

Cátedra de Sistemas Operativos

Unidad 2 – Administración de Archivos

Parte 3 – **Sistema de Archivos de Windows y Linux**

2024

Temario – Sistemas de Archivos

Windows:

- Características de NTFS
- NTFS
- Implementación del sistema de archivos en NTFS
- MFT
- Archivos en NTFS

Linux:

- Ext2, ext3, ext4
- Implantación de directorios
- Archivos compartidos

Windows - Características de NTFS

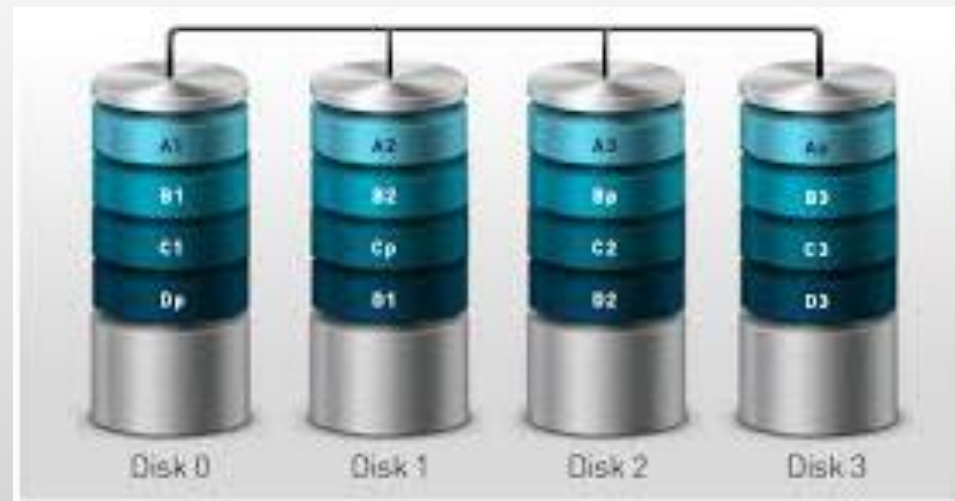
1. Recuperabilidad (Registro de transacciones)
2. Tolerancia a fallos
3. Compresión de archivos
4. Cifrado de archivos

1. Recuperabilidad (Registro de transacciones)

- Garantiza la integridad del sistema de archivos
- Permite recuperarse frente a fallos en el sistema y en el disco
- Mantiene un “registro de transacciones pendientes”
- Conocido como Journaling o archivos por bitácora
- Cuando se completa de manera exitosa una transacción, se elimina del registro
- Sólo se utiliza para la actualización de metadatos (actualización de directorios, actualización de cuotas de disco, etc.)
- Se implementan “transacciones atómicas”

2. Tolerancia a fallos

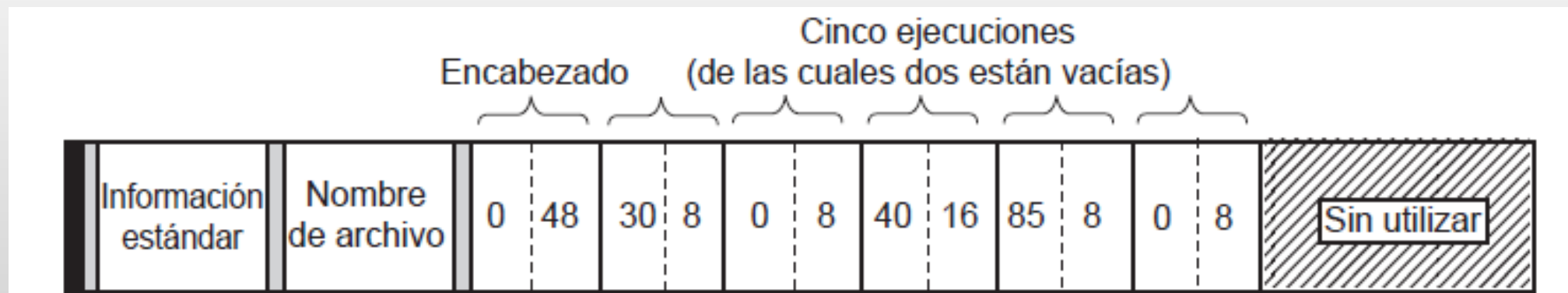
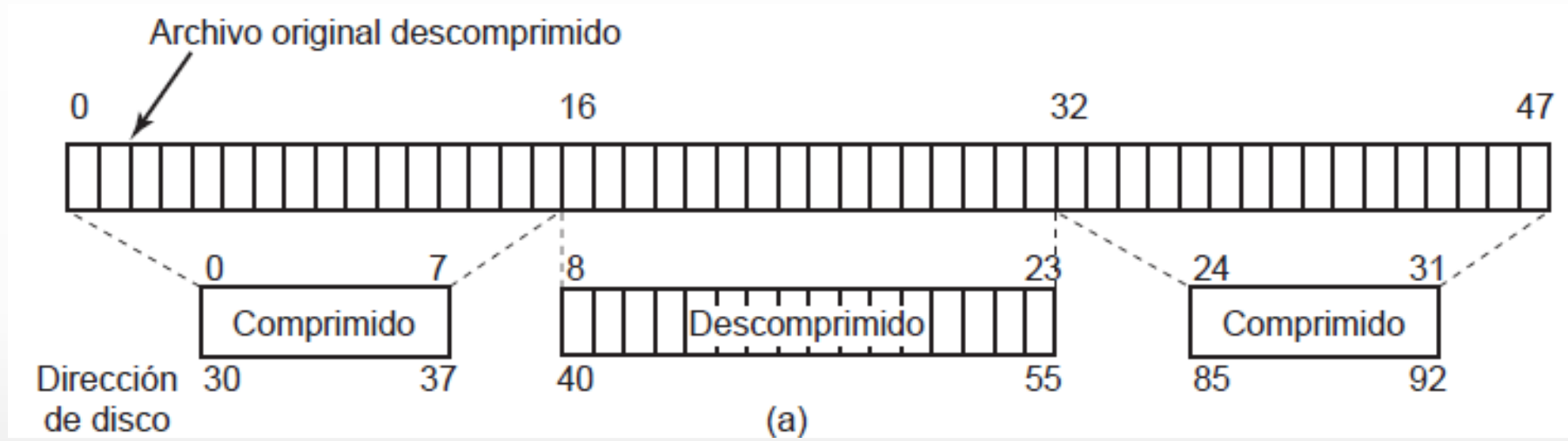
- Se brinda tolerancia a fallos mediante “redundancia”
- Objetivo → que el sistema siga funcionando a pesar de la falla de un disco
- Las versiones actuales de Windows soportan la tecnología RAID 1, RAID 5 y RAID 6



