

Introducción al sistema operativo GNU/Linux

1. ¿Qué es Linux?	2
2. Interfaz en modo texto. El shell de Linux	2
3. Órdenes básicas de Linux	3
4. Directorios y carpetas en Linux	3
Estructura jerárquica de archivos y directorios	4
5. El sistema de archivos en Linux	4
Características del sistema de archivos en Linux	4
Tipos de sistemas de archivos en Linux	5
Directorios más importantes en Linux	6
Caminos o trayectorias en Linux	7
Metacaracteres en Linux	7
6. Ayuda en Linux	8
7. Operaciones sobre directorios y carpetas	8
Cómo crear directorios o carpetas	8
Ver el contenido de los directorios o carpetas y moverse por ellos	9
Visualizar la estructura de directorios o carpetas	9
Eliminar directorios o carpetas	9
Mover y copiar directorios o carpetas	10
Cambiar el nombre a un directorio o carpeta	11
8. Atributos de los directivos o carpetas	11

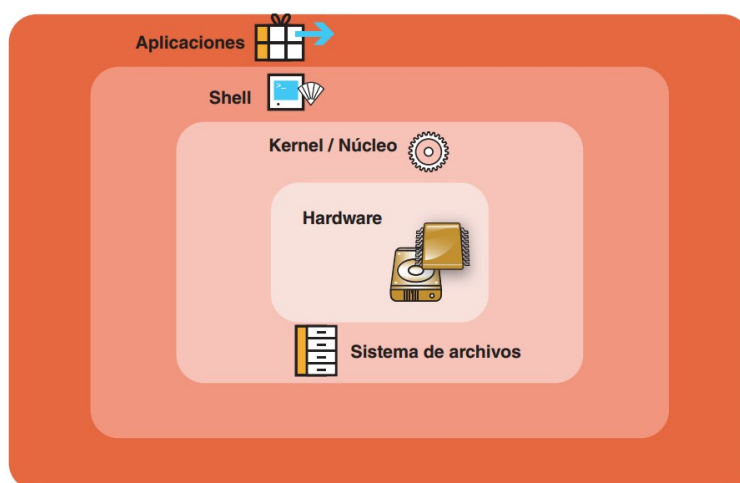
1. ¿Qué es Linux?

Linux es un **sistema operativo de gran difusión** en la actualidad, debido principalmente a que es de **libre distribución**. Permite trabajar al usuario en **modo monopuesto** o en **modo de red**, al contrario que otros sistemas operativos como Windows XP o Vista, permitiendo iniciar sesión localmente en la máquina y de forma remota.

Es una versión de UNIX desarrollada para funcionar en plataformas de ordenador personal. También recibe el nombre de **GNU/Linux** o sistema operativo libre con licencia pública general de GNU. Su nombre proviene del **núcleo de un sistema operativo** desarrollado en 1991 por **Linus B. Torvalds**. Aunque se usa como sistema operativo monopuesto, su **funcionalidad principal queda patente en sistemas servidores y supercomputadores**.

Se puede instalar en **gran variedad de plataformas hardware**, incluidos equipos con arquitecturas de 32 bits (x86), 64 bits (x64) y plataformas Macintosh y PowerPC; también en telefonía móvil (Android) o consolas de juegos.

De forma general, la estructura de un sistema operativo UNIX/Linux es la que se muestra a continuación.



Este sistema operativo consta de **dos componentes principales**:

- **Shell**. Es el equivalente al intérprete de comandos de Windows (command.com en MS-DOS) y sirve para introducir órdenes, ejecutar programas, etc.
- **Núcleo**. Como en cualquier sistema operativo, es la parte del sistema que interactúa con el hardware.

2. Interfaz en modo texto. El shell de Linux

Una forma de interactuar con el sistema es a través del denominado **Terminal**, similar al intérprete de comandos de Windows. Al Terminal de Linux se le llama **shell**. Linux, en sus diferentes distribuciones, incorpora alguno de los siguientes tipos de shell.

- ✓ **Shell Bourne (sh)**. Creado por S. Bourne, es el más utilizado en la actualidad. **Su símbolo de sistema es \$**. Es el shell estándar de AT&T y el que se monta en casi todos los sistemas UNIX/Linux.
- ✓ **C-Shell (csh)**. Procedente del sistema BSD, proporciona funciones tales como control de trabajos, historial de órdenes, etc. Ofrece importantes características para los programadores que trabajan en lenguaje C. Su símbolo de sistema es %.

- ✓ **Shell job (jsh).** Incorpora algunas características de control al shell estándar del sistema.
- ✓ **Shell Korn (ksh).** Escrito por David Korn, amplía el shell del sistema añadiendo historial de órdenes, edición de la línea de órdenes y características ampliadas de programación.
- ✓ **Bourne Again shell (Bash).** Fue creado para usarlo en el proyecto GNU. BASH, por lo tanto, es un shell o intérprete de comandos GNU que incorpora la versión Ubuntu. Es compatible con el shell sh. Además, incorpora algunas características útiles de ksh y csh, y otras propias como la edición de línea de comandos, tamaño ilimitado del historial de comandos, control de trabajos y procesos, funciones y alias, cálculos aritméticos con números enteros, etc. Su símbolo de sistema es **nombre_usuario@nombre_equipo:~\$**.

