Estructura de directorios en Linux

Cuando un sistema operativo GNU/Linux es instalado en un medio de almacenamiento, se crean un conjunto de directorios donde se despliegan los archivos y directorios del propio sistema, en una estructura en forma de árbol. Esto es, un directorio del que cuelgan un conjunto de subdirectorios y, dentro de estos últimos, se alojan otros subdirectorios, y así sucesivamente.

Para hacer referencia a la localización de un directorio dentro de la estructura arbórea, es emplea el término **ruta**, **camino** o **path:**

- **Ruta absoluta** → Ruta completa indicada desde el directorio raíz (/)
- **Ruta relativa** → Ruta indicadada desde el directorio de tabajo actual.
  Se suele hacer uso de :

  "**..**" , "**.**" , "**~**"

Estos caracteres hacen referencia a los **directorios especiales**:

/	**Directorio inicial** de la estructura de directorios. De él cuelga el resto de directorios y archivos.
.	**Directorio actual**, referencia al directorio en que estamos
..	**Directorio padre,** del directorio actual
~	**Directorio personal,** de cada usuario

En Linux se utilizan **metacaracteres** con un significado especial para búsquedas y operaciones sobre archivos

*	→ Cualquier serie de caracteres (0 o más)
?	→ Cualquier carácter (uno)
[…]	→ Cualquiera de los caracteres encerrados. Podemos incluir un conjunto de caracteres o un rango de caracteres, pero el corchete sustituye sólo uno.

Ejemplos:

[bf]	*b , f*
[b-f]	*b , c , d , e , f*
[!bf]	*ni b ni f*
a[1-5]	*a1, a2 , a3 , a4 , a5*
a1[xy]3	*a1x3 , a1y3*

La salida de un comando puede ser **redireccionada** a otro destino según nuestros intereses:

>	Direcciona la salida estándar
<	Direcciona la entrada estándar
2>	Direcciona el error estándar
>>	Direcciona la salida estándar(al final)

Los pipes o tuberías permiten **encadenar** la salida de un comando para que sea la entrada del siguiente:

- El carácter usado es |

Ejemplos:

cat archivo1 archivo2 > archivo3	*Crea el archivo3 con la concatenación de archivo1 y archivo2*
ls -l < lista_direct	*Lista el contenido de los directorios que están el archivo lista_direct*
ls /home/user1/carpeta | wc -c	*Muestra el número de archivos que están en el directorio "carpeta"*

Gestión de archivos por línea de comandos

En entorno domésticos o de oficina se emplea la interfaz gráfica para el manejo y gestión de archivos. No obstante, los administradores de sistemas suelen hacer uso de la interfaz por línea de comandos, ya que su versatilidad y potencia de uso la convierte en una herramienta ideal:

Los comandos de linux siguen una sintaxis:

comando [*opciones*] [*argumentos*]

ls

Muestra información sobre ficheros y directorios. Sin parámetros muestra el directorio actual.

ls [opciones] [argumentos]

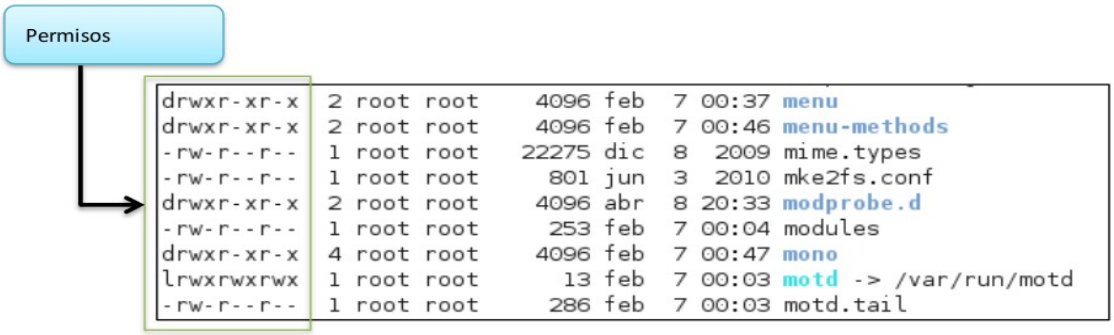
Opciones

-d	Información del directorio, no del contenido
-a	Muestra los archivos y directorios que comienzan por un punto (ocultos)
-l	Información en formato largo
-i	Muestra el número de i-nodo del fichero
-t	Información ordenada por día y hora de creación (de mayor a menor)
-r	Información ordenada en orden inverso
-R	Muestra los directorios por debajo del actual de forma recursiva