3. Compresión de archivos

- Se comprime un archivo de manera transparente
- Objetivo → ocupar menos espacio de disco
- Funcionamiento:
 - Se toman 16 bloques
 - Se aplica el algoritmo de compresión
 - Si se reduce al menos UN bloque, se almacena comprimido
 - Si no se reduce al menos UN bloque se almacenan los bloques SIN comprimir
 - Se toman los siguientes 16 bloques del archivo
- Inconveniente → lento acceso aleatorio

3. Compresión de archivos



4. Cifrado de archivos

- ¿Cómo se pueden proteger los archivos del disco?
- Seguridad → cifrar los archivos en el disco
- Se cifra la carpeta o directorio completo
- EFS (Encryption File System):
 - Driver que realiza el proceso de cifrado y descifrado
 - Cifra bloque por bloque con claves de 128 bits
 - Utiliza el algoritmo de cifrado DES

NTFS - New Technology File System

Características:

- Sistema de archivos complejo y sofisticado
- Utilizado en las últimas versiones de Windows
- Utiliza direcciones de disco de 64 bits
- Nombres de archivos de hasta 255 caracteres en Unicode
- Rutas completas hasta 32.767 caracteres
- Es un sistema de archivos jerárquico → utiliza “\” como separador

Implementación del sistema de archivos NTFS

Partición de NTFS:

Registro de arranque de Partición	MFT (Master File Table)	Archivos del Sistema	Área de Archivos (Datos del usuario)
--	--------------------------------	-----------------------------	---

- Contiene archivos, directorios, mapas de bits y otras estructuras de datos
- Un volumen (partición) → secuencia lineal de bloques o clústeres
- Tamaño del bloque → entre 512 bytes y 64 KB

MFT – Master File Table

- Principal estructura de datos de cada partición
- MFT → es un archivo de 2^{48} registros como máximo
- Tabla formada por una secuencia lineal de registros de tamaño fijo de 1 KB
- Cada registro de la MFT representa un archivo o directorio
- Cada registro contiene los atributos del archivo y la lista de direcciones de disco de sus bloques
- Si el archivo es muy grande puede ocupar varios registros de la tabla
- Primeros 16 registros de la MFT → archivos de metadatos de NTFS

MFT – Metadatos

#	Filename	Nombre	Descripción
0	\$MFT	Master File Table	Puntero a si mismo para consistencia del modelo.
1	\$MFTMirr	Master File Tabla Mirror	Puntero donde se encuentra el respaldo de la MFT. Solo se respaldan los registros más importante. En NT3.5 se ubica a la mitad de la partición, en versiones posteriores al final.
2	\$LogFile	Log File	Archivo de log de transacciones sobre el volumen.
3	\$Volume	Volume Descriptor	Contiene información crucial sobre la partición en si: versión NTFS, nombre, etc.
4	\$AttrDef	Attribute Definition Table	Nombres y tipos de los atributos que tendrá cada archivo.
5	\$	Root Directory	Puntero a donde parte el directorio raíz del sistema.
6	\$Bitmap	Cluster Allocation Bitmap	Mapa que muestra los clusters (bloques) disponibles
7	\$Boot	Volume Boot Code	Puntero al segmento que contiene el código de arranque, en el caso de que esta sea una partición activa (booteable)
8	\$BadClus	Bad Cluster File	Lista de todos los bloques "malos" del disco, para no volver a utilizarlos.
9	\$Secure	Security File	Contiene descriptores únicos para cada archivo del volumen
10	\$UpCase	Upper Case Table	Tabla de conversiones mayúsculas/minúsculas en UNICODE.
11	\$Extend	NTFS Extensions	Usado por varias extensiones como cuotas, repase, identif. de objeto, etc.

