

DIRECTORIOS PRINCIPALES DE UBUNTU

Si vienes del mundo de Windows lo primero que te sorprenderá es que no está la unidad C:. La distribución de los discos y particiones en Linux es diferente. Ya en la instalación te puedes dar cuenta cómo se nombran los discos en Linux:

- El primer disco duro es /dev/sda
- El segundo disco duro es /dev/sdb

El disco tendrá una o varias particiones, así las del primer disco duro (/dev/sda) serán /dev/sda1, /dev/sda2,... . Éstas tiene el equivalente a lo que en Windows se llama C:, D:, ...etc.

En Linux todo es un archivo. Desde los archivos de datos hasta las particiones de los discos pasando por el ratón y la tarjeta de sonido. Los directorios también son archivos. Existe un directorio raíz (/) del que cuelga todo.

- **/:** Raíz.
- **/usr:** Aquí se encuentra la gran mayoría de los archivos existentes en un sistema Linux, tales como documentación, casi la totalidad de los comandos de usuario, juegos, librerías, etc.
- **/bin:** Aquí están los comandos que pueden usar todos los usuarios (incluido el administrador o root).
- **/sbin:** Aquí están los comandos que sólo puede usar el administrador o root.
- **/dev:** Ahí están todos los dispositivos de nuestra máquina.
- **/home:** Lugar donde se almacenan las cuentas de usuarios.
- **/lib:** Enlaces a las librerías que se necesitan para el sistema.
- **/var:** Contiene información variable, como por ejemplo los logs del sistema (/var/log), correo local, etc.
- **/tmp:** Directorio temporal.
- **/etc:** Configuración global de los programas.
- **/root:** Cuenta del administrador.
- **/boot:** Aquí está todo lo necesario para arrancar el sistema.
- **/media:** Punto de montaje para sistemas de archivos montados localmente.
- **/mnt:** Antiguo punto de montaje para sistemas de archivos montados localmente. Hoy en día está en desuso y se conserva por razones históricas.
- **/proc:** Sistema de archivos virtual de información de procesos y del kernel.

Algunas referencias de Archivos conocidos:

- **/etc/apt/sources.list:** Este archivo configura la lista de repositorios
- **/etc/X11/xorg.conf:** Este archivo define y configura el entorno gráfico a cargar.
- **/etc/fstab:** Este archivo configura el acceso a los diferentes sistemas de archivos que hay en nuestro sistema.
- **/etc/passwd:** Este archivo controla el uso de usuarios, en contraseñas, con permisos y grupos que pertenecen a cada usuario, archivo muy importante si uno quiere tener un usuario root más que el ya conocido root.

LOS METACARACTERES PARA ARCHIVOS

Linux tenemos un conjunto de caracteres especiales muy útiles para realizar búsquedas u otras operaciones sobre ficheros y directorios. En Linux, dicho conjunto de caracteres se denomina **metacaracteres** y son los siguientes:

- **?** Sustituye a cualquier carácter, uno solo.
- Sustituye a cualquier grupo de caracteres (En los ficheros ocultos, no puede sustituir al punto inicial).
- **[rango]** Cualquier carácter del rango.

- [!rango] Cualquier Carácter distintos del rango.
- ~ Contiene la ruta absoluta de la carpeta de trabajo del usuario activo en el sistema.

Los rangos se pueden establecer de las siguientes formas:

pe[abc] pea peb pec

mio[a-d] caracteres desde la a la d según el código ASCII.

xx[!123] Todos los nombres que no terminen e 123

ca[x-z]01?*

CARACTERES ESPECIALES:

Algunos caracteres tienen una interpretación especial como pueden ser el &, \$, *, \, >, <, >>, << etc. Si se quiere usar estos caracteres como caracteres normales, se puede emplear:

- Usar la Barra invertida \, inhibe la interpretación del carácter siguiente.
- Comillas simples: Impide que el Shell interprete todo lo incluido entre las comillas.
- Comillas dobles: Evita también que el Shell interprete todo lo incluido entre las comillas excepto el símbolo \$, el \, y las comillas simples y dobles.
- Acento grave (francés): Ejecuta la orden y sitúa su salida estándar en el ámbito donde se ejecutó. Es lo que Linux denomina como sustitución de orden.

