

# Linux filesystem

## Concepto de archivo

Un archivo **es una colección de información relacionada**, con un nombre, que se graba en **almacenamiento secundario** siendo esta la unidad más pequeña de almacenamiento y **tiene una determinada estructura definida** que dependerá de su tipo. Particularmente un archivo **ejecutable es una serie de secciones de código** que el cargador puede cargar en memoria y ejecutar.

## Atributos de archivo

Para **referirnos** a los archivos **utilizamos un nombre**. Los **atributos** típicamente son los siguientes:

- ☐ **Tamaño:** Tamaño del archivo
- ☐ **Dispositivo:** El dispositivo de almacenamiento en el cual se encuentra
- ☐ **Identificador (i-node).** Número que identifica el archivo dentro del sistema de archivos
- ☐ **Hard links.** Número de enlaces duros al archivo.
- ☐ **Tipo.** Para sistemas que soporten diferentes tipos de archivos.
- ☐ **Ubicación.** Puntero a un dispositivo y a la ubicación del archivo dentro de dicho dispositivo.
- ☐ **Tamaño.** Tamaño actual del archivo.
- ☐ **Protección.** Quién puede leer el archivo, escribir en el archivo, ejecutarlo, etc. De aquí podremos resaltar tres cosas:
  - **UID:** Id del usuario dueño del archiv
  - **GID:** Grupo al que pertenece el archivo.
  - **Fecha, hora:** Creación, acceso etc...

La **información** acerca **de los archivos se almacena en la estructura de directorios** compuesta del nombre de un **archivo** y de su **i-nodo** unívoco que permite localizar los demás atributos del archivo.

## Operaciones con los archivos

El sistema operativo puede proporcionar llamadas al sistema para

- **Creación de un archivo.** Para esto hay que encontrar espacio para el archivo dentro del sistema de archivos e incluirlo en el directorio creando una entrada para el nuevo archivo.

- **Escribir un archivo:** Es una llamada al sistema. El sistema debe mantener **un puntero de escritura que haga referencia a la ubicación dentro del archivo** en la que **debe tener lugar la siguiente escritura**. El puntero de escritura debe actualizarse cada vez que se escriba en el archivo nueva información.
- **Lectura de un archivo.** Es una llamada al sistema. El sistema necesitará mantener un **puntero de lectura** que haga **referencia a la ubicación dentro del archivo en la que tiene que tener lugar la siguiente lectura**. Una vez que la lectura se ha completado, **se actualiza el puntero de lectura**.
- **Reposicionamiento dentro de un archivo.** Se reposiciona el puntero de posición actual dentro de un archivo.
- **Borrado de un archivo.** Para borrar un archivo, **liberamos todo el espacio del archivo**, de modo que pueda ser reutilizado por otros archivos, y **borramos también la propia entrada del directorio y la entrada de archivos abiertos**.
- **Truncado.** Vacía el archivo manteniendo sus atributos, excepto la longitud del archivo.

## Tipos de archivo

Se refiere a como el sistema operativo debe reconocer y soportar el concepto de tipo de archivo. Si un sistema operativo reconoce el tipo de un archivo, podrá operar con ese archivo de formas razonables.

**El sistema UNIX utiliza un simple número mágico** almacenado al principio de algunos archivos para indicar, aproximadamente, el tipo del archivo.