En general, todas las órdenes de UNIX/Linux son programas que están almacenados en el sistema de archivos. Su sintaxis es similar a la de Windows y es: **\$ comando [-modificadores] [argumentos]**.

3. Órdenes básicas de Linux

Hasta no hace demasiado tiempo, toda la gestión del sistema operativo Linux se basaba en la **ejecución de órdenes en entorno comando** o a través de la introducción de las mismas en los terminales de trabajo. En general, el **formato de las órdenes** en Linux en entorno comando es la que se recoge en la tabla.

ORDEN	Indica al intérprete de comandos o shell la acción a realizar .
OPCIONES	Las opciones o modificadores se introducen seguidas del carácter - .
ARGUMENTOS	Entrada del comando. El argumento puede ser un archivo o un directorio u otra información.

De forma general, lo primero que se introduce es la **orden**, seguida de los argumentos o parámetros de la misma. Cuando tenemos que introducir parámetros o un argumento (caracteres como tal) se pone un espacio en blanco entre la orden y el parámetro o modificador, y entre estos y los argumentos.

4. Directorios y carpetas en Linux

Los directorios en Linux tienen esencialmente las mismas características que los directorios en Windows. Se trata de un conjunto de archivos y/u otros directorios agrupados en función de su contenido, su propósito o cualquier otro criterio de agrupación. Un **directorio** no es ni más ni menos que **un tipo de archivo**, que **almacena información acerca de los archivos y directorios que contiene**.

Al igual que los archivos, los directorios o carpetas en Linux tienen **características que los clasifican, catalogan y ubican dentro del sistema de archivos**.

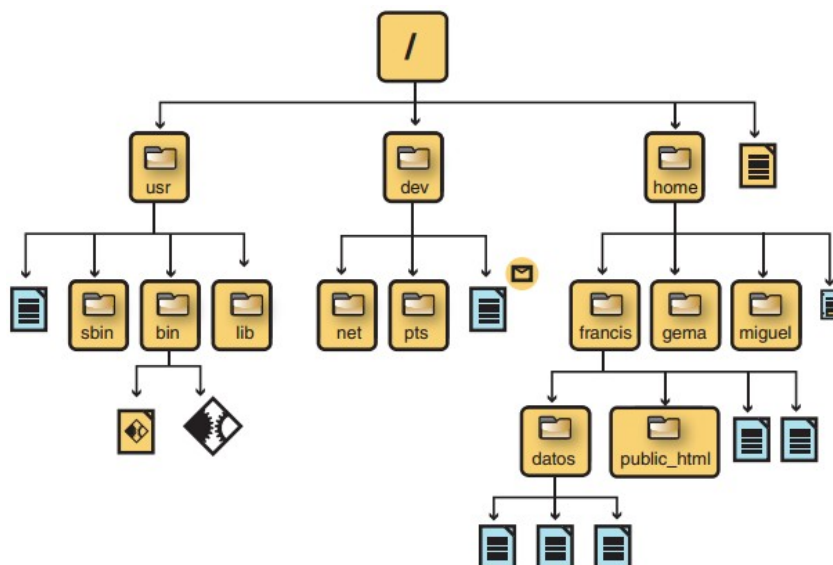
Respecto de los **nombres de las carpetas**, indicar que se puede utilizar cualquier carácter (al igual que para los ficheros), a **excepción del carácter /**, ya que tiene un significado especial. Es conveniente no utilizar caracteres especiales, acentos, etc. Aunque se permite, no es conveniente incluir caracteres como los siguientes en los nombres de archivo/directorio: ! # & () ' " ; | > < @ \$ { } * ? \ Tabulador Espacio + -.

Hay que destacar que en Linux **los nombres de archivo/directorio son diferentes en mayúsculas y en minúsculas**. Se recomienda utilizar siempre minúsculas.

Las **operaciones a realizar con carpetas** o directorios en Linux son las mismas que en Windows: **creación** (*create*), **consulta** (*opendir*), **actualización** (*updatedir*), **borrado** (*delete*) y **renombrado** (*rename*).

Estructura jerárquica de archivos y directorios

La estructura del sistema operativo Linux en cuanto al almacenamiento de la información es más compleja que la utilizada por Windows, siendo **jerárquica en árbol**. En Linux **no hay límite en la cantidad de archivos y directorios a crear dentro de un directorio**. La imagen muestra un ejemplo de una estructura típica de directorios en el sistema Linux.



En el **Terminal** o en modo comando, **Linux muestra en el símbolo del sistema la identificación del usuario, el equipo en el que está identificado y detrás la ubicación o ruta en la que nos encontramos**. Con el comando **pwd** se mostrará por pantalla la **ruta absoluta** en la que está situado.

```
paco@ubuntupaco:~$ pwd
/home/carpetas_de_trabajo
```

5. El sistema de archivos en Linux

Un sistema de archivos es el **modo en que el sistema operativo organiza los archivos en el dispositivo de almacenamiento**, gestionándolo de manera que los datos estén de forma estructurada y libre de errores. La estructura de archivos de Linux es una estructura jerárquica en forma de árbol, donde el **directorio principal (directorio raíz)** es el **directorio /**, del que **cuelga toda la estructura del sistema**.