EJECUCIÓN DE ORDENES:

- Una orden se puede ejecutar en background (ejecución en 2º plano) añadiendo un símbolo & al final de esta, de esta manera, el indicador del Shell queda a disposición del usuario y no hay que esperar a que le proceso lanzado termine. Al lanzar una orden en background, se muestra por pantalla el nº de PID de esta.
- Pueden usarse varias órdenes en una sola línea separadas por el carácter; se ejecutan de izquierda a derecha.
ord1; ord2; ord3
- Si además se agrupan entre paréntesis, tiene un significado especial en ciertos casos como redireccionamientos o para que todas ejecuten en segundo plano

(ord1; ord2; ord3) > mifchero.dat

ord1; ord2; ord3 > mifchero.dat *Solo implica a ord3.*

(ord1; ord2; ord3) &

ord1; ord2; ord3 & *Solo implica a ord3.*

- Otras posibilidades que se suelen usar sobre todo en programación son:

ord1 && ord2 *Solo se Ejecuta ord2 si la ejecución de ord1 ha tenido éxito.*
ord1 || ord2 *Solo se Ejecuta ord2 si la ejecución de ord1 no ha tenido éxito.*

VARIABLES DEL SHELL:

El Shell dispone de una serie de variables denominadas, **variables de entorno** que sirven para mantener el control del sistema, son variables globales a todos los procesos. Las variables de entorno siempre se escriben en mayúsculas.

Para visualizar cualquier variable se pone el símbolo \$ delante del nombre de la misma:

\$ echo uso un terminal \$TERM y mi nombre es \$LOGNAME

Ejemplos de algunas variables de entorno son:

- ✦ TERM Tipo de terminal que estamos usando.
- ✦ LOGNAME Nombre de usuario.
- ✦ HOME Directorio del usuario.
- ✦ PATH Caminos donde el shell buscara las órdenes
- ✦ PS1 y PS2 Indicadores del prompt primario y secundario.
- ✦ SHELL tipo de Shell que estamos usando.

El usuario puede crear también sus propias variables, solo con asignarles un valor, una variable no asignada tiene valor nulo. Las variables del usuario son locales al Shell donde se encuentra.

*\$ nombre=jose
\$ echo \$nombre*

Si se quiere que una variable se vea en todo el entorno Linux hay que exportarla con el comando export:

\$ export nombre

ALGUNOS COMANDOS INICIALES:

clear Borra la pantalla.

who Muestra por pantalla una línea por cada usuario que en ese momento está conectado al sistema. Muestra, de izquierda a derecha, el nombre de presentación del usuario, el número de terminal y la fecha y hora de presentación.

whoami - Indica el usuario que está trabajando en la terminal actual.

Sintaxis:

whoami

finger - Presenta una información completa de los usuarios conectados a la red.

Sintaxis:

finger [-bfilpqsw][login1 login2]

uname - Proporciona el nombre del sistema en el que se está trabajando.

Sintaxis:

uname [-opciones]

Como opciones principales tenemos:

-a indica, además, la versión, fecha y tipo de procesador.

-m indica, además, el tipo de procesador.

-r indica, además, la versión.

-v indica, además, la fecha.

logname - Indica el nombre del usuario conectado al sistema (el que ha hecho login).

Sintaxis:

logname

info - Proporciona ayuda resumida acerca de un comando en cuestión.

cal - Muestra el calendario del mes o año actual formato anglosajón

ncal - Muestra el calendario del mes o año actual formato español

Sintaxis:

cal [año]

cal [mes] [año]

man Orden Manual de Linux en línea, se suele especificar con el nombre de alguna orden. Se detiene cada pantalla, salir con q. Existe info, help, learn y otros.