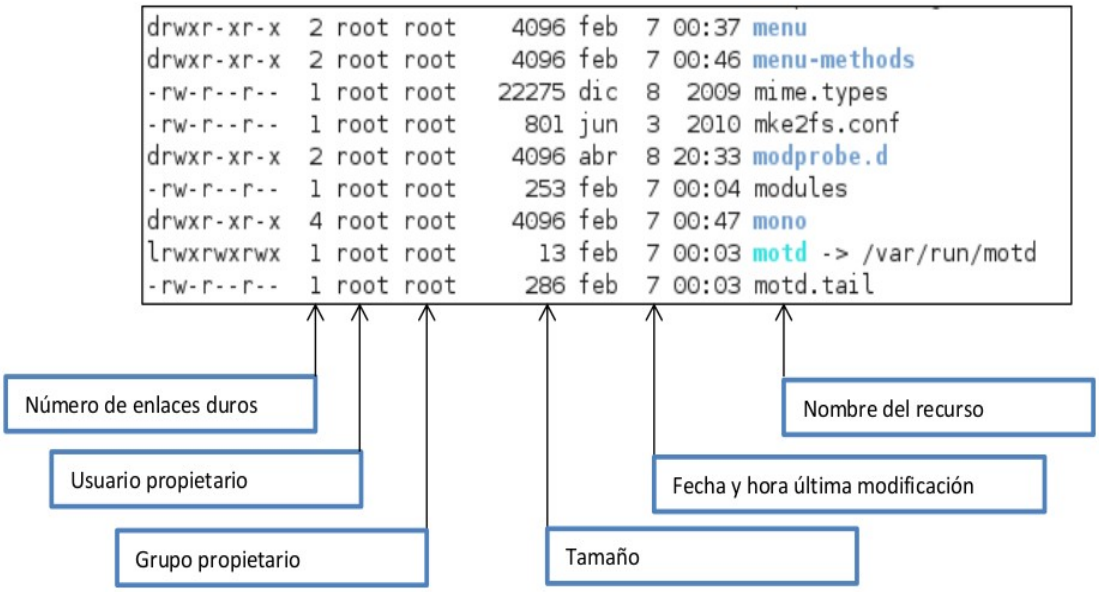
Patrones

Podemos añadir el carácter * si queremos buscar patrones (por ejemplo ls d* para buscar ficheros que empiecen por d en el directorio actual)

Información ls



- ✔ **1** → Directorio(d), enlace(l), dp. bloque(b), disp. caracteres(c), socket(s), tubería(p), archivo(-)
- ✔ **2,3,4** → Permisos para el usuario dueño del recurso.
- ✔ **5,6,7** → Permisos para el grupo dueño del recurso.
- ✔ **8,9,10** → Permisos para el resto de usuarios



- ✔ El número de enlaces duros que posee. En el caso de un directorio, indica la cantidad de subdirectorios que contiene contando a los directorios especiales ``." y ``.."
- ✔ El identificador del dueño.
- ✔ El identificador del grupo.
- ✔ El tamaño en bytes si es un fichero y si es un directorio el tamaño en bloques que ocupan los ficheros contenidos en él.
- ✔ La fecha y hora de la última modificación. Si la fecha es de hace 6 meses, se coloca el año.
- ✔ El nombre del recurso.

pwd

[print working directory] → Muestra la ruta absoluta del directorio actual

mkdir

[make directory] → Crear directorios

mkdir [opciones][directorio/s]

Opciones

-p Si no existen los directorios intermedios, se crean también

cd

[*change directory*] → Cambiar de directorio

cd [directorio]

rmdir

[*remove directory*] → Borrar directorios si están vacíos

rmdir [opciones] directorio/s

Opciones

-p Borra todos los directorios vacíos que encuentra en una ruta

rm

[*remove*] → Borra ficheros y directorios

rm [opciones] [argumentos]