Linux **admite nombres de ficheros largos** y se puede usar cualquier carácter, excepto los indicados en el punto anterior, teniendo especial importancia las mayúsculas y minúsculas.

Características del sistema de archivos en Linux

El sistema de archivos o *file system* de Linux permite al usuario crear, borrar y acceder a los archivos sin necesidad de saber el lugar exacto en el que se encuentran. **No existen unidades físicas**, sino **ficheros que hacen referencia a ellas**, integrados en la estructura de archivos como cualquier otro. El sistema de archivos Linux consta de varias partes importantes:

- ✓ **Superbloque.**
- ✓ **Tabla de inodos.**
- ✓ **Bloques de datos.**

En Linux **cada bloque es de 512 KB** (como en muchos otros sistemas operativos) **o de múltiplos de 512**. Al igual que el cluster era la parte más importante en el sistema de archivos FAT, en **ext4 es el bloque**.

- El **bloque de carga** o **bloque cero** de cada sistema está reservado para almacenar un programa que utiliza el sistema para gestionar el resto de las partes del sistema de archivos.
- El **superbloque** o **bloque uno** contiene la información sobre el sistema de archivos.
- La **tabla de i-nodos** es el equivalente a las entradas de la FAT. por cada archivo, Linux tiene asociado un elemento en esta tabla que contiene un número. Este número identifica la ubicación del archivo dentro del área de datos.
- El **área de datos**, como su nombre indica, ocupa el resto del disco, equivalente a la zona de datos en MS-DOS. En esta zona están almacenados los ficheros y directorios de archivos.

Un directorio no es más que un archivo que contiene nombres de archivos (o directorios) con el número del i-nodo que contiene su información.

Bloque de carga			
Superbloque			
I-nodo	I-nodo	I-nodo	I-nodo
Área de datos			

Tipos de sistemas de archivos en Linux

Linux soporta una **gran variedad de sistemas de ficheros**, desde sistemas basados en discos, como pueden ser **ext2, ext3, ext4, ReiserFS, XFS, JFS, UFS, ISO9660, FAT, FAT32 y NTFS**, a sistemas de ficheros que sirven para comunicar equipos en la red de diferentes sistemas operativos, como **NFS** (utilizado para compartir recursos entre equipos Linux) o **SMB** (entre máquinas Linux y Windows).

Particionar un disco duro es realizar una **división** en él de modo que, a efectos prácticos, **el sistema operativo crea que tiene varios discos duros**, cuando en realidad **solo hay un único disco físico** dividido en varias partes. De este modo, se pueden modificar o borrar particiones sin afectar al resto de datos.

Las **particiones básicas** se llaman **primarias** y puede haber **como máximo cuatro**. Esto puede ser suficiente para nuestros intereses. Como a veces no es así, se crearon las **particiones extendidas** que pueden albergar otras particiones dentro, llamadas lógicas.

Los **sistemas de archivos** indican el **modo en que se gestionan los archivos dentro de las particiones**. Según su complejidad, tienen características como previsión de apagones, posibilidad de recuperar datos, indexación para búsquedas rápidas, reducción de la fragmentación para agilizar la lectura de los datos, etc. Hay varios tipos, normalmente ligados a sistemas operativos concretos.

A continuación se enumeran los más representativos:

- **FAT32 o VFAT**: es el **sistema de archivos tradicional de MS-DOS** y las primeras versiones de Windows. Por esta razón es considerado como un sistema universal, aunque tiene una gran desventaja que es su gran fragmentación y además es poco estable.
- **NTFS**: es el nuevo sistema de Windows, usado a partir de Windows NT 4.0. Es **muy estable**. El problema es que es **privativo**, con lo cual **otros sistemas operativos no pueden acceder a él de manera transparente**. Desde Linux solo se recomienda la lectura, siendo la escritura en estas particiones un poco arriesgada.

- **ext3:** es la versión mejorada de ext2, con **previsión de pérdida de datos por fallos del disco o apagones**. En contraprestación, es totalmente **imposible recuperar datos borrados**. Es compatible con el sistema de archivos ext2.
 - **Actualización.** Debido a que los dos sistemas comparten el mismo formato, es posible llevar a cabo una **actualización a ext3**, incluso aunque el sistema ext2 esté montado.
 - **Fiabilidad y rendimiento.**
- **ext4:** el sistema de archivos ext4 nace para **implementar algunas nuevas características al ext3**, dado que se hacía imposible incorporarlas al propio ext3. Algunas de sus principales ventajas son:
 - **Tasas de transferencia bastante superiores** a ext3 y **mejor rendimiento**.
 - Sistema de **ficheros de mayor tamaño**: permite tamaños de hasta 1 exabyte (1.024 x 1.024 TB).
 - Se **mejora la eficiencia de los descriptores de disco**, reduciendo los tiempos de borrado de ficheros largos, además de otras ventajas.
 - Más subdirectorios: **más de 32.000 subdirectorios** por directorio.
 - Apenas se fragmenta, pero añade soporte para la desfragmentación, que mejorará el rendimiento.
 - **Undelete.** El sistema de archivos ext4 soporta undelete (desborrado), herramienta para recuperar archivos que han sido borrados de forma accidental.
- **ReiserFS:** sistema de archivos para otras distribuciones de Linux, que entre otras cosas **gestiona muy eficientemente archivos pequeños** y permite **journaling**.
- **swap:** es el sistema de archivos para la **zona de intercambio de Linux**. Todos los sistemas Linux necesitan una partición de este tipo para **cargar los programas y no saturar la memoria RAM** cuando se excede su capacidad.

