

Sistemas Operativos

Grado en Ingeniería Informática del Software

Tema 6: Gestión de Ficheros

1. El Sistema de Gestión de Ficheros
2. Ficheros
3. Directorios
4. El sistema de ficheros
5. Respaldo y recuperación

1.-El sistema de gestión de ficheros

Definición

- **Sistema de Gestión de ficheros, gestor de ficheros o servidor de ficheros:**
 - Parte del Sistema Operativo encargada del manejo y la organización de los ficheros. Administra la memoria secundaria y estructura el Sistema de Ficheros sobre ella.
 - Implementa los algoritmos y estructuras de datos necesarios para hacer corresponder la visión del usuario con el sistema físico de almacenamiento secundario

1.-El sistema de gestión de ficheros

Funciones

- Interfaz de **Llamadas al sistema**: Recibir llamadas al sistema sobre gestión de ficheros
- Gestionar **ficheros**: Localizar los bloques del disco que contienen los datos de un fichero, conectando la imagen lógica del fichero con la física del disco
- Gestionar **directorios**: localizar ficheros en la jerarquía de directorios y proporcionar sistemas de nombrado lógico de los ficheros
- Gestionar el **espacio libre** de los discos: Localizar bloques libres para asignarlos a nuevos ficheros o liberar bloques cuando se borra un fichero.
- Gestionar la **caché** de disco localizada en memoria
- Proporcionar **protección**: efectuar el control de acceso a los ficheros
- **Comunicarse** con el Sistema de **Gestión de E/S** (manejador de dispositivos) para indicarle la operación a realizar sobre los bloques físicos del disco

1. El Sistema de Gestión de Ficheros
2. Ficheros
3. Directorios
4. El Sistema de Ficheros
5. Respaldo y recuperación

Definición

- *Abstracción que utiliza el SO para almacenar datos de manera persistente. Agrupación de datos con un nombre*

Operaciones

- Crear
- Borrar
- Abrir
- Leer
- Escribir
- Cerrar
- Añadir
- Posicionarse
- Obtener y actualizar atributos
- Renombrar
- ...

Tipos

- En el pasado había muchos tipos distintos (secuencias de registros, árboles, ficheros directos, ...)
- Hoy son secuencias de bytes.

2.- Ficheros

Métodos de acceso

Método de acceso: Modo de acceder a la información de los ficheros.

- Secuencial
 - La información se procesa en orden secuencial
 - Las operaciones básicas se reducen a lectura de siguiente posición y escritura de siguiente posición
- Directo (por dirección)
 - Permite el acceso directo a cualquier dirección del mismo
 - Los registros pueden ser de cualquier tipo (incluidos los bytes)
- Directo (por clave) (no en sistemas de ficheros actuales)
 - Los archivos son secuencias de registros, y cada registro contiene un campo clave. Se pueden solicitar lecturas/escrituras sobre un registro con una determinada clave.
 - Pueden ser ficheros de acceso aleatorio o ficheros indexados.

2.- Ficheros

Características de un ficheros

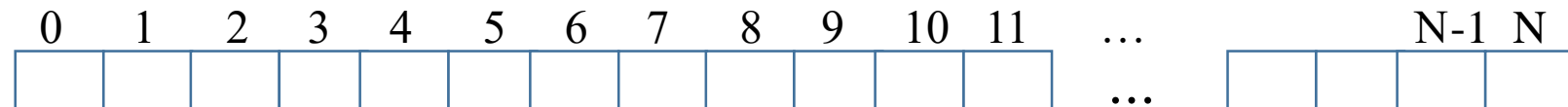
Elementos característicos de un fichero (atributos)