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

passwd permite cambiar la clave de acceso. El administrador puede poner restricciones a la Clave y cambiarla, pero nunca verla.

date Visualizar fecha y hora (usuarios), el administrador puede cambiarla.

date mmddhhmm[yy] (Sólo superusuario) o date [+formato]

La especificación de un formato consiste generalmente en una cadena entre comillas sencillas. Los caracteres entrecomillados se presentan literalmente, excepto los campos particulares donde los datos vienen representados entre signos de porcentaje (%) seguidos por una letra de código.

%n	Insertaría una nueva línea.	%%	Escribe un signo %.
%t	Insertaría un carácter de tabulación.	%r	Notación am o pm.
%m	Mes del año, 01 al 12.	%j	Día del año 1-366.
%d	Día del mes, 01 al 31.	%T	Hora HH:MM:SS.
%a	Día de la semana.	%S	Segundos 00-59.

%h	Nombre del mes	%M	Minutos 00-59.
%y	Ultimos dos dígitos del año 00-99.	%H	Hora 00-23.
%D	Fecha formato mm/dd/yy.		

Ejemplo: date +'FECHA:%d/%m/%y%nHORA:%T'.

Generaría esta salida FECHA: 10/01/95
HORA: 12:45:00

type orden Informa donde se encuentra el fichero de esa orden, o si es una rutina del SHELL.

echo argumentos

Muestra los argumentos en el dispositivo de salida estándar (stdout) pantalla. Se usa para ver valores de Variables, parámetros del Shell y sobre todo en Guiones (programas o Shell-Scripts).

La orden echo entiende las siguientes secuencias de escape (caracteres de control), las cuales pueden estar embebidas en los argumentos de echo:

\c	Imprime la línea sin NEWLINE
\n	NEWLINE
\t	Tab
\\	Backslash
\On	El código ASCII octal de cualquier carácter.

tty Información de nuestro terminal. Nos va a dar el nombre del fichero especial asociado.

su o su usuario o su - usuario

Switch User. Cambiar de usuario. Si un usuario tiene más de una cuenta, no es necesario desconectar y volver a conectar para cambiar. Pide la clave correspondiente. Si se utiliza sin nombre de usuario, el Shell supone que se desea cambiar a administrador (Superusuario) y espera el password del administrador. Siempre se crea un subshell (para volver al anterior se úsala orden exit).

Si se pone un – delante del nombre del usuario, se toma los parámetros de configuración de este, en caso contrario se queda uno con los del anterior.

id

Muestra información sobre el UID, su GID principal y otros grupos a los que pudiese pertenecer. Estos números no cambian nunca, y son la identificación interna del usuario.

Whereis

Nos indica la ubicación de los binarios de un comando.

PRINCIPALES ÓRDENES PARA MANTENIMIENTO DIRECTORIOS Y FICHEROS:

Orden mkdir (Crear directorio)

- mkdir directorio1 directorio directorio3....
- mkdir -p ruta_de_directorios

El usuario debe tener permiso de escritura para el directorio. Si se usa la opción -p, se p crear los

directorios padres automáticamente si no existen.

```
$ mkdir mio1 mio2 /usr/vadai01/ejemplos
```

```
$ mkdir -p mio/ejemplos/ejem1
```

 Los directorios que no existan se crean.

```
$ mkdir -p Musica/{Pop,Metal/Heavy,BSO,Rock/{Gotico,Progresivo},Clasica/Barroca}
```

Para crear un directorio con permisos específicos, debemos la opción -m (-mode). La sintaxis para asignar permisos es la misma que con el comando chmod

- mkdir -m 775 nuevodirectorio

Orden rmdir (Borrar Directorio)

- rmdir directorio1 directorio2 directorio3
- rmdir -p ruta_de_directorios

El usuario debe tener permiso de escritura para el directorio. Si se usa la opción -p, se borran los directorios padres si existen y están vacíos.

```
$ rmdir mio1 mio2 /usr/vadai01/ejem.c
```

```
$ rmdir -p otro/datos/dato1
```

 Otro y Datos se borran si están vacíos.