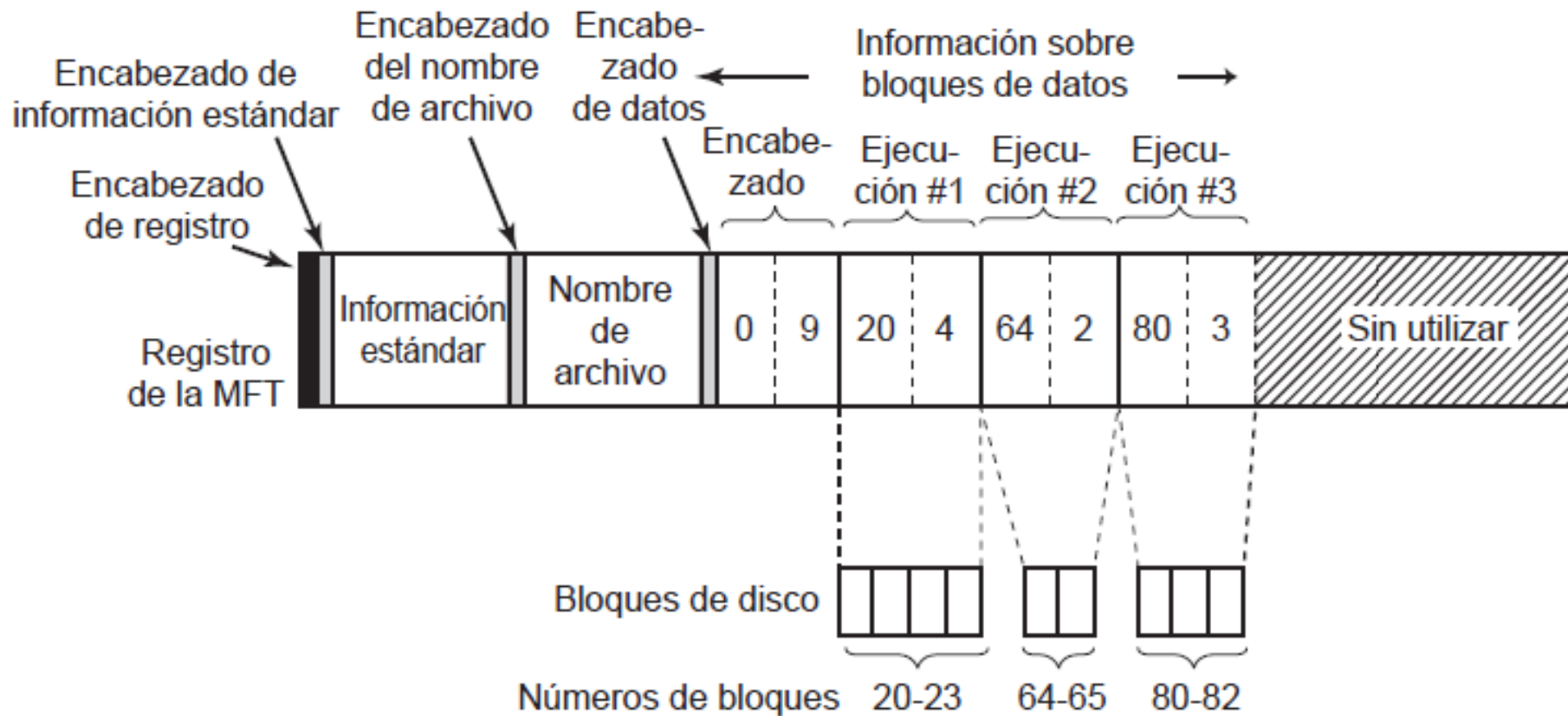
Archivos en NTFS

- Un archivo es un conjunto de atributos
- Cada atributo se representa como un flujo de bytes
- Archivo:
 - Flujos cortos → nombre e ID de objeto de 64 bits
 - Flujos largos → datos
- Flujo → nombre del archivo:nombre del flujo (archA:flujo1)

Tipos de Archivos en NTFS

- Archivos inmediatos → su contenido se almacena en el propio registro de la MFT
- Archivos normales → se asignan en bloques consecutivos
- Archivos dispersos → se asignan en varios conjuntos de bloques consecutivos
- Archivos muy dispersos → pueden ocupar varios registros de la MFT