- **Nombre** (mecanismo de nombrado, nombres estándar)
- **Identificador** único (función de (nº mayor, menor de disp e inodo))
- **Tipo de fichero** (normales, directorio, dispositivo, ejecutables...)
- **Mapa de fichero** (localización en disco)
- **Protección** (lectura, escritura, ejecución...)
- **Tamaño**
- **Información de creación, modificación, etc** (fecha y hora)
- **Información de control** (archivo oculto, normal, directorio, etc)
- **Propietario** (usuario)
- El SO guarda esta información en un **Descriptor del Fichero**
 - **i-nodo** de Unix, **registro de MFT** de Windows NT, directorio de MS-DOS

2.- Ficheros

Estructura y almacenamiento

- El disco se organiza en bloques (bloques físicos del dispositivo)



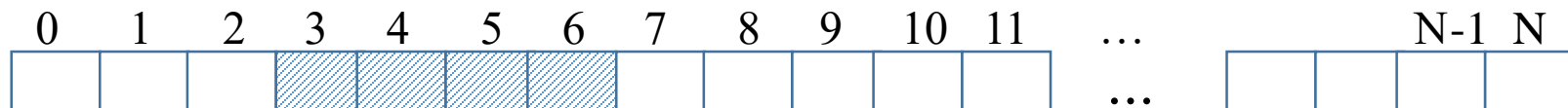
- Los ficheros se dividen en bloques (bloques lógicos del fichero)
- Dos posibilidades de asignación
 - Asignación contigua** (se guardan todos los bloques el fichero en bloques consecutivos del disco).
 - Asignación No contigua** (los bloques del fichero no están almacenados consecutivamente en disco). Ej: discos duros. Habrá que utilizar una estructura que relacione bloque lógico-físico.

2.- Ficheros

Estructura y almacenamiento: Asignación contigua

- Para cada fichero, hará falta almacenar en su descriptor:
 - Bloque de inicio del fichero.
 - Número de bloques que ocupa.
- En caso de aumento del tamaño del fichero, si no hay espacio libre a continuación del que tiene asignado no podrá crecer.
- En un sistema con alto grado de volatilidad (ficheros que se crean y eliminan a menudo) provocará fragmentación, con lo que habrá que compactar periódicamente.
- Utilizado solamente en algunos dispositivos donde los ficheros se graban al crearse y ya no se modifican (cintas, CD-ROM, ...)

Fichero: bloques 3, 4, 5, 6. En el descriptor: 3



2.- Ficheros

Estructura y almacenamiento: Asignación no contigua

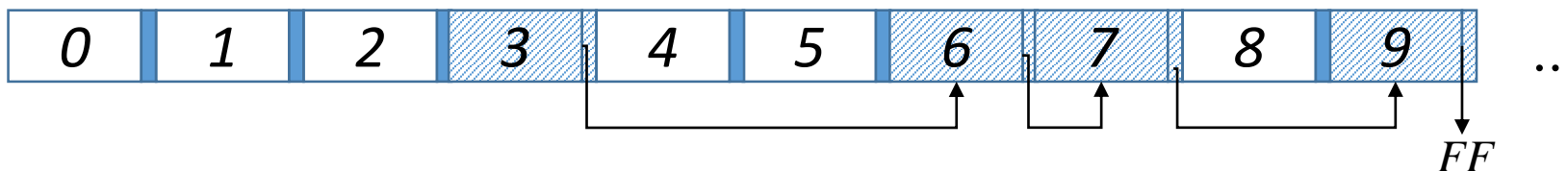
- Los bloques de cada fichero no tienen que estar almacenados en posiciones contiguas.
- Aunque la fragmentación no es desde el punto de vista lógico un problema, sí puede serlo en cuanto al rendimiento.
- Para localizar los bloques que pertenecen a cada fichero puede utilizarse se utiliza una lista con los bloques que pertenecen al fichero. Esta lista puede implementarse:
 1. Asignación con lista enlazada en los propios bloques de datos
 2. Asignación con FAT
 3. Asignación indexada con tabla de índices
 4. Asignación indexada con árbol equilibrado

Estructura y almacenamiento: Asignación no contigua

1. Asignación enlazada

- Dentro de cada bloque de datos existe un puntero al siguiente bloque de datos.
- En el descriptor se almacena la dirección del primer bloque del fichero.
- Hay que recorrer los bloques anteriores para llegar a uno dado.
- El puntero ocupa un espacio y está incluido en el fichero.