Orden rm: Borrar ficheros (y directorios).

- rm fichero1 fichero2 fichero 3
- rm -r fichero_o_directorio ← Se pueden usar comodines.

El usuario debe tener permiso de escritura para el directorio que contiene el fichero, en cambio no hace falta tener permiso sobre el fichero que se quiere borrar. Si se intenta borrar un fichero que no tiene permiso de escritura, pedirá confirmación.

Si se usa la opción -r, se eliminan también todos los subdirectorios que existan. Si se trata de un directorio, se elimina este con todo su contenido.

Existe la opción -i para pedir confirmación por cada fichero a borrar, es una opción importante para obtener seguridad con esta operación.

```
$ rm datos 1 otro.c /usr/vdai01/ejer
```

```
$ rm cal*
```

```
$ rm -r usr/dai01/aj[1-5]*
```

Orden cp: Copia ficheros.

- cp fichero_origen fichero_destino
- cp fichero1 fichero2 fichero3.... directorio_destino
- cp fichero/s directorio_destino

Se pueden usar comodines, si el fichero ya existe es reemplazado.

Se puede usar la opción -r copia la estructura del árbol incluida en la trayectoria origen, y también se puede usar la opción -i que pide confirmación antes de sobrescribir un archivo.

Orden mv (Mover –renombrar- ficheros. También sirve para directorios):

- `mv fichero_origen fichero_destino`
- `mv fichero1 fichero2 fichero3.... directorio_destino`
- `mv fichero/s directorio_destino`

Esta orden hace lo siguiente:

1. Sirve para trasladar el fichero o los ficheros indicados al destino que será un directorio.
2. Sirve para renombrar un fichero si en destino indicamos el nuevo nombre del fichero.
3. Sirve para trasladar y renombrar un fichero si en la trayectoria destino ponemos la trayectoria de un fichero que no existe en otro directorio.

Se pueden usar comodines.

Se puede usar `-i` para pedir confirmación y `-f` para forzar el movimiento o cambio sin pedir confirmación.

```
$ mv nombre1 nombre2
$mv usr/vdai01/miejer.x usr/vadai01/ejer    ejer es un nombre de directorio
```

Si se usan como origen y destino dos nombres de directorios, entonces la orden sirve para mover y/ o renombrar directorios:

- `mv trayectoria_origen trayectoria_destino`

En este caso dependiendo del valor de trayectoria origen y destino (ambos deben ser directorios), podemos:

1. Renombrar el directorio.
2. Trasladar el directorio.
3. Renombrar y trasladar el directorio.

Orden ln (Enlaces)

- `ln fichero_origen nombre_enlace`

Sirve para que un fichero aparezca en el sistema de archivos bajo dos o más nombres diferentes, pero con una única copia. Con la orden `ln`, al contrario que con `cp`, no se hace una copia del archivo origen, solamente se crea otro nombre de archivo que hace referencia al archivo físico. Esto permite que una única copia de un archivo aparezca en varios directorios con varios nombres. De esta forma se puede compartir información de forma cómoda.

Si en algún momento se elimina alguno de los archivos que hacen referencia a la misma copia física, solo eliminaremos el nombre, pero no la copia real. El fichero existirá mientras haya algún enlace.

Este tipo de enlaces se denomina hard links o enlaces fuertes, y no son aplicables a ficheros que se encuentren en diferentes sistemas de archivos (por ejemplo, particiones u otras unidades) ni tampoco son aplicables a directorios.

Para diferentes sistemas de archivos o para directorios hay que utilizar los que se denominan enlaces suaves o Soft links, mediante la opción `-s` (Si se borra el origen los enlaces permanecen sin apuntar a nada concreto.)

- `ln -s fichero_origen nombre_enlace` Enlace a un fichero en el mismo u otro sistema de archivos.
- `ln -s directorio nombre_enlace` Enlace con un directorio.