## EXT4 (AKA Linux) filesystem

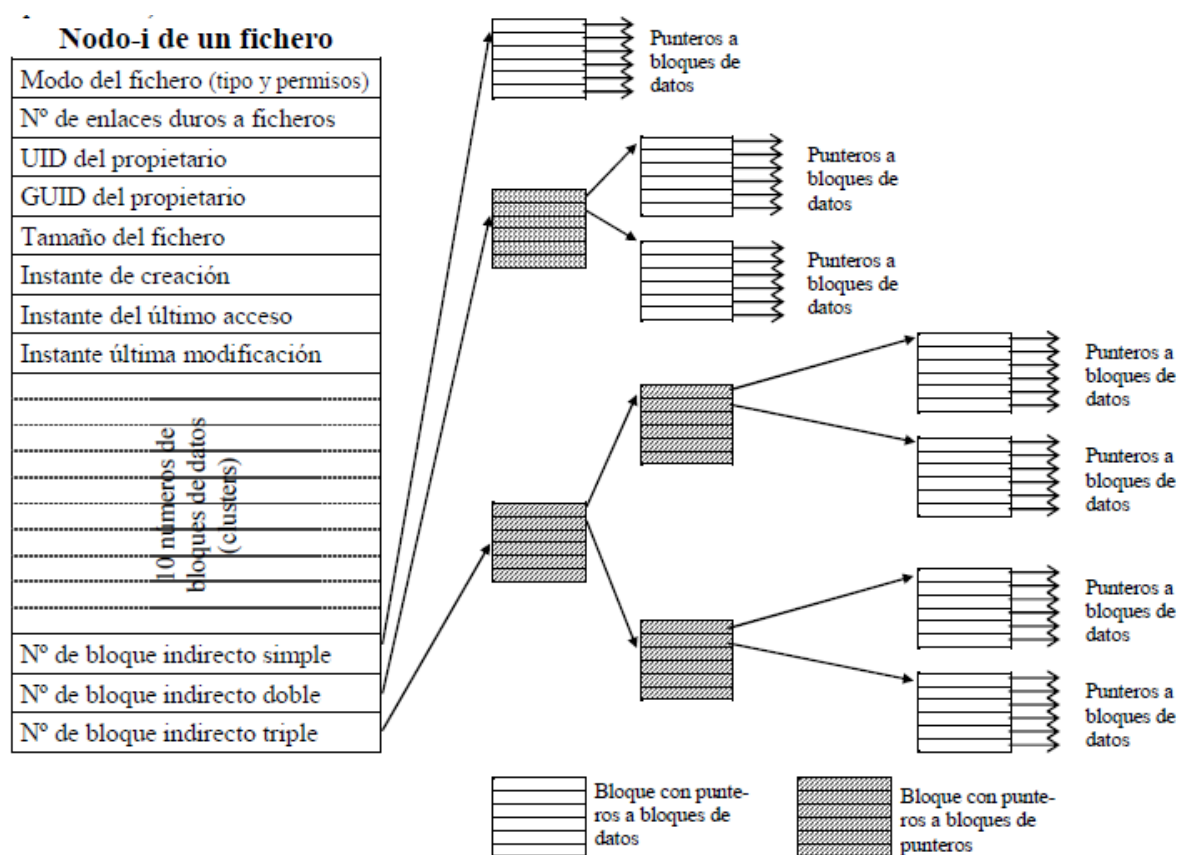
El filesystem con el que contaremos por defecto dentro de los sistemas de archivos **dentro de Linux es el conocido como ext4**. Este filesystem define **la forma en que leemos, escribimos y únicamente puede ser accedido si es montado**.

Este es un tipo de filesystem que dentro de sus características abordaremos dos principales:

1. La primera es que es del tipo transaccional lo que implica que cuenta con un Journaling.
2. La segunda es que estamos ante un filesystem que utiliza **i-nodos** y esta es tal vez la parte más complicada de entender.

Estos i-nodos, que los encontraremos a partir de un lugar específico **al principio del filesystem que formateamos previamente**, es una estructura de datos. Esto implica que cada uno de estos i-nodos tiene un tamaño fijo, y cada registro dentro del mismo

también. A su vez la información que contienen estos se la conoce como **metadata**. Básicamente, lo que vimos antes como **atributos de archivos**.



Esta estructura de datos llamada i-nodos aparte cumple no solo otorgar la metadata del archivo sino que también **indica las posiciones de los bloques que conforman el archivo dentro del dispositivo de almacenamiento secundario**.

Como vemos en la imagen anterior, hay una cierta cantidad máxima de bloques que se pueden direccionar por i-nodos, por lo que **¿Que pasa si mi archivo ocupa más bloques que el máximo por i-nodo?**.

Si este es el caso utilizaremos lo que se conoce como **direccionamiento indirecto**. Esto es que cada uno de estos registros en lugar de apuntar a un bloque apunta a **otro i-nodo**. Esto, de manera inmediata, nos muestra que **por cada registro directo de inodo se transforma en 10 bloques**.

Momento, aún nos queda una cosa más para terminar de entender **ext4**. Hablemos de Hardlink.

## Hardlinks

En Linux los archivos no se referencian de forma única por su nombre, sino que para el sistema operativo **un archivo es su i-nodo**. Esto quiere decir que **podremos tener diferentes entradas al mismo i-nodo en diferentes carpetas**.

Pero que quiere decir esto. Bien, ¿recuerdan que habíamos dicho que los directorios tenían una entrada por cada archivo, y que cada entrada tenía un i-nodo?. Bien, lo que queremos decir es que en diferentes archivos podemos tener una entrada en cada carpeta donde el nombre sea diferente; **pero el nodo sea el mismo**. Entonces, siempre que se tenga el mismo i-nodo **se va a referenciar exactamente al mismo archivo**.