Opciones

-i	Pregunta antes de borrar cada fichero o directorio
-r | -R	Borra directorios y ficheros contenidos en los directorios, recusivamente
-f	Borra el directorio actual y subdirectorios, sin preguntar

cp

[*copy*] → Copia uno o varios ficheros en otro fichero o en un directorio

cp [opciones] fichero/s destino

Opciones:

-f	Fuerza la copia
-i	pregunta antes de sobreescribir
-R | -r	Copia los directorios por debajo del actual, recursivamente

mv

(move) Mueve uno o más ficheros a otros ficheros o directorios (no hace falta –R como en el comando cp). Puede ser usado para renombrar ficheros.

mv [opciones] fichero/s destino

Opciones

-u	Mueve sólo si el destino no existe o es anterior
-i	Pregunta antes de sobreescribir
-f	Fuerza la sobreescritura

file

Muestra el tipo de fichero

file nombre

touch

Actualiza la fecha de acceso y modificación de ficheros. Si no existe, lo crea vacío.

touch [opciones] fichero

Opciones

-c	No crea ningún fichero

cat

Muestra el contenido de los ficheros que se le pasan como argumento

cat [opciones] fichero/s

head/tail

Muestra las 10 primeras (últimas) líneas de los ficheros que se le indican

head [opciones] fichero/s

Opciones

-n	Muestra las n primeras (últimas) líneas, en vez de 10

more

Muestra el contenido de los ficheros de forma paginada. Se usa la barra espaciadora para pasar la página.

more fichero/s

less

Igual que more, pero pudiendo utilizar las flechas de cursor

less fichero/s

sort

Muestra en orden ascendente el contenido de los ficheros que se le pasan como argumentos

sort [opciones] [fichero/s]

Opciones

-r	Ordena en sentido inverso
-t	Añade un separador (tabulador por defecto)
-k	Según el separador, se ordena por número de campo (-k1, -k2, etc)

wc

Orden que contabiliza el número de líneas, palabras y caracteres de un archivo.

Opciones

-c	Cuenta sólo caracteres.
-w	Cuenta sólo palabras.
-l	cuenta sólo líneas.

cut

Muestra sólo ciertas líneas verticales de los ficheros argumentos

cut [opciones] [fichero/s]

Opciones:

-c	Muestra sólo los caracteres especificados
-d	Usa el carácter especificado como delimitador
-f	Muestra los campos indicados en lista
-s	No muestra líneas sin delimitador

grep

Muestra las líneas de un fichero que contienen un cierto patrón

grep [opciones] patrón [fichero/s]

Opciones:

-r	Busca recursivamente en los ficheros de un directorio
-i	No distingue mayúsculas y minúsculas
-v	Muestra las líneas que no contienen el patrón

whereis

Localiza ficheros ejecutables o binarios, fuentes y páginas del manual de los programas o comandos argumentos

whereis argumento/s

which

Muestra la ruta absoluta del archivo del comando argumento

which argumento/s

locate

Busca archivos en el sistema de archivos por nombre.

locate fichero

find

Busca ficheros en un árbol de directorios. Encuentra ficheros según un conjunto de criterios

find [opciones] [directorio] [criterios] [accion]

Opciones:

-type	tipo	Busca por tipo de archivo
-name	nombre	Busca por nombre

ENLACES en LINUX

En linux vamos a encontrar un comando que nos creará fácilmente enlaces denominado:

ln [--help]

Para ver su i-nodo: **ls –li**

Para encontrar los enlaces de un fichero: **find / -inum xxxxx**

Vamos a utilizar **2 tipos** de enlaces:

- **Simbólicos**
- **Duros**

Simbólico

Es un tipo de enlace que no es real, sino que supone un atajo a un archivo o carpeta, sería parecido a los accesos directos. Cada enlace simbólico dispone de su **propio número de inodo** y es diferente al del archivo original

Si borramos la carpeta/archivo a la que apunta, el enlace persistiría pero **no sería útil.**

Ejemplo

ln -s [opciones] origen [dest]

Duro

Un enlace duro es un archivo que apunta al mismo contenido almacenado en disco que el archivo original. Los archivos originales y los enlaces duros dispondrán del **mismo inodo** y consecuentemente ambos estarán apuntando hacia el mismo contenido almacenado en el disco duro

Si borramos la carpeta/archivo a la que apunta, el **enlace persistiría.**

Ejemplo

ln [opciones] origen [dest]