

# Comandos Linux

El conocimiento de los comandos linux es importante para las tareas de administración y programación. Los comandos linux son necesarios para tener un dominio del sistema operativo. La línea de comandos es la manera más directa de enviar ordenes a su máquina. La línea de comandos de GNU/Linux es más potente que los prompts que puede haber usado con anterioridad.

## Comandos linux de archivos y directorios

Comandos linux de ficheros y directorios (crear y borrar directorios; listar, copiar, renombrar y borrar archivos, crear enlace entre archivos).

<b>ls</b>	: Listar archivos y directorios
<b>cp</b>	: Copiar archivos y directorios
<b>pwd</b>	: Mostrar el nombre del directorio de trabajo actual
<b>cd</b>	: Cambiar de directorio
<b>sort</b>	: Ordenar ficheros
<b>mkdir</b>	: Crear directorios
<b>touch</b>	: Crear o actualizar ficheros
<b>rm</b>	: Borrar archivos y/o directorios
<b>rmdir</b>	: Borrar directorios vacíos
<b>mv</b>	: Mover o renombrar archivos
<b>more</b>	: Mostrar ficheros página a página
<b>less</b>	: Mostrar ficheros página a página
<b>cat</b>	: Mostrar ficheros de forma continua
<b>head</b>	: Ver el inicio de un archivo
<b>tail</b>	: Ver las últimas líneas de un archivo
<b>find</b>	: Buscar archivos
<b>grep</b>	: Buscar el patrón pasado como argumento en uno o más archivos
<b>wc</b>	: Calcular la cantidad de cadenas y palabras en archivos
<b>ln</b>	: Crear enlace entre ficheros

## Comandos linux para programar la ejecución de comandos

Comandos linux para programar la ejecución de comandos una vez o de forma reiterada.

<b>crontab</b>	: Ejecutar comandos linux a intervalos de tiempo regulares
<b>at</b>	: Ejecutar un comando linux una vez

## Comandos linux para archivar y comprimir ficheros

Comandos linux usados para archivar, extraer y comprimir ficheros.

<b>tar</b>	: Almacenar y extraer ficheros de un fichero archivador
<b>gzip</b>	: Comprimir ficheros
<b>bzip2</b>	: Comprimir ficheros

# Comandos linux de atributos y permisos

Comandos linux para atributos y permisos de archivos y directorios.

**chown** : Cambiar el propietario de un archivo  
**chgrp** : Cambiar el grupo propietario de un archivo o grupo de archivos  
**chmod** : Cambiar los permisos de accesos a los archivos y directorios

## Comandos linux para el control de procesos

Comandos linux para el control de los procesos en ejecución, información del estado de los mismos, terminar procesos, establecer prioridad de programas.

**ps** : Informar del estado de los procesos  
**pstree** : Mostrar los procesos en forma de estructura de árbol  
**top** : Mostrar los procesos que se están ejecutando en ese momento, mostrando los que más CPU consumen  
**kill** : Terminar un proceso por ID  
**killall** : Terminar procesos por nombre  
**nice** : Ejecutar un programa con la prioridad de planificación modificada  
**renice** : Alterar la prioridad de los procesos en ejecución

## Comandos linux del Sistema

Comandos linux del sistema (reiniciar y apagar máquina, cerrar sesión de trabajo, uso de memoria, instalar paquetes).

**date** : Mostrar fecha y hora actual del sistema  
**cal** : Mostrar el calendario del mes actual  
**uptime** : Tiempo que lleva encendida la máquina  
**df** : Espacio libre en los discos  
**du** : Espacio usado por los directorios y ficheros  
**free** : Uso de memoria y swap  
**dmesg** : Mostrar mensajes del kernel durante el arranque del ordenador  
**reboot** : Reiniciar la máquina  
**halt** : Apagar la máquina  
**shutdown** : Cerrar el sistema  
**init** : Control de inicialización de procesos  
**exit** : Cerrar sesión actual  
**clear** : Borrar pantalla  
**bc** : Calculadora  
**man** : Formatear y mostrar las páginas del manual en línea  
**uname** : Imprimir información del sistema  
**hostname** : Mostrar o establecer el nombre del sistema  
**domainname** : Establecer el nombre de dominio NIS/YP del sistema  
**mkfs** : Construir un sistema de ficheros de linux  
**fdisk** : Manipular tablas de particiones de linux  
**fsck** : Chequear y reparar sistemas de ficheros  
**mount** : Montar un sistema de ficheros  
**umount** : Desmontar sistema de ficheros  
**rpm** : Instalar los paquetes RedHat

# Comandos linux de administración de usuarios

Comandos linux para las operaciones vinculadas con usuarios (adicionar, borrar y cambiar usuario, crear grupo,etc).