Directorios más importantes en Linux

Veamos algunos de los directorios más importantes en Linux y lo que contienen:

- ✓ El **directorio raíz** es /. De aquí cuelgan todos los directorios del sistema. Dentro del directorio raíz encontramos varios subdirectorios importantes.
- ✓ **/bin.** Contiene ficheros de comandos ejecutables utilizables por todos los usuarios. Aquí tenemos los programas que pueden lanzar todos los usuarios.
- ✓ **/sbin:** aquí encontramos los archivos de configuración del sistema.
- ✓ **/home:** es un directorio donde tenemos los directorios personales de todos los miembros del sistema.
- ✓ **/usr:** contiene utilidades y programas generales de usuario. Este directorio contiene los programas de uso general para todos los usuarios.
- ✓ **/dev:** contiene archivos especiales de bloques y caracteres asociados a dispositivos hardware. Aquí encontramos todos los dispositivos físicos del sistema (todo nuestro hardware).
- ✓ **/lib:** contiene librerías y compiladores del sistema. Contiene las bibliotecas necesarias para que se ejecuten los programas que tenemos en **/bin** y **/sbin** únicamente.

- ✓ **/etc**: directorio con ficheros de configuración y utilidades para la administración. Aquí encontramos los archivos de configuración del sistema.
- ✓ **/media**: contiene todas las unidades físicas que tenemos montadas: discos duros, DVD, pen drive, etc.

Camino o trayectorias en Linux

Los **camino, rutas o trayectorias en Linux tienen el mismo fundamento que en Windows**. Existen trayectorias relativas y absolutas. Los nombres completos de las trayectorias absolutas van desde el directorio raíz hasta el propio archivo o directorio. Hay que tener en cuenta que en Linux **no existe la referencia a ninguna unidad lógica**, solamente al directorio raíz de almacenamiento principal de la estructura de datos.

En cuanto a las **trayectorias relativas, su utilización es igual que en modo comando en Windows**.

Metacaracteres en Linux

En entorno comando, en muchas expresiones y órdenes de Linux se utiliza un **conjunto de caracteres con significado especial** para búsquedas o para operaciones sobre archivos o directorios. El manejo de trayectorias, caminos o rutas en Linux es igual que en Windows. La diferencia está en la forma de hacer referencia al directorio raíz.

En **Windows, los caracteres comodín** servían para sustituir determinados caracteres cuando hacíamos referencia a nombres de archivos o directorios. En **Linux**, a los caracteres comodín se les llama **metacaracteres** y son los que se recogen en la tabla.

Metacarácter	Significado
.	Concuerda con cualquier carácter (salvo fin de línea en modo multilínea).
()	Subexpresión o grupo.
[]	Conjunto de caracteres.
[-]	Rango de caracteres.
[^] o [!]	Excepto ese conjunto de caracteres.
	permite una alternativa para elegir entre dos expresiones.
//	Delimita una expresión regular.
\	Protege el siguiente metacarácter.

En la tabla siguiente vemos algunos ejemplos del uso de metacaracteres con la orden **ls**, **que es el equivalente a la orden DIR del entorno comando de Windows**.

Orden con la expresión	Referencia los archivos y directorios que
[ab]	Contengan el carácter a o el b .
archi[A-P]	Empiecen por archi y que a continuación tengan cualquier carácter comprendido entre A mayúscula y P mayúscula.
texto1[1-4]	Se llamen texto1 y el siguiente carácter sea un número comprendido entre 1 y 4 , ambos inclusive.
[1-3]???t*	Su nombre empiece por un número comprendido entre 1 y 3 , seguido de tres caracteres cualesquiera, una t y el resto sea cualquier cosa.

<code>[123]???t*</code>	Igual que el anterior.
<code>[dD]ocs</code>	Empiecen por D mayúscula o d minúscula y el resto del nombre sea ocs .
<code>[^0-9] o [!0-9]</code>	No tengan un número como primer carácter. Solamente se visualizarán los archivos que empiecen por una letra.
<code>texto[^123] o texto[!123]</code>	Se llamen texto y que el siguiente carácter no sea un 1 , un 2 o un 3 .
<code>le[ae]me</code>	Se llamen leame o leeme .
<code>[a-z]asa.txt</code>	Empiecen por cualquier letra en minúscula y las tres siguientes sean asa y la extensión sea .txt : casa.txt , tasa.txt , rasa.txt , etc.
<code>[A-Z]asa.txt</code>	Igual que el ejemplo anterior, pero la primera letra en mayúsculas: Casa.txt , Tasa.txt , Rasa.txt , etc.
<code>[a-zA-Z]asa.txt</code>	Los dos ejemplos anteriores unidos.