Fichero: bloques 3, 6, 7, 9. En el descriptor: 3



2. Asignación con FAT (*File Allocation Table*)

- El sistema mantiene en el disco una tabla con una entrada para cada bloque del disco.
- En el descriptor se almacena la dirección del primer bloque del fichero.
- En la entrada de la FAT correspondiente al primer bloque del fichero se almacena la dirección del segundo. En la entrada correspondiente a la del segundo, la del tercero, ...
- En la entrada correspondiente al último bloque del fichero se almacena una marca de fin de fichero.
- Para acceder directamente a un bloque hay que hacer un recorrido por la FAT, no por los bloques.
- FAT16 utiliza 16 bits para cada entrada de la FAT; FAT 32 usa 32.
- Usado en MS-DOS y hoy en día en dispositivos extraíbles (USBs, tarjetas SD, ...) puesto que hay circuitería desarrollada para soportarlo (reproductores MP-3, autoradios, ...)

2.- Ficheros

Estructura y almacenamiento: Asignación no contigua

Fichero: bloques 3, 6, 7, 9. En el descriptor: 3

0	1	2	3	4	5	6	7	8	9	...
---	---	---	---	---	---	---	---	---	---	-----

Bloque	Siguiente
1	
2	
3	6
4	
5	
6	7
8	

Bloque	Siguiente
9	FF
10	
11	
12	
13	
14	
15	

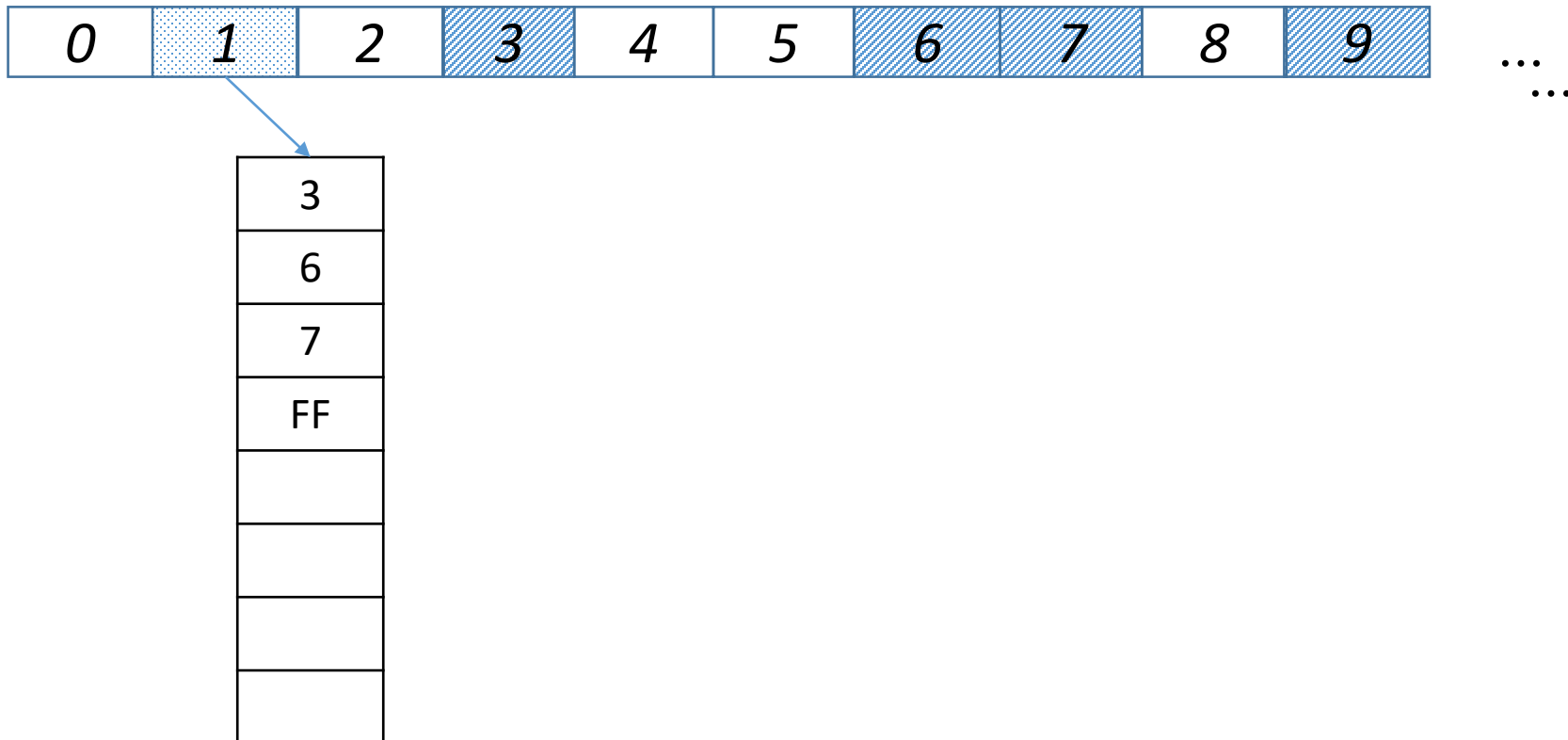
3. Asignación indexada con tabla de índices