<b>useradd</b>	: Adicionar nuevo usuario
<b>userdel</b>	: Borrar cuenta de usuario
<b>passwd</b>	: Cambiar contraseña de usuario
<b>su</b>	: Cambiar de usuario
<b>logname</b>	: Mostrar el nombre de usuario
<b>id</b>	: Mostrar los datos de identificación del usuario
<b>who</b>	: Mostrar los usuarios del sistema
<b>last</b>	: Mostrar información de los últimos usuarios que han usado el sistema
<b>groupadd</b>	: Crear un nuevo grupo
<b>nail</b>	: Enviar y recibir correo de Internet
<b>pine</b>	: Lector de correo en modo texto
<b>mesg</b>	: Controlar el acceso a escritura a la terminal
<b>write</b>	: Enviar mensaje a otro usuario
<b>wall</b>	: Enviar mensaje a todos los usuarios
<b>w</b>	: Mostrar qué usuarios están conectados a la máquina y qué están haciendo

# Comandos linux de red

Comandos linux para operaciones de red (mostrar estado de la red, configurar red, enviar paquetes, etc.)

<b>netstat</b>	: Mostrar el estado de la red
<b>ifconfig</b>	: Configurar interfaz de red
<b>nmap</b>	: Escanear red
<b>ping</b>	: Enviar paquetes
<b>nslookup</b>	: Consultar servidores de nombres de dominio
<b>telnet</b>	: Comunicación interactiva con otro host
<b>route</b>	: Manipular tabla de enrutamiento IP

# Comando ls

El comando **ls** se utiliza para listar ficheros y directorios; es equivalente al dir de Windows.

## Sintaxis:

**ls [opciones] [archivo | directorio] [archivo | directorio ...]**

Si no se especifica archivo o directorio, el comando **ls** se encargará de listar los ficheros y directorios del directorio actual.

## Opciones:

- **-a** : con esta opción se pueden listar ficheros y directorios, incluyendo los ocultos.
- **-r** : invierte el orden de la operación de listar ficheros.
- **-R** : permite listar ficheros recursivamente.
- **-s** : muestra el tamaño en kilobytes junto a cada archivo listado.
- **-l** : muestra información adicional sobre los archivos listados.
- **-i** : muestra el número de i-nodo junto a cada archivo.
- **-d** : opción para tratar a los directorios como si fueran archivos normales en vez de listar su contenido.

## Ejemplos:

- **ls -R** : lista recursivamente el contenido del directorio actual.
- **ls -is images/ ..** : lista los archivos en el directorio **images/** y en el directorio padre del actual, e imprime, para cada archivo, su número de i-nodo y su tamaño en kilobytes.
- **ls -al images/\*.png** : lista todos los archivos (incluso los archivos ocultos) del directorio **images/** cuyo nombre termina con **.png**. Note que esto también incluye al archivo **.png** si es que existe uno.

# Comando cp

El comando **cp** se emplea para copiar archivos y directorios. El comando **cp** es similar a los comandos copy, xcopy de Windows, y agrega más opciones. Se puede copiar un fichero a un destino dado o copiar varios ficheros a un directorio destino. Si el último argumento se refiere a un directorio existente, el comando **cp** copia cada fichero fuente a ese directorio.

## Sintaxis:

```
cp [opciones] <archivo | directorio> [archivo | directorio ...] <destino>
```

## Opciones:

- **-a** : al copiar archivos y directorios preserva la estructura y atributos de los ficheros originales pero no preserva la estructura de directorio.
- **-d** : copia los enlaces simbólicos como tales en lugar de copiar los archivos a los que apunten, preserva la relación de los enlaces duros entre ficheros fuentes.
- **-R** : copia archivos recursivamente; obligatoria para copiar directorios.
- **-i** : se pide confirmación para sobrescribir archivos.
- **-f** : lo opuesto de **-i**.
- **-p** : preserva los permisos, el propietario, el grupo y los tiempos de los ficheros originales al copiar los archivos y directorios.
- **-P** : forma el nombre de cada fichero de destino añadiendo al directorio destino una barra inclinada y el nombre especificado del directorio origen.
- **-v** : reporta las acciones del comando **cp** al copiar archivos y directorios.