6. Ayuda en Linux

En entorno comando o terminal, cuando tengamos alguna duda de **cómo utilizar un comando específico**, acudiremos a la ayuda que ofrece el sistema operativo. Para obtener ayuda de los comandos, hay dos formas:

- ✓ Teclear en el terminal el **comando man** seguido de un **espacio en blanco** y del **comando sobre el que solicitamos la ayuda**.
- ✓ Teclear el **comando** en el símbolo del sistema, un **espacio en blanco** seguido de **--help**.

De esta forma, siempre podremos conocer las múltiples opciones y parámetros que se utilizan con los comandos para determinar las acciones a realizar.

7. Operaciones sobre directorios y carpetas

Las operaciones sobre directorios tienen equivalencia directa con lo que hemos visto en Windows, por lo que solamente se indicará con qué instrucciones se realiza cada una de las operaciones deseadas y, si existieran, las diferencias respecto de las órdenes utilizadas en Windows.

Cómo crear directorios o carpetas

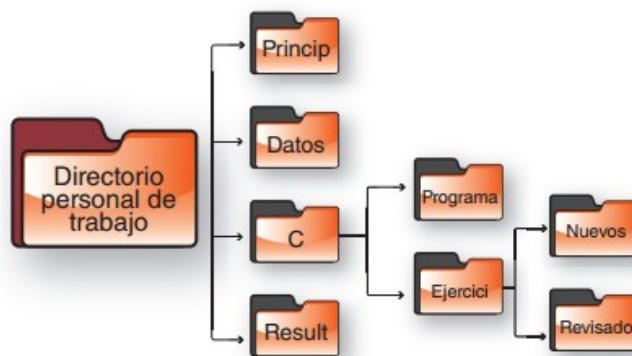
Para crear una carpeta en entorno gráfico, tendremos que tener abierta la carpeta sobre la que queremos crear otra u otras carpetas. Esta carpeta principal, llamada carpeta padre, será de la que cuelgue la estructura de carpetas o directorios que deseamos crear.

Si utilizamos la vista en modo árbol, en la parte derecha tendremos que estar viendo el contenido de la carpeta padre, y, si no usamos este tipo de vista, tendremos que tener en la ventana actual el contenido de la carpeta padre.

Para crear la nueva carpeta pulsaremos el botón derecho del ratón sobre cualquier zona que no tenga iconos. Del menú contextual seleccionaremos la opción **Crear una carpeta**.

Para crear la estructura de directorios de la figura dentro de tu carpeta personal **/home/nombre_usuario**, mediante la interfaz de texto o símbolo del sistema. Suponiendo que el **usuario** que ha iniciado sesión en el equipo es **paco**, indicará lo siguiente en el prompt: **paco@ubuntupaco:~\$**.


```
paco@ubuntupaco:~$ mkdir Princip Datos C Result
paco@ubuntupaco:~$ mkdir C/Programa C/Ejercici
paco@ubuntupaco:~$ mkdir C/Ejercici/Nuevos C/Ejercici/Revisado
```



Ver el contenido de los directorios o carpetas y moverse por ellos

Si en entorno gráfico utilizamos la **vista en modo árbol**, para ver el contenido de una carpeta haremos clic sobre ella. De esta forma, en la parte derecha de la ventana podremos ver su contenido. En este caso, se mostrará la información de carpetas que contiene así como los ficheros, si los hubiera. **La forma más habitual de ver el contenido de una carpeta es haciendo doble clic sobre ella**, dependiendo de cómo tengamos personalizado el aspecto de las ventanas.

Para **ir hacia delante** haremos doble clic sobre la carpeta deseada, y así sucesivamente hasta llegar al archivo o carpeta elegido. Si queremos **ir hacia atrás**, y dependiendo de la versión de Linux que utilicemos, lo podremos hacer pulsando el icono **Subir, Atrás** o **Adelante**.

Sitúate en tu directorio o carpeta personal. Usa la misma estructura del ejemplo anterior y ve hasta el directorio **Ejercici** y muestra su contenido. Después, sitúate de nuevo en tu carpeta de trabajo.

```
paco@ubuntupaco:~$ cd C/Ejercici
paco@ubuntupaco:~/c/ejercici$ ls -la
paco@ubuntupaco:~$ cd ../../
```

Visualizar la estructura de directorios o carpetas

Para visualizar en entorno gráfico la estructura de árbol de todas las carpetas hemos utilizado el Explorador de Windows. Ya sabemos que en Linux podemos hacerlo poniendo la vista de la carpeta en modo Árbol. En **entorno comando de Windows**, para visualizar la estructura de subdirectorios de un directorio concreto utilizábamos la orden **TREE**.