- Se guardan todos los números de bloque (índices) de un fichero en una estructura llamada *tabla de índices*. Esta tabla se almacena en bloques de disco.
- En la i-ésima entrada se localiza el índice del i-ésimo bloque del fichero.
- Se organizan los bloques de índices de alguna manera (lista enlazada, árbol, ...)
- Existen en el disco dos tipos de bloques
 - De datos
 - De índices (bloques con números bloques físicos del fichero)
- En cada bloque de índices caben muchos índices de bloques
 - Tamaño de bloque / nº de bytes del índice
 - Ej: $4K/4Bytes = 1024$ índices caben en un bloque de 4K

2.- Ficheros

Estructura y almacenamiento: Asignación no contigua

*Fichero: bloques 3, 6, 7, 9. En el descriptor: **1***



2.- Ficheros

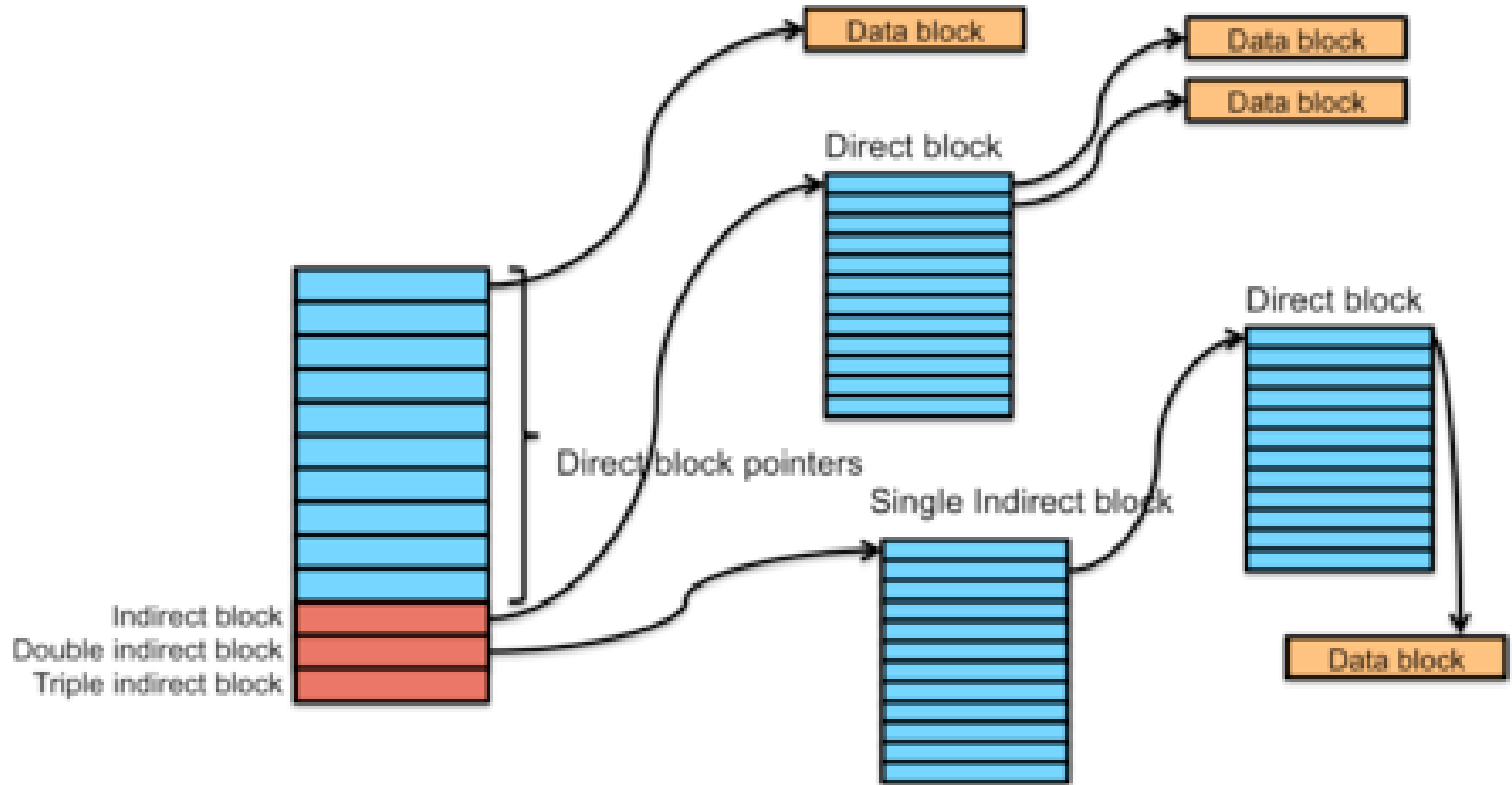
Estructura y almacenamiento: Asignación no contigua

Ejemplo: Sistemas de ficheros de Unix System V, FFS, ext2...

- Cada fichero tiene una tabla de 13 entradas en el i-nodo
- Las 10 primeras entradas (de la 0 a la 9) contienen los 10 primeros índices a los 10 primeros bloques de datos
- La entrada 10 contiene un índice a un bloque de índices. Este bloque de índices contiene índices a los siguientes bloques de datos del fichero (al 10, al 11, ...)
- La entrada 11 contiene una doble indirección: un índice a un bloque de índices que contiene índices a bloques de índices que contienen índices a bloques de datos del fichero
- La entrada 12 contiene una triple indirección: un índice a un bloque de índices que contiene índices a bloques de índices que contienen índices a bloques de índices que contienen índices a bloques de datos