## Ejemplos:

- **cp -i /tmp/imagenes/\* imagenes/**
- **cp -vR docs/ /shared/mp3s/\* miscosas/**
- **cp pepe pupu**

# Comando pwd

El comando **pwd** tiene un sintaxis muy sencilla y se emplea para mostrar directorios; empléelo por ejemplo, para mostrar el nombre del directorio de trabajo actual.

## Sintaxis:

**pwd directorio**

## Ejemplo:

- **\$pwd**  
/home/reina

# Comando cd

El comando **cd** se utiliza para cambiar de directorio; es exactamente igual que en Windows, con alguna funcionalidad extra: si ingresa el comando "**cd**" solo, será llevado de vuelta a su directorio personal, si se ingresa "**cd -**" será llevado al último directorio en el cual estuvo; y "**cd ~Pedro**" le llevará al directorio personal del usuario Pedro. Como usuario no privilegiado normalmente no puede ingresar a los directorios personales de otros usuarios, a menos que esos usuarios lo hayan autorizado explícitamente o esa sea la configuración predeterminada del sistema; o bien es usted **root**.

El comando **cd** puede usar "." para referirse al directorio actual, ".." para referirse al directorio padre del directorio actual y "~" para referirse a su directorio personal.

El comando **cd** no tiene ninguna opción que sea de importancia.

## Sintaxis:

**cd directorio**

**Ejemplo:** Uso del comando **cd** para cambiar de directorio, en este caso del directorio /root al directorio /usr/share/doc/HOWTO:

- **\$ pwd**  
/root
- **\$ cd /usr/share/doc/HOWTO**
- **\$ pwd**
- /usr/share/doc/HOWTO

# Comando sort

La función del comando **sort** es ordenar ficheros de texto a partir de varios criterios. Escribe una concatenación ordenada de todos los ficheros.

## Sintaxis:

**sort [opciones] [archivo(s)]**

El comando **sort** permite ordenar ficheros de manera ascendente comenzando por el primer campo.

## Opciones:

- **-u** : opción para ordenamiento estricto: los campos duplicados se descartan.
- **-f** : ignora capitalización (trata a las minúsculas como mayúsculas).
- **-n** : el ordenamiento que tiene lugar se realiza o tiene lugar sobre datos numéricos.
- **-r** : opción que permite ordenar de forma descendente las líneas de texto.
- **-i** : considera solamente los caracteres imprimibles.

**Ejemplo:** Uso del comando **sort** para ordenar el fichero `texto.txt` de forma estricta y descendente:

- **sort -ur texto.txt**



# Comando **mkdir**

Permitir crear directorios es la función del comando **mkdir**.

## Sintaxis:

```
mkdir [opciones] <directorio> [directorio ...]
```

## Opciones:

Sólo una opción es de interés: la opción **-p**. La misma hace dos cosas:

- Crea directorios padre si es que aún no existían. Si no se especifica esta opción y los directorios padres no existen, el comando **mkdir** fallará, quejándose de que dichos directorios padre no existen.
- Retornará silenciosamente si el directorio que desea crear ya existe; si no especificó la opción **-p**, en tal caso el comando **mkdir** retornará un mensaje de error, quejándose de que el directorio ya existe.

## Ejemplos:

Uso del comando **mkdir** para crear el directorio denominado pepe en el directorio actual:

- **mkdir pepe**

Uso del comando **mkdir** para crear el directorio misc en el directorio imagenes, creando primero este último si es que no existía; también crea el directorio docs en el directorio actual:

- **mkdir -p imagenes/misc docs**

# Comando touch

El comando **touch** está orientado a cambiar la fecha de un fichero. Cambia la fecha de acceso y/o modificación del archivo especificado.

## Sintaxis:

```
touch [opciones] archivo [archivo ...]
```

## Opciones:

- **-a** : cambia solamente el tiempo de acceso.
- **-c** : no crea archivos nuevos al intentar cambiar la fecha.
- **-d, --date=fecha** : usa fecha para producir el cambio de fecha; el formato de fecha que se debe utilizar es: MMDDHHMMAAAA.
- **-m** : cambia solamente la fecha de modificación.