En Linux utilizaremos un **modificador para el comando ls**. El modificador es **-R**. Con él, visualizamos la estructura de directorios creada anteriormente.

```
paco@ubuntupaco:~$ ls -R
```

Eliminar directorios o carpetas

Esta operación es de las más sencillas. Para eliminar una carpeta en entorno gráfico nos situaremos sobre ella, la seleccionaremos y pulsaremos la tecla **Supr** o el **botón derecho del ratón** para seleccionar la opción **Mover a la papelera**.

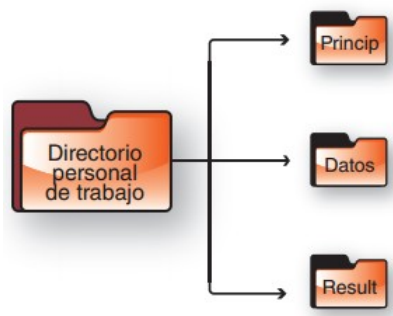
Si queremos eliminarla definitivamente, iremos a la **Papelera** y **eliminaremos lo que queramos de forma permanente**. También podemos eliminar de forma permanente un archivo o carpeta cuando lo tengamos seleccionado. Pulsaremos la tecla **Shift y, sin soltarla, pulsaremos la tecla Supr**.

Si, por alguna circunstancia, no se pudiera eliminar el contenido de una carpeta, incluida ella misma, será porque hay archivos o carpetas dentro de ella que tienen determinadas propiedades o atributos y el propio sistema no nos permite eliminarlos. También se nos puede denegar la eliminación de una carpeta si tenemos archivos abiertos que pertenezcan a la carpeta que queremos eliminar.

Elimina completa y definitivamente de la estructura el **directorio C** y todo lo que dependa de él.

```
paco@ubuntupaco:~$ rmdir C/Ejercici/Nuevos C/Ejercici/Revisado
paco@ubuntupaco:~$ rmdir C/Ejercici C/Programa
paco@ubuntupaco:~$ rmdir C
```

La orden **rmdir** no borra directorios que tengan algún contenido. Sin embargo, con la **orden rm con algunos parámetros es posible realizar un borrado completo de un directorio con o sin contenido**. La nueva estructura de directorios quedaría como se muestra en la figura.



Volvemos a crear lo que falta de la estructura de directorios para tener la misma que en el ejemplo inicial. Copiamos el directorio **Ejercici** y todo su contenido en el directorio **Princip**. Después, eliminamos toda la estructura de directorios de **Ejercici**, el origen de la copia.

```
paco@ubuntupaco:~$ cp -R C/Ejercici Princip
paco@ubuntupaco:~$ rmdir C/Ejercici/Nuevos C/Ejercici/ Revisado
paco@ubuntupaco:~$ rmdir C/Ejercici
```

Mover y copiar directorios o carpetas

En entorno gráfico esta operación se realiza de la misma forma que en Windows. Por esta razón, tendremos en cuenta las mismas consideraciones para realizar este tipo de operaciones.

- ✓ **Copiar**. Si lo que queremos hacer es copiar las carpetas o archivos a otro lugar.
- ✓ **Cortar**. Si lo que queremos hacer es mover.
- ✓ **Pegar**. Materializa la acción de copiar o cortar en el destino adecuado, teniendo en cuenta que cuando se copia se mantiene el archivo o carpeta original y cuando se corta, el archivo original desaparece.
- ✓ **Copiar a y Mover a**. Directamente al escritorio o a nuestra carpeta personal.

Sitúate en la carpeta **Ejercici** y **cambia el nombre de la carpeta Datos por el nombre Textos**. Para realizar esta operación en modo texto, primero nos tendremos que situar en la carpeta Ejercici.

```
paco@ubuntupaco:~$ cd Princip/Ejercici
```

Desde este punto, nos situaremos de nuevo en nuestro directorio de trabajo:

```
paco@ubuntupaco:~$ cd ../../
paco@ubuntupaco:~$ mv Datos Textos
```

Situados en el directorio **Ejercici**, también podemos ejecutar una única orden:

```
paco@ubuntupaco:~$ mv ../../Datos ../../Textos
```

Cambiar el nombre a un directorio o carpeta

Para realizar esta operación en entorno gráfico, simplemente seleccionaremos la carpeta con el botón izquierdo del ratón y **pulsaremos la tecla F2**. De esta forma editamos el nombre de la carpeta para poder cambiarlo.

Otra manera de realizar esta operación es haciendo **clic con el botón derecho del ratón en la carpeta** y seleccionando la opción **Renombrar**.

8. Atributos de los directivos o carpetas

En Linux, cuando visualizamos la estructura de directorios comprobamos que cada archivo queda identificado por **diez caracteres**. De estos diez caracteres, **el primero empezando por la izquierda hace referencia al tipo de archivo** que es. El resto, es decir, los **nueve siguientes** de izquierda a derecha y en bloques de tres, hacen referencia a los **permisos** que se le conceden respectivamente al **propietario**, al **grupo** y al **resto de usuarios** del equipo.