2.- Ficheros

Estructura y almacenamiento: Asignación no contigua

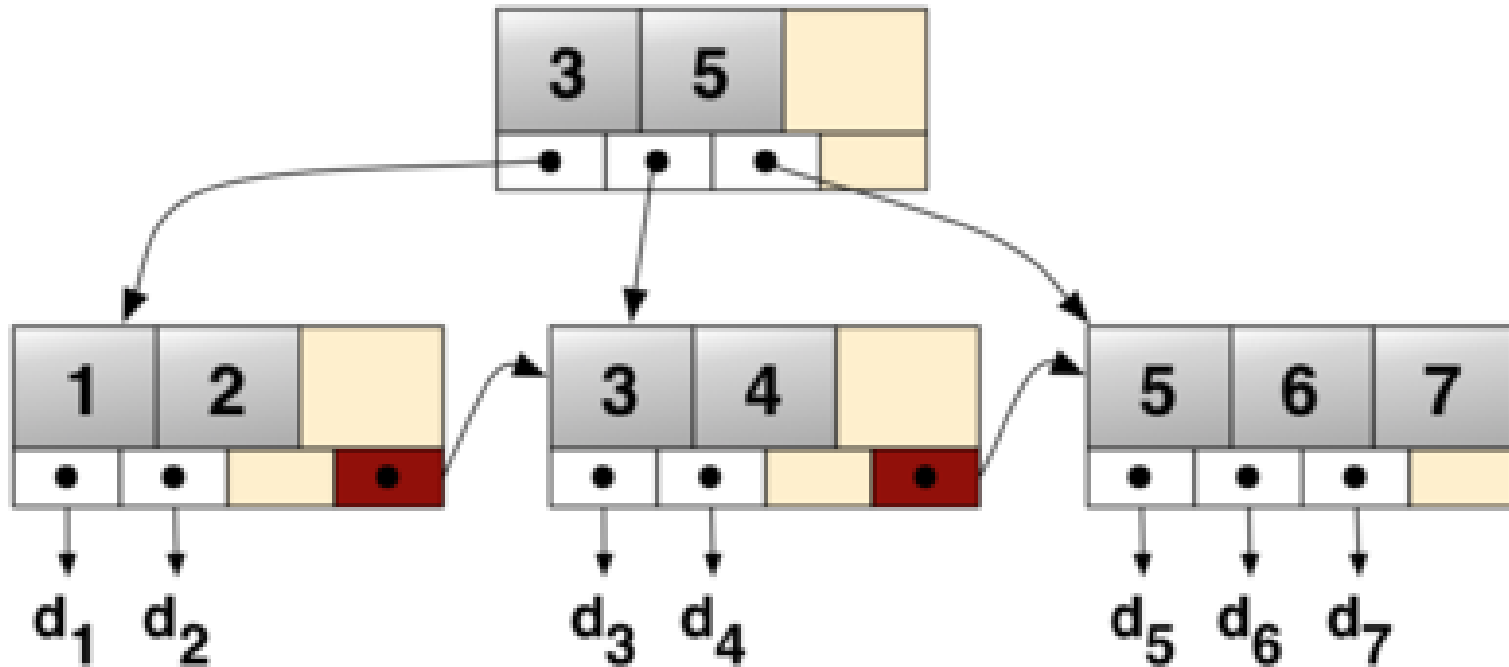


4 Asignación indexada con Árboles equilibrados

- Representa la estructura de un fichero mediante árboles equilibrados (con número fijo de hijos en cada nivel).
- Cada bloque contiene datos y punteros a los hijos.
- La raíz del árbol se almacena en el descriptor del fichero.
- En cada nodo puede haber datos y enlaces a otros nodos.
- La clave de búsqueda es el número lógico del bloque.
- Usado en NTFS, donde sólo el descriptor y los nodos hoja contienen datos, los intermedios contienen índices a otros.
- Los accesos directos por dirección pueden ser muy rápidos.

2.- Ficheros

Estructura y almacenamiento: Asignación no contigua



Ejemplo de árbol B

2.- Ficheros

Estructura y almacenamiento: Gestión del espacio libre

Además de gestionar el espacio ocupado el SO tiene que saber qué bloques del disco están libres, para poder asignarlos a ficheros cuando sea necesario. Hay varias posibilidades:

- Usar un mapa de bits, con una entrada para cada bloque.
- En los sistemas con FAT cada entrada de la propia FAT se utiliza para indicar si el bloque está libre.
- Usar una lista conteniendo los bloques libres. Esta lista puede estar implementada de manera similar a lo visto para los ficheros (lista enlazada, lista enlazada de bloques de índices, ...)