Archivos dispersos

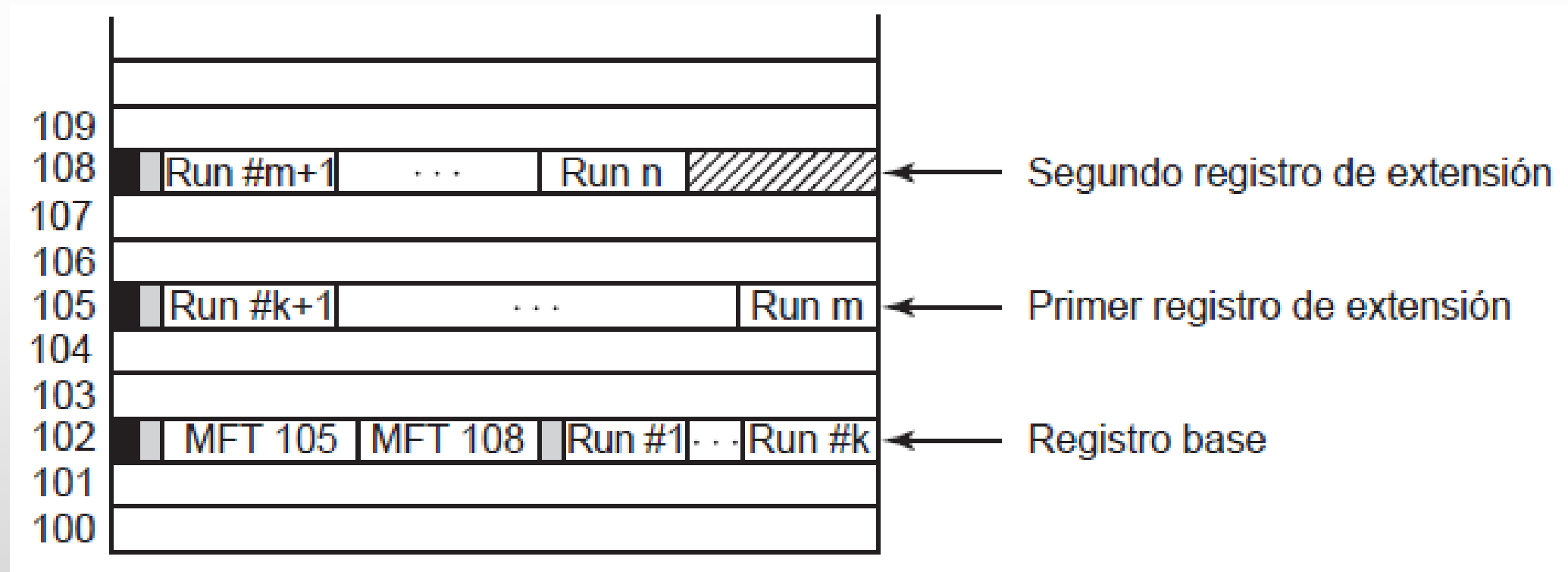


Aspectos a tener en cuenta

- **Ejemplo:** Vamos a suponer que un disco está dividido en bloques de 1KByte
- Caso 1:
 - ¿Puede un archivo de 10 Mbyte almacenarse en un único registro de la MFT?
- Caso 2:
 - ¿Puede un archivo de 100KByte, que está disperso en el disco en 100 bloques físicos separados, almacenarse en un solo registro de la MFT?

Archivos muy dispersos

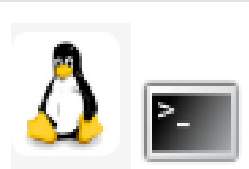
MFT – Master File Table



Linux – Sistema de archivos : Ext2



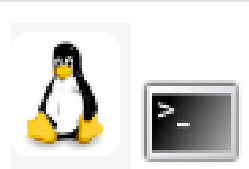
- Second Extended File System.
- Fue introducido en 1993.
- Desarrollado por Rémy Card.
- Desarrollado para superar las limitaciones del sistema de archivos originales ext.
- No tiene característica de journal.
- En las unidades flash, unidades USB, ext2, se recomienda, ya que no desgasta la memoria por hacer journaling.
- El tamaño máximo de archivo individual puede ser de 16 GB a 2 TB.
- En general el tamaño del sistema de archivos ext2 puede ser de 2 TB a 32 TB



Linux – Sistema de archivos : Ext3



- Tercer Sistema de Archivos extendido.
- Introducido en Noviembre de 2001 (Linux 2.4.15).
- Desarrollado por Stephen Tweedie.
- Implementa el *registro por diario o journaling*.
- Puede ser montado y usado como un sistema de archivos ext2.
- Utiliza un árbol binario balanceado.
- Incorpora el asignador de bloques de disco Orlov.
- Permite actualizar de ext2 a ext3 sin perder los datos almacenados, ni tener que formatear el disco.
- Tiene un menor consumo de CPU y está considerado más seguro que otros sistemas de ficheros en Linux dada su relativa sencillez y su mayor tiempo de prueba.

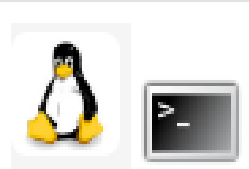


Tamaños de Ext3



- El número de máximo de directorios: 32000.
- Tiene dos límites de tamaño distintos: uno para archivos y otro para el tamaño del sistema de archivos entero.
- Limitaciones:

□	Tamaño del bloque	Tamaño máximo de los archivos	Tamaño máximo del sistema de ficheros
	1 KiB	16 GiB	2 TiB
	2 KiB	256 GiB	8 TiB
	4 KiB	2 TiB	16 TiB
	8 KiB	2 TiB	32 TiB



Linux – Sistema de archivos : Ext4



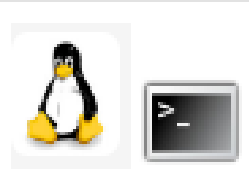
- Cuarto sistema de archivos extendido.
- Fue introducido en 2008.
- Desarrollado por Mingming Cao, Dave Kleikamp, Alex Tomas, Andrew Morton, y otros.
- A partir del kernel Linux 2.6.19 ext4 estuvo disponible.
- Compatible con gran tamaño de archivos individuales y el tamaño total del sistema de archivos.
- El tamaño máximo de archivo individual puede ser de 16 GB a 16 TB.
- El tamaño global máximo del sistema de archivos ext4 es 1 EB.
- El directorio puede contener un máximo de 64.000 subdirectorios.
- También puede montar un sistema de archivos ext3 existente como ext4 fs (*sin tener que actualizar*).
- Mejor rendimiento y fiabilidad.



