done
```

Lo que hace es para cada palabra en lista, asigna esa palabra a la variable y ejecuta los comandos indicados entre el do y el done.

Ejemplos:

Imprimir los números del 1 al 100

```
for i in `seq 100`
do
    echo $i
done
```

Imprimir la lista de ficheros que acaben en .jpg

```
for f in *.jpg
do
    echo $f
done
```

O el ejemplo de generar miniaturas de fotografías.

```
for f in *.jpg
do
    echo Generando miniatura de: $f
    convert -resize 200 $f petita_$f
done
```