Cuando hacemos un listado con la orden `ls`, ficheros y directorios que son los enlaces soft aparecen con una `l` en la parte izquierda de los permisos, con todos activados, (`lrwxrwxrwx`), en realidad los permisos son los del fichero o directorio original. En la parte derecha del listado se observa cual es el fichero o directorio origen.

Orden pwd (Print Working directory):

- `pwd`

Visualiza el directorio de trabajo con trayectoria absoluta.

/usr/vdai05/miscartas

Orden cd (Cambiar de directorio)

- `cd ..` Lleva un blanco antes de los dos puntos.
- `cd`
- `cd /`
- `cd directorio`

Sirve para cambiar de directorio:

1. Si se usa `cd ..` se sube un nivel en el árbol (directorio padre).
2. Si se usa `cd` se va al directorio del usuario, también llamado directorio Home ya que es el definido por la variable de sistema HOME.
3. `cd /` nos sitúa en el raíz (root).
4. `cd directorio`, nos lleva al directorio especificado, podemos usar trayectorias absolutas y relativas:

\$cd /dev/tty/typ

\$ cd ejemplos/pascal

Orden cat (Visualizar, crear, añadir archivos ASCII)

- `cat fichero`
- `cat`
- `cat > fichero` Crea el fichero
- `cat >> fichero` Añade al final del fichero
- `cat fichero1 fichero2 fichero3..... > ficherosalida` Concatena en el fichero
- `cat fichero1 fichero2 fichero3..... >> ficherosalida` C. y añade al final del fichero

Opciones:

- s suprime el mensaje de error si el fichero no existe.
- v Visualiza los caracteres no imprimibles que pueda contener el fichero.

Si usamos `cat` sin argumentos, entonces lee caracteres desde la entrada standard (stdin) teclado hasta que pulsemos Ctrl.-D y después visualizará lo que terminemos de escribir.

Si usamos \$ **cat** > **fichero**, estamos creando el fichero desde el teclado. Usando >> en vez de > añadimos el al final del fichero.

Si usamos varios ficheros y los dirigimos a uno de salida, se concatenan (de hay la palabra cat conCATenar).

Orden ls (Listar archivos):

- ls [-opcion/es] [directorio]
- ls [-opcion/es] [fichero/s]

Visualiza los ficheros de un directorio en orden alfabético. La orden ls sin argumentos lista el contenido del directorio de trabajo actual. Se pueden usar comodines si se usa con ficheros.

Es una orden con un amplio espectro de opciones, algunas de ellas son:

- -a Lista todos los ficheros, también archivos ocultos
- -F Pone Un slash “/” antes de cada fichero que sea un directorio y un asterisco “*” detrás de cada fichero que sea ejecutable (permiso de ejecución).
- -R Lista recursivamente el contenido de los subdirectorios.
- -l Listado largo, muestra información detallada.
- -d Siempre con l, solo muestra información del directorio.
- -r Invierte el orden de salida (orden alfabético inverso).
- -s Muestra el tamaño de cada archivo en bloques.
- -x Listado por columnas ordenado horizontalmente.
- -C Listado por columnas ordenado verticalmente.
- -u Listado ordenado por fecha de último acceso al fichero.
- -i N° de i-node de los ficheros.

La opción más detallada nos la da -la que produce una salida similar a esta:

```
-rw-rw-r-- 1 vdai01 group 5782 Mar17 16:30 ejem.c
```

De izquierda a derecha:

- El primer carácter indica el tipo de fichero: - ordinario.
 - d directorio l
 - enlace soft.
 - b especial (dispositivo bloque)
 - c especial (dispositivo carácter)

- Permisos de usuario, grupo y otros (nueve caracteres): -rw-rw-r-- un símbolo – indica la ausencia de permisos.
- N° de enlaces, siempre tiene 1 como mínimo ya que un fichero siempre esta enlazado a su directorio, los directorios tienen como mínimo 2 (correspondientes a . y ..)
- Nombre del propietario: vdai01
- Nombre del grupo: group
- Tamaño en bytes
- Fecha última modificación.
- Nombre del fichero (directorio o enlace).