**Ejemplo:** Se creará un archivo vacío (archivo1) en el directorio actual y un archivo vacío (archivo2) en el directorio **imagenes** si dichos archivos no existían:

- **touch archivo1 imagenes/archivo2**

# Comando **rmdir**

Borrar directorios vacios es la finalidad del comando **rmdir**. Como es lógico, antes de utilizar este comando se deben borrar todos los ficheros que contenga, incluidos los ficheros invisibles, el (o los) directorio(s) a eliminar. Si el directorio que se va a borrar contiene algún fichero, cuando se ejecute el comando **rmdir** dará un mensaje de error y no se realizará la operación de borrar directorios.

Para borrar un directorio que no esté vacío utilice el comando **rm**.

## Sintaxis:

```
rmdir [opciones] directorio ...
```

## Opciones:

- **-p** : Con esta opción, si el directorio incluye más de un componente por el camino, se borra.

**Ejemplo:** Borra el directorio prueba si está vacío:

- **rmdir prueba**

# Comando mv

El comando **mv** tiene como finalidad mover o renombrar archivos.

## Sintaxis:

```
mv [opciones] <archivo | directorio> [archivo | directorio ...] <destino>
```

## Opciones:

- **-f** : fuerza la operación; debe hacer uso de esta opción con moderación pues no hay advertencia alguna en caso de que la operación sobrescriba un archivo que ya existe.
- **-i** : lo contrario; pide confirmación antes de sobrescribir un archivo existente; no haga uso de esta opción a menos que sean muchos ficheros o directorios los involucrados en la operación y que se tenga la certeza total.
- **-v** : reportar todos los cambios y la actividad del comando **mv**.

## Ejemplos:

Uso del comando **mv** para mover todos los archivos del directorio `/tmp/pics/` cuyos nombres terminan en `.png` al directorio actual (`.`), pidiendo confirmación antes de sobrescribir cualquier archivo:

- **mv -i /tmp/pics/\*.png .**

Empleo del comando **mv** para cambiar el nombre del archivo `juan` por `pedro`. Si ya hubiera un directorio `pedro`, el efecto del comando **mv** sería mover todo el directorio `juan` (el directorio en sí mismo más todos los archivos y directorios que contenga, recursivamente) dentro del directorio `pedro`. Es suficiente que un usuario no privilegiado tenga derecho de escritura sobre un directorio para que pueda borrar los archivos que se encuentran en el mismo, incluso si dicho usuario no es el dueño de los archivos:

- **mv juan pedro**

Mover, sin pedir confirmación, todos los archivos del directorio actual cuyos nombres comiencen con `archivo` junto con todo el directorio `imagenes/` al directorio `tacho/`, y mostrar cada operación llevada a cabo por el comando **mv**:

- **mv -vf archivo\* imagenes/ tachos/**

# Comando more

El comando **more** es un paginador; permite al usuario ver ficheros largos página por página (pantalla por pantalla). Se prefiere utilizar el comando **less** ya que da más posibilidades que el comando **more**; si quiere ver el fichero completo sin pausa puede usar también el comando **cat**.

## Sintaxis:

```
more [opciones] <fichero>
```

## Opciones:

- **-num** : especifica un entero que indica el tamaño de la pantalla (en líneas).
- **-d** : muestra los diferentes mensajes en pantalla indicándole qué tecla presionar para hacer la operación deseada.
- **+num** : ver ficheros comenzando en el número de línea especificado por num.

## Ejemplo:

- `more /etc/fstab`

# Comando less

El comando **less** es un paginador que le permite al usuario ver archivos largos página por página. El comando **less** es interactivo y muy intuitivo.

## Sintaxis:

**less** [archivo...]

Para navegar por el archivo mediante el comando **less** use las teclas de las flechas para arriba y para abajo, así como las que a continuación se mencionan:

- **Barra Espaciadora** : permite avanzar una pantalla.
- **Enter** : se navega línea a línea
- **G** : permite ir al final del texto
- **g** : permite ir al principio del texto
- **q** : para salir del programa
- **/** : para buscar palabras en el documento, avanzando en el texto
- **?** : para buscar palabras en el documento, retrocediendo en el texto
- **n** : va a la siguiente aparición de la palabra buscada
- **AvPág** : adelanta una página
- **RePág** : retrocede una página
- **R** : repinta la pantalla
- **v** : carga el fichero indicado en la ejecución del comando **less** en el editor de texto

## Ejemplo:

- **less** archivo.txt

# Comando cat

El comando **cat** concatena archivos y los imprime en la salida estándar. Es ampliamente utilizado para visualizar ficheros.