Tamaños de Ext4



- El número de máximo de directorios: 64.000.
- El tamaño máximo de un archivo: 16 TB.
- Tamaño máximo del sistema de archivos: 1 EB.



Implementación de directorios

- Un directorio es un archivo especial del sistema que se utiliza para almacenar otros archivos y/o directorios
- Un directorio posee una “entrada” por cada archivo que posee
- Localización de un archivo → se utiliza la ruta para acceder al archivo
- Un aspecto importante es decidir dónde almacenar los atributos
- Se puede resolver de manera diferente:
 - En el propio directorio (FAT)
 - En el i-nodo
 - En cada registro de archivo de la MFT

Implementación de directorios

FAT – (File Allocation Table)

- Utilizado en la implementación de lista ligada y un índice
- Los atributos del archivo se registran en cada entrada del directorio
- Cada entrada de directorio posee los siguientes campos:

Nombre	Tipo	Atributos	Hora	Fecha	Número primer bloque	Tamaño
---------------	-------------	------------------	-------------	--------------	---------------------------------	---------------

Implementación de directorios

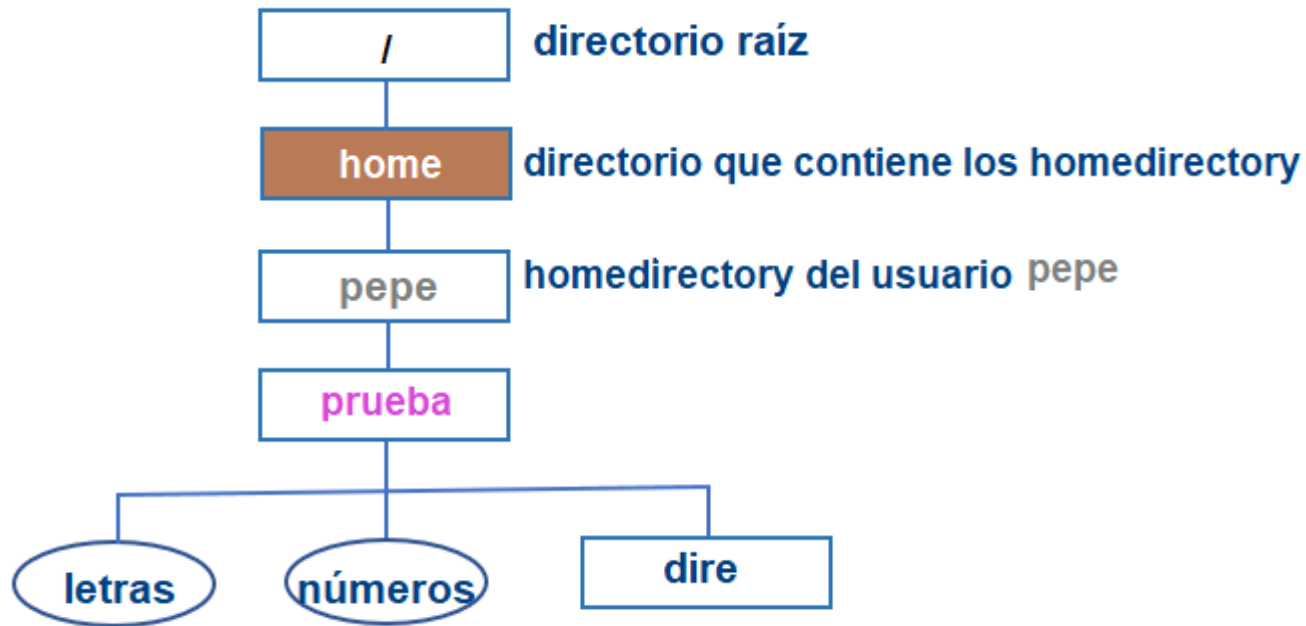
Linux

- Utilizado cuando se utilizan i-nodos
- Cada entrada de directorio posee:
 - Nombre del archivo
 - Número de i-nodo

Nombre Archivo	Nro. I-Nodo
A1	100
A2	105

- Directorios: Contienen información referida a los archivos ubicados en él.