Y el enlace duro lo podemos sintetizar en **la cantidad de veces que dentro del filesystem se hace referencia a un mismo i-nodo**.

Pongamos todo junto

Ahora llegó el momento de entender como se accede al archivo

Directorio raiz	El nodo-i6 es de /usr		El nodo-i 26 es de /usr/ast																																																																	
<table><tr><td>1</td><td>.</td></tr><tr><td>1</td><td>..</td></tr><tr><td>4</td><td>bin</td></tr><tr><td>7</td><td>dev</td></tr><tr><td>14</td><td>lib</td></tr><tr><td>9</td><td>etc</td></tr><tr><td>6</td><td>usr</td></tr><tr><td>8</td><td>tmp</td></tr></table>	1	.	1	..	4	bin	7	dev	14	lib	9	etc	6	usr	8	tmp	<table><tr><td>modo</td><td></td></tr><tr><td>tamaño</td><td></td></tr><tr><td>tiempos</td><td></td></tr><tr><td>132</td><td></td></tr><tr><td></td><td></td></tr></table>	modo		tamaño		tiempos		132				<table><tr><td>6</td><td>.</td></tr><tr><td>1</td><td>..</td></tr><tr><td>19</td><td>dick</td></tr><tr><td>30</td><td>erick</td></tr><tr><td>51</td><td>jim</td></tr><tr><td>26</td><td>ast</td></tr><tr><td>45</td><td>bal</td></tr></table>	6	.	1	..	19	dick	30	erick	51	jim	26	ast	45	bal	<table><tr><td>modo</td><td></td></tr><tr><td>tamaño</td><td></td></tr><tr><td>tiempos</td><td></td></tr><tr><td>406</td><td></td></tr><tr><td></td><td></td></tr></table>	modo		tamaño		tiempos		406				<table><tr><td>26</td><td>.</td></tr><tr><td>6</td><td>..</td></tr><tr><td>64</td><td>grants</td></tr><tr><td>92</td><td>books</td></tr><tr><td>60</td><td>mboks</td></tr><tr><td>81</td><td>minix</td></tr><tr><td>17</td><td>scr</td></tr></table>	26	.	6	..	64	grants	92	books	60	mboks	81	minix	17	scr
1	.																																																																			
1	..																																																																			
4	bin																																																																			
7	dev																																																																			
14	lib																																																																			
9	etc																																																																			
6	usr																																																																			
8	tmp																																																																			
modo																																																																				
tamaño																																																																				
tiempos																																																																				
132																																																																				
6	.																																																																			
1	..																																																																			
19	dick																																																																			
30	erick																																																																			
51	jim																																																																			
26	ast																																																																			
45	bal																																																																			
modo																																																																				
tamaño																																																																				
tiempos																																																																				
406																																																																				
26	.																																																																			
6	..																																																																			
64	grants																																																																			
92	books																																																																			
60	mboks																																																																			
81	minix																																																																			
17	scr																																																																			
usr produce el nodo-i 6	El nodo-i 6 indica que /usr está en el bloque 132	/usr/ast es el nodo-i 26	El nodo-i 26 indica que /usr/ast es el bloque 406	/usr/ast/mbok es el nodo-i 60																																																																

Bien supongamos que queremos acceder al archivo **/usr/ast/mbok**. El proceso sería:

1. Accede al “/”. Dependiendo del OS puede ser **siempre el i\_nodo 1 o 2**. Y a partir de ahí arma el recorrido de la ruta.
2. De ahí el inodo le indica en que bloques del disco se encuentra el listado.
3. Obtiene el listado de esos bloques.
4. En el listado aparece la estructura de datos **nombre-inodo. Aquí va a buscar que inodo corresponde a usr**
5. Va a buscar el i-nodo 6
6. De ahí el inodo le indica en que bloques del disco se encuentra el listado.
7. Obtiene el listado de esos bloques.
8. En el listado aparece la estructura de datos **nombre-inodo. Aquí va a buscar que inodo corresponde a ast**
9. Va a buscar el i-nodo 26
10. De ahí el inodo le indica en que bloques del disco se encuentra el listado.
11. Obtiene el listado de esos bloques.
12. En el listado aparece la estructura de datos **nombre-inodo. Aquí va a buscar que inodo corresponde a mbox**
13. Va a buscar el i-nodo 60
14. Obtiene el listado de esos bloques.
15. Y ahora, como en el tipo de archivo dice que esto **no es un directorio sino un archivo común** no va a buscar el par nombre -inodo sino que ya lo procesa como un archivo