## Sintaxis:

```
cat [opciones] [archivo(s)]
```

## Opciones:

- **-n** : escribe todos los números de las líneas de salida
- **-A** : opción para mostrar todo

## Ejemplo:

- **cat /var/log/mail/info**

# Comando head

El comando **head** se utiliza para mostrar archivos. Por defecto se muestran las primeras 10 líneas de un fichero.

## Sintaxis:

```
head [opciones] [archivo(s)]
```

## Opciones:

- **-n <num>**: Muestra las primeras num líneas del archivo.
- **-c <num>**: Imprime los primeros num bytes del fichero a mostrar.

## Ejemplo:

- **head /var/log/mail/info**

# Comando tail

El comando **tail** es similar al comando **head**, pero mostrando las últimas líneas de un archivo. Por defecto, se muestran las últimas 10 líneas del fichero.

## Sintaxis:

```
tail [opciones] [archivo(s)]
```

## Opciones:

- **-n <num>**: Muestra las últimas num líneas del archivo.
- **-c <num>**: Imprime los últimos num bytes del fichero a mostrar.

## Ejemplo:

- **tail -n2 /var/log/mail/info**



# Comando find

El comando **find** barre directorios y busca archivos que se correspondan con un conjunto de criterios en esos directorios. Aunque es útil, usarlo requiere cierta práctica.

## Sintaxis:

**find [opciones] [directorios] [criterio] [acción]**

Si no especifica directorio, buscará archivos en el directorio actual.

Si no especifica el criterio, localizará todos los archivos.

## Opciones:

- **-xdev** : no busca archivos en los directorios ubicados en otros sistemas de archivos.
- **-mindepth <n>**: desciende al menos n niveles bajo el directorio especificado antes de comenzar a buscar los archivos.
- **-maxdepth <n>**: busca archivos que se encuentran a lo sumo n niveles bajo el directorio especificado.
- **-follow** : sigue los vínculos simbólicos si apuntan a directorios. Por defecto no los sigue.
- **-daystart**: cuando se usan las pruebas relativas a la fecha y la hora (ver debajo), toma el comienzo del día actual como etiqueta temporal en vez del predeterminado (24 horas antes de la hora actual).

## Criterios:

- **-type <tipo>**: busca archivos de un tipo dado; tipo puede ser: f (archivo regular), d (directorio), l (vínculo simbólico), s (socket), b (archivo en modo de bloques), c (archivo en modo carácter) o p (tubería nombrada).
- **-name <patrón>**: busca los archivos cuyo nombre se corresponde con el patrón dado.
- **-iname <patrón>**: como -name, pero sin tener en cuenta la capitalización.
- **-atime <n>**, **-amin <n>**: busca los archivos a los que se ha accedido por última vez n días atrás (- time) o hace n minutos (-amin). También puede especificar +n o -n, en cuyo caso la búsqueda se hará para los archivos accedidos respectivamente hace al menos o a lo sumo n días o minutos.
- **-anewer <archivo>**: busca los archivos que han sido accedidos más recientemente que el archivo archivo
- **-ctime <n>**, **-cmin <n>**, **-cnewer <archivo>**: igual que -atime, -amin y -anewer, pero se aplica a la última fecha en la cual se modificó el contenido del archivo.
- **-regex <patrón>**: como -name, pero patrón se trata como una expresión regular.
- **-iregex <patrón>**: criterio similar a -regex, pero sin tener en cuenta la capitalización.

## Acciones:

- **-print** : imprime el nombre de cada archivo en la salida estándar; predeterminada.
- **-ls**: imprime en la salida estándar el equivalente de `ls -l` para cada archivo que encuentra.
- **-exec <comando>**: ejecuta el comando comando sobre cada archivo encontrado. comando debe terminar con un ; que deberá desactivar para que el shell no lo interprete; la posición del archivo se representa con {}.
- **-ok <comando>**: igual que -exec pero pide confirmación para cada comando que se use.

## Ejemplos:

- **find /usr/share -type d**
- **find \( -name \*.htm -o -name \*.html \) -a -ctime -30 -exec In {} /var/www/obsolete \**