Ejemplo: directorio **prueba**



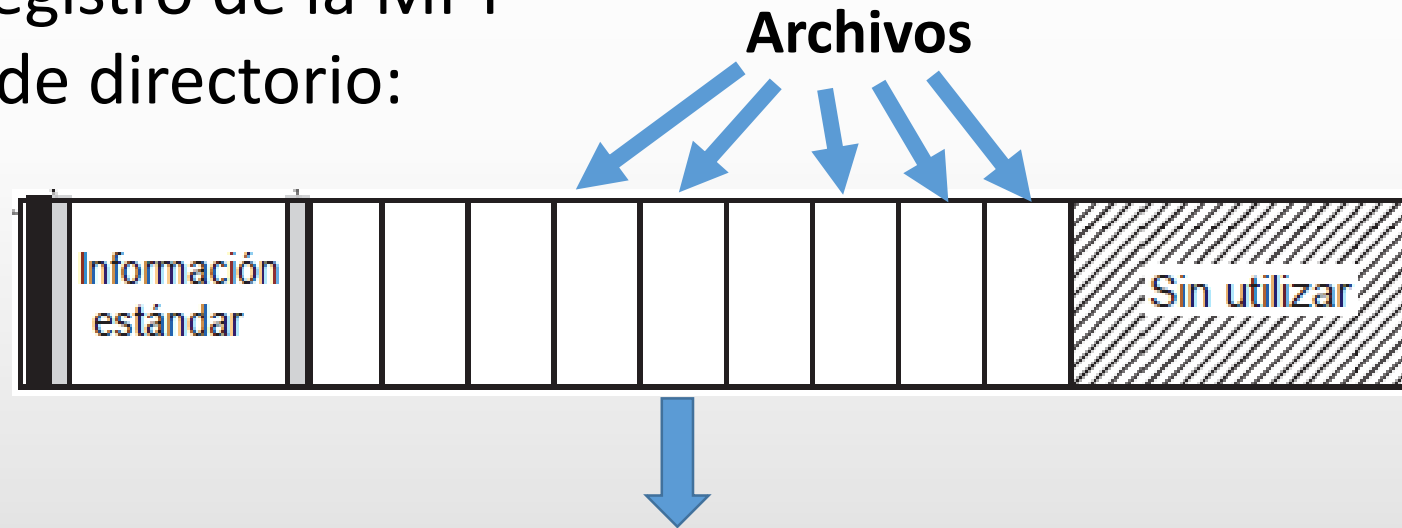
Número de nodo i nombre del archivo

1235	letras
2134	números
3127	<u>dire</u>

Implementación de directorios

NTFS

- Directorios pequeños → la información de los archivos está en un ÚNICO registro de la MFT
- Entrada de directorio:

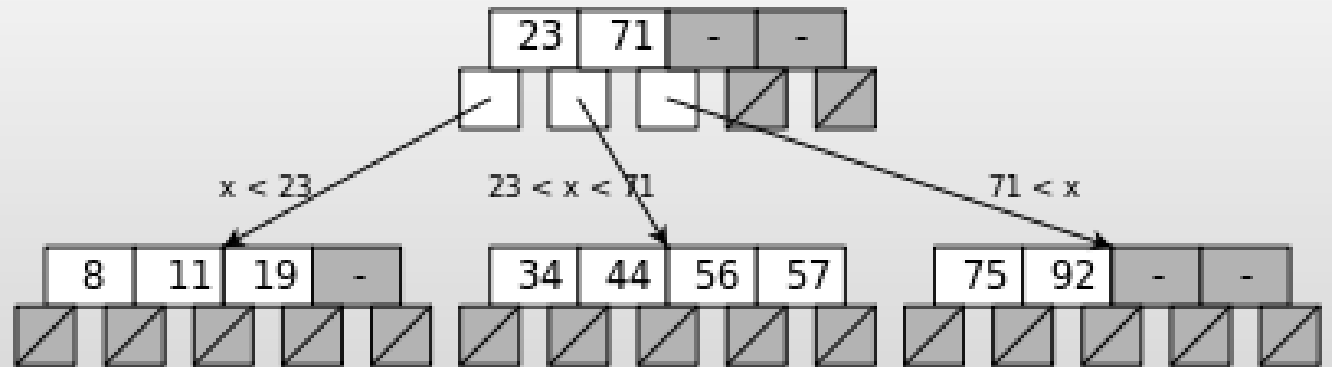


- Número de registro de la MFT
- Nombre del archivo
- Longitud del nombre del archivo

Implementación de directorios

NTFS

- Los directorios grandes utilizan un formato diferente
- Se utiliza un árbol B+:
 - Posibilita la búsqueda alfabética
 - Facilita la inserción de nuevos nombres en el directorio



Archivos compartidos

- Muchos usuarios pueden compartir archivos, por ejemplo en el disco rígido de un servidor
- Permiten trabajar en proyectos de manera conjunta
- ¿Copiar es lo mismo que enlazar?
- Un archivo puede ser accedido desde varios directorios
- Formas de compartir archivos:
 - Enlace Duro
 - Enlace Simbólico

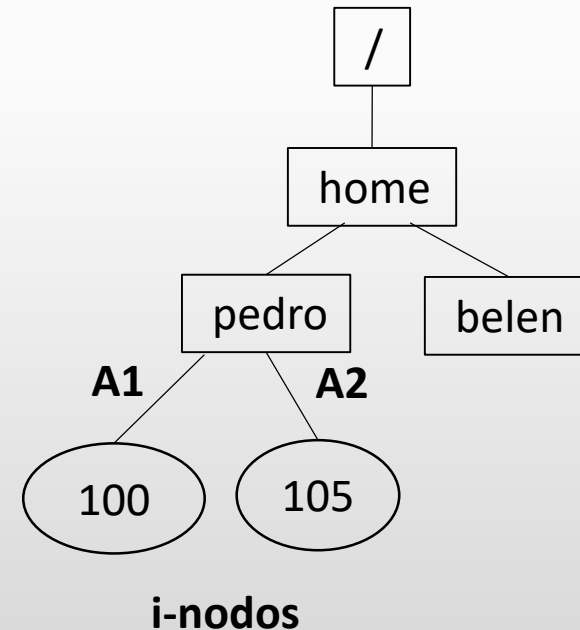
Archivos compartidos – Enlace duro

- Un archivo puede aparecer en más de un directorio. Incrementa el contador de i-nodos