PROPIEDADES O PERMISOS	ATRIBUTOS	ARCHIVOS
d rwx r-x r-x	2 root root 4096 2008-07-24 09:35	.
d rwx r-x r-x	37 root root 4096 2008-07-24 09:24	..
- rw- r-- r--	1 root root 184 2008-07-21 12:50	1111.desktop
- rw- r-- r--	1 root root 7926 2008-07-21 18:57	alacarte.desktop
l rwx rwx rwx	1 root root 44 2008-07-20 11:53	/etc/kubuntu-default-settings/directory-home

Los tipos de permisos sobre directorios son:

- ✓ **Lectura (r)**: permite saber **qué archivos y directorios contiene el directorio que tiene este permiso**. Concretamente podremos utilizar órdenes como **ls**.
- ✓ **Escritura (w)**: permite **crear archivos en el directorio**, bien sean archivos ordinarios o nuevos directorios. Se pueden borrar directorios, copiar archivos en el directorio, mover o cambiar el nombre entre otras.
- ✓ **Ejecución (x)**: permite **situarse sobre el directorio para poder examinar su contenido y copiar archivos de él**. Si además se dispone de los permisos de escritura y lectura, se podrán realizar todas las operaciones posibles sobre archivos y directorios.

Concretamente, el primer carácter de los directorios, que podemos ver en la tabla siguiente (como archivos que son) puede ser el siguiente:

Carácter	Identifica
-	Archivo ordinario.
d	Directorio.
b	Archivo de bloques especiales.
c	Archivo de caracteres especiales.
l	Archivo de vínculo o enlace.
p	Archivo especial de cauce.

Los **siguientes nueve caracteres son los permisos que se les conceden a los usuarios del sistema**. Cada **tres caracteres**, hacen referencia a los permisos de **propietario, grupo y resto de usuarios** respectivamente. Dependiendo de si los permisos se asignan a **archivos ordinarios o directorios**, su significado no es exactamente el mismo.

A los diez caracteres que acompañan a cada archivo y directorio en Linux se le suele denominar **máscara**. Esta máscara **determina el tipo de operaciones que se pueden realizar** sobre el directorio o archivo **dependiendo del usuario** que haya iniciado sesión o se conecte al sistema.

Todo archivo o directorio en Linux **es propiedad del usuario que lo creó**. Cada usuario necesariamente pertenece a un **grupo de usuarios**. Por último, todo archivo o directorio puede ser utilizado por **Otros usuarios** que no sean los usuarios del grupo al que pertenece el propietario.

Permiso	Identifica
u	Propietario.
g	Grupo al que pertenece el usuario.
o	Resto de usuarios.

Permiso	Identifica
-	Sin permiso.
r	Permiso de lectura.
w	Permiso de escritura.
x	Permiso de ejecución.

Permiso	Identifica
-	Elimina derechos.
=	Asigna permisos.
+	Añade derechos.

La **asignación y eliminación de privilegios** en los directorios o carpetas **se puede hacer en entorno gráfico o en modo texto**, como la mayoría de las operaciones en Linux. En entorno gráfico, podemos ver los atributos o máscara de un archivo o directorio seleccionando la carpeta deseada y haciendo clic con el botón derecho del ratón en **Propiedades**.

En este punto es importante indicar que **root como súper usuario (administrador) del sistema podrá hacer cualquier cosa en cuanto a privilegios se refiere**.

Por el contrario, cada usuario hará lo que quiera sobre sus directorios o carpetas, pero sobre los demás solo podrá hacer lo que **root** le haya asignado. En entorno comando o terminal se utiliza un conjunto de órdenes para modificar, asignar o eliminar privilegios sobre archivos o directorios: la orden **chmod**.

La orden **ATTRIB de Windows** es la única que tiene **algo de semejanza con la gestión de privilegios**, aunque en Windows solamente podemos otorgar a un archivo o directorio unos privilegios muy escasos. Con la orden **chmod** podremos quitar o eliminar derechos a cada tipo de usuario.

Si no se especifica el tipo de usuario al que le queremos quitar, poner o asignar privilegios, lo que haremos será realizar la operación **a todos los usuarios simultáneamente**. La sintaxis de esta orden es muy similar a la orden attrib. Se especifica **la orden**, seguida del **tipo de usuario** sobre el que queremos actuar, el **carácter +, - o =**, **tipo de permiso de privilegios** y **archivo o directorio**.

ORDEN	RESULTADO
<code>\$chmod g+x doc1</code>	Con esta orden lo que estamos haciendo es otorgar privilegios de ejecución al grupo al que pertenece el archivo llamado doc1 .
<code>\$chmod rwx doc1</code>	Se asignan los privilegios de lectura, escritura y ejecución a todos los usuarios para el archivo doc1 .
<code>\$chmod go-wx doc1</code>	Se quitan los privilegios de escritura y ejecución al grupo y al resto de usuarios del archivo doc1 .
<code>\$chmod =x doc1</code>	Asigna a todos los usuarios el permiso de ejecución. Esta misma orden también se podría poner <code>\$chmod ugo+x doc1</code> .
<code>\$chmod = doc1</code>	Quita todos los privilegios a todos los usuarios del archivo doc1 .

El que asigna privilegios solamente tendrá que tener en cuenta cuáles de ellos quiere asignar y realizar un pequeño análisis de su equivalencia con el número en octal y el correspondiente desglose en binario.

Vamos a ver los permisos de todas las carpetas de nuestro ejercicio y a modificarlos. En primer lugar vamos a ver los privilegios de las carpetas creadas. Ejecutaremos la siguiente orden y veremos algo como:

```
paco@ubuntupaco:~$ ls -la
drwxr-xr-x 6 root root 4096 2012-07-24 09:58 .
drwxr-xr-x 38 root root 4096 2012-07-24 09:59 ..
drwxr-xr-x 2 root root 4096 2012-07-24 09:57 c
drwxrwxr-x 2 root root 4096 2012-07-24 10:12 datos
drwxr-xr-x 2 root root 4096 2012-07-24 09:57 princip
drwxr-xr-x 2 root root 4096 2012-07-24 09:58 result
```

Como podemos apreciar, todo lo que tenemos en nuestro directorio **son directorios ya que el primer carácter de la máscara es la letra d**. Se puede ver que el **propietario tiene privilegios de rwx**, el **grupo r-x** y los **otros r-x**. Si queremos hacer que el **grupo** de usuarios tenga **privilegios** de escritura sobre la carpeta **Princip**, ejecutaremos la siguiente orden:

```
paco@ubuntupaco:~$ chmod g+w princip
```

Con la misma orden (`ls -la`) podremos comprobar que el nuevo privilegio está puesto. Para **quitar a usuarios, grupo y otros el privilegio de lectura** sobre Princip, podremos hacerlo con cualquiera de las siguientes ordenes:

```
paco@ubuntupaco:~$ chmod ugo-r princip
paco@ubuntupaco:~$ chmod -r princip
```

Para dejar la estructura de archivos igual que estaba al principio haremos lo siguiente:

```
paco@ubuntupaco:~$ chmod g-w princip
paco@ubuntupaco:~$ chmod ugo=rx princip
paco@ubuntupaco:~$ chmod u+w princip
```

Hay otra forma de usar la orden **chmod** que, para muchos usuarios, resulta más cómoda de utilizar, aunque a priori sea algo más compleja de entender. Tenemos que tener en cuenta que **cada uno de los tres caracteres que representan los privilegios para cada tipo de usuarios se pueden representar mediante la combinación de tres dígitos en octal**.

Los caracteres binarios tenían correspondencia con dígitos octales y hexadecimales directamente. Recordemos que **en base 8** los dígitos utilizados **son desde el 0 al 7, ambos incluidos**. Sabemos que un dígito octal lo podemos representar con sus correspondientes dígitos en binario: $2^0 = 1$, $2^1 = 2$ y $2^2 = 4$.

Situando posicionalmente, los dígitos binarios según el exponente de menor a mayor, de derecha a izquierda y empezando por el exponente cero, podemos obtener la correspondencia con los dígitos octales. Por ejemplo, 3 en binario es 010. **Los bits uno multiplican a la potencia a la que corresponden posicionalmente y los bits cero no.** Por lo tanto, en octal: $0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 = 3$.

En este caso, no relación de los dígitos binarios con potencias de 2, sino con los privilegios. En la tabla podemos ver una **correspondencia entre el dígito octal con sus caracteres binarios y los privilegios** que representan. Al igual que en los sistemas de numeración, cuando el dígito binario es un 1 indica que esa potencia entra a formar parte del dígito octal, y cuando es un 0 indica que no entra a formar parte.

DÍGITO OCTAL	DÍGITO BINARIO		
	$r = 2^2$	$w = 2^1$	$x = 2^0$
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Realizaremos las mismas operaciones que en el ejemplo anterior, pero utilizando el formato numérico. Vamos a dar al grupo el permiso de escritura sobre la carpeta **Princip**. Para ello ejecutaremos la orden:

```
paco@ubuntupaco:~$ chmod 775 princip
```

El número 775 pasado a binario es: **111 111 101**, y su equivalente a permisos es **rwX rwX r-X**. Se puede ver que, **por defecto, las carpetas** tienen unos **permisos octales** con valor de 755. Para quitar a usuarios, grupo y otros el privilegio de lectura sobre Princip, primero calcularemos la máscara en función de los privilegios que queremos obtener. Buscamos que los **privilegios sobre Princip en definitiva sean: -rx -rx --x**. Podremos hacerlo con la siguiente orden:

- **Usuarios:** queremos obtener **-rx** equivalente a **011 = 3**.
- **Grupo:** queremos obtener **-rx** equivalente a **011 = 3**.
- **Otros:** queremos obtener **--x** equivalente a **001 = 1**.

Por lo tanto, la orden a ejecutar será:

```
paco@ubuntupaco:~$ chmod 331 princip
```

Para dejar la estructura de archivos igual que estaba al principio haremos lo siguiente:

```
paco@ubuntupaco:~$ chmod 755 princip
```