Directorio de pedro

Nombre Archivo	Nro. I-nodo
A1	100
A2	105

ln /home/pedro/A2 /home/belen/L-A2



Archivos compartidos – Enlace duro

- Link → crea una entrada en un directorio. Un archivo puede aparecer en más de un directorio. Incrementa el contador de i-nodos

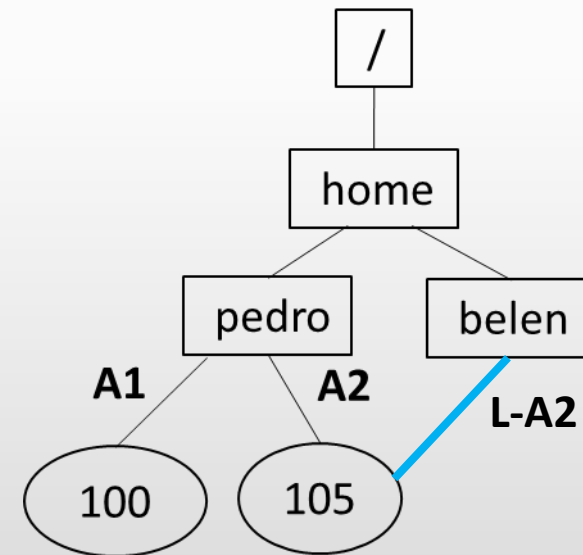
Directorio de pedro

Nombre Archivo	Nro. I-nodo
A1	100
A2	105

Directorio de belen

Nombre Archivo	Nro. I-nodo
L-A2	105

ln /home/pedro/A2 /home/belen/L-A2



El atributo número de enlaces al i-nodo 105 ahora vale 2 i-nodos

Archivos compartidos - Enlace duro

- Unlink → elimina una entrada de un directorio. Reduce el contador de i-nodos

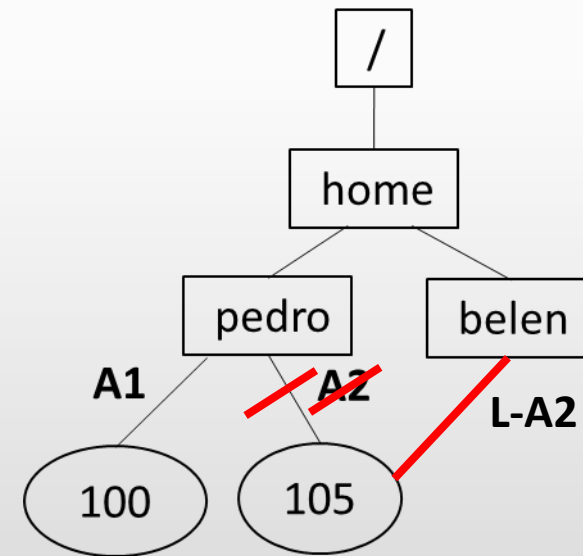
Directorio de pedro

Nombre Archivo	Nro. I-nodo
A1	100
A2	105

Directorio de belen

Nombre Archivo	Nro. I-nodo
L-A2	105

rm /home/pedro/A2



El atributo número de enlaces al i-nodo 105 ahora vale 1 i-nodos

Archivos compartidos - Enlace duro

- Inconveniente de los enlaces duros

Se enlaza el archivo desde el directorio de Belén

Directorio de pedro

Nombre Archivo	Nro. I- nodo
A2	105



Propietario = Pedro
Nro. Enlaces = 1

Directorio de pedro

Nombre Archivo	Nro. I- nodo
A2	105



Propietario = Pedro
Nro. Enlaces = 2

Directorio de belen

Nombre Archivo	Nro. I- nodo
L-A2	105



Directorio de belen

Nombre Archivo	Nro. I- nodo
L-A2	105



Propietario = Pedro
Nro. Enlaces = 1

¿Qué sucederá si Pedro borra el archivo A2?

Archivos compartidos

Enlaces simbólicos

- Permiten compartir archivos
- El sistema crea un archivo de tipo LINK el cual contiene la “ruta completa” al archivo original
- Cuando se accede al archivo vinculado, el sistema operativo sigue la ruta hasta llegar al archivo original
- En Windows se conocen como “accesos directos”
- En Linux se crean con el comando **# ln -s**
- ¿Qué sucede si se borra el archivo original?
- Ventaja → se pueden enlazar archivos ubicados en máquinas diferentes



Enlaces en Linux – Comando ln

Crea un enlace duro de un archivo (sólo podemos crearlo en una misma partición de disco)

**Al crear un enlace no generamos un nuevo archivo.
Agregamos un puntero hacia el nodo i de un archivo ya existente.**

Observando el árbol podemos decir que, al crear un enlace agregamos una nueva ruta hacia un archivo ya existente.

También podemos considerar a los enlaces como un mismo archivo con nombres diferentes.

Desde cualquiera de los enlaces podemos visualizar y modificar el contenido del archivo o sus permisos de acceso.

Comando ln - Sintaxis



```
$ ln ruta_direct/nombre_archivo ruta_direct_destino/nombre_enlace
```

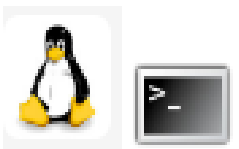
El enlace puede ser creado en el mismo directorio o en otro directorio.

Si es creado en el mismo directorio, entonces debe tener distinto nombre.

Si es creado en otro directorio, entonces puede tener el mismo nombre.

Si omitimos el nombre del enlace, es creado con el mismo nombre del archivo original.

Si omitimos el directorio destino, por defecto el enlace es creado en el directorio actual.



Comando In



¿Qué sucede cuando eliminamos uno de los enlaces?
Sólo borramos un puntero hacia el nodo i , es decir que solo borramos un acceso (una ruta) hacia el archivo.

¿De qué manera comprobamos que los enlaces son un mismo archivo?

Visualizando el número de nodo i .



```
clarisa@DESKTOP-D79LATI:~$ mkdir prueba
clarisa@DESKTOP-D79LATI:~$ cat >prueba/semana
martes jueves
miércoles sabado
lunes martes
viernes domingo
```

```
clarisa@DESKTOP-D79LATI:~$ ls -l prueba/semana
-rw-r--r-- 1 clarisa clarisa 61 Sep 29 20:14 prueba/semana
```

```
clarisa@DESKTOP-D79LATI:~$ ln prueba/semana semanalink1
```

```
clarisa@DESKTOP-D79LATI:~$ ls -l ./semanalink1
-rw-r--r-- 2 clarisa clarisa 61 Sep 29 20:14 ./semanalink1
```

```
clarisa@DESKTOP-D79LATI:~$ ln semanalink1 prueba/dir2/semanalink3
```

```
clarisa@DESKTOP-D79LATI:~$ ls -l prueba/dir2/semanalink3
-rw-r--r-- 3 clarisa clarisa 61 Sep 29 20:14 prueba/dir2/semanalink3
```

El archivo posee tres enlaces.





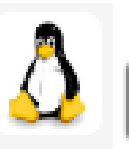
Se generaron tres enlaces hacia un mismo archivo, es por ello que tienen el mismo número de nodo-i:

```
clarisa@DESKTOP-D79LATI:~$ ls -li prueba/dir2/semanalink3
9288674232093514 prueba/dir2/semanalink3
clarisa@DESKTOP-D79LATI:~$ ls -li semanalink1
9288674232093514 semanalink1
clarisa@DESKTOP-D79LATI:~$ ls -li prueba/semana
9288674232093514 prueba/semana
```



Puede accederse al contenido del archivo a través de cualquiera de los tres enlaces:

```
clarisa@DESKTOP-D79LATI:~$ cat prueba/dir2/semanalink3
martes jueves
miércoles sabado
lunes martes
viernes domingo
clarisa@DESKTOP-D79LATI:~$ cat prueba/semana
martes jueves
miércoles sabado
lunes martes
viernes domingo
clarisa@DESKTOP-D79LATI:~$ cat semanalink1
martes jueves
miércoles sabado
lunes martes
```



Eliminar el contenido del archivo, para ello podrá accederse al mismo desde cualquiera de los 3 enlaces.

```
clarisa@DESKTOP-D79LATI:~$ vi prueba/semana
```

A través de cualquiera de los tres enlaces puede comprobarse que fue eliminado el contenido del archivo:

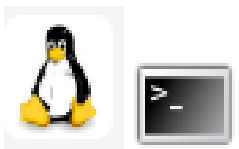
```
clarisa@DESKTOP-D79LATI:~$ cat prueba/semana
clarisa@DESKTOP-D79LATI:~$ cat semanalink1
clarisa@DESKTOP-D79LATI:~$ cat prueba/dir2/semanalink3
```



Al borrar un enlace se elimina una de las rutas de acceso al archivo, pero el archivo sigue existiendo:

```
clarisa@DESKTOP-D79LATI:~$ rm prueba/semana
clarisa@DESKTOP-D79LATI:~$ ls -l semanalink1
-rw-r--r-- 2 clarisa clarisa 0 Sep 29 20:42 semanalink1
clarisa@DESKTOP-D79LATI:~$ rm semanalink1
clarisa@DESKTOP-D79LATI:~$ ls -l prueba/dir2/semanalink3
-rw-r--r-- 1 clarisa clarisa 0 Sep 29 20:42 prueba/dir2/semanalink3
clarisa@DESKTOP-D79LATI:~$ rm prueba/dir2/semanalink3
```

El archivo queda eliminado cuando se borran todos los enlaces.





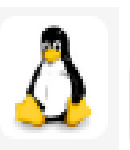
Comando ln -s

```
$ ln -s ruta_directorio/nombre de archivo ruta_directorio  
destino/nombre del enlace
```

Se utiliza para crear un enlace simbólico de un archivo ya existente.

Al crear un enlace simbólico no se genera una nueva entrada al nodo -i del archivo ya existente.

Al eliminar el archivo ya existente, el enlace simbólico queda apuntando “a la nada”.



¿Dudas o Inquietudes?