1. El Sistema de Gestión de Ficheros
2. Ficheros
3. Directorios
4. El Sistema de Ficheros
5. Respaldo y recuperación

Concepto

Desde el punto de vista del usuario

Contenedor de ficheros

Desde el punto de vista del sistema operativo

Objeto que relaciona el nombre simbólico de un fichero con el descriptor interno usado por el S.O.

Estructura de los directorios

- De uno o dos niveles
- De árbol
 - Nodo raíz, nodos internos son directorios y hojas son ficheros
- De grafos acíclicos
 - Útil para compartición de directorios por usuarios diferentes
 - Se crean enlaces a archivos compartidos

Denominación

- El nombre del fichero lo marca el camino desde el directorio raíz hasta el fichero.
- Obtención del nombre
 - Nombre absoluto- Desde el directorio raíz
 - Nombre relativo- Desde algún otro subdirectorio de la jerarquía

Operaciones genéricas sobre directorios

- Crear
- Borrar
- Abrir
- Cerrar
- Leer
- Renombrar
- Cambiar directorio actual
- Enlazar

Estructura y almacenamiento de los directorios

1. En **ficheros contiguos**, almacenan atributos y número del primer bloque y tamaño.
2. En **ficheros FAT**, almacenan atributos, número del primer bloque y tamaño.
3. En **ficheros indexados**, identificador del descriptor del fichero.
 - En Unix System V nombre de fichero y número de i-nodo.
 - Los atributos van en el i-nodo localizado en otra zona del disco
 - Los i-nodos pueden ser de ficheros o de directorios
 - En Windows NTFS en el directorio están algunos atributos y el número del i-nodo (registro en la MFT) que es el nodo raíz del árbol equilibrado

1. El Sistema de Gestión de Ficheros
2. Ficheros
3. Directorios
4. El Sistema de Ficheros
 1. Estructura general del Sistema de Ficheros en disco
 2. Algunos ejemplos
5. Respaldo y recuperación

4.-El sistema de Ficheros

Estructura

- Los discos se dividen en una o varias particiones o volúmenes **Partición:** *porción del disco a la que se le dota de entidad propia y puede ser manipulada por el SO como entidad independiente*
- Algunos SO permiten construir particiones extendidas que abarcan varios discos
- **Sistema de ficheros:** Modo de organización de la información en el disco, en un formato inteligible para el sistema operativo
- Para la definición del sistema de ficheros, el S.O proporciona órdenes como *format* o *mkfs*:
 - Ej: *Mkfs -c /dev/hda3 -b 8192 123100*
8192 es el tamaño del bloque que se define (8 K)
123100 es el número de bloques que tendrá el sistema de ficheros

4.-El sistema de Ficheros

Estructura

- El bloque es la **unidad de transferencia** mínima (1 o más sectores)
- El Sistema de ficheros (partición) se divide en bloques **numerados** lógicamente del 0 al Máximo del Sistema de Ficheros. El tamaño del bloque suele poder definirse al formatear.
- Los ficheros también se dividen en bloques (de igual tamaño) numerados lógicamente del 0 al Máximo del Fichero
- **A cada bloque lógico del fichero se le asigna un bloque lógico del Sistema de Ficheros** (denominado en ocasiones bloque físico)
- El tamaño del bloque influye en :
 - **Uso del disco.** Cada fichero tiene un número entero de bloques
 - **Velocidad de acceso** (búsqueda y transferencia de bloques)
 - Tamaños grandes producen fragmentación interna, pero son más rápidos
 - Tamaños típicos (512, 1K, 2K)

4.-El sistema de Ficheros

Estructura

Información almacenada en los bloques del disco (en general)

1. Información de arranque del S.O. (boot)

- Contiene código del sistema operativo para instalación del mismo
- Sólo contiene información en los discos de arranque

2. Metainformación sobre Sistema de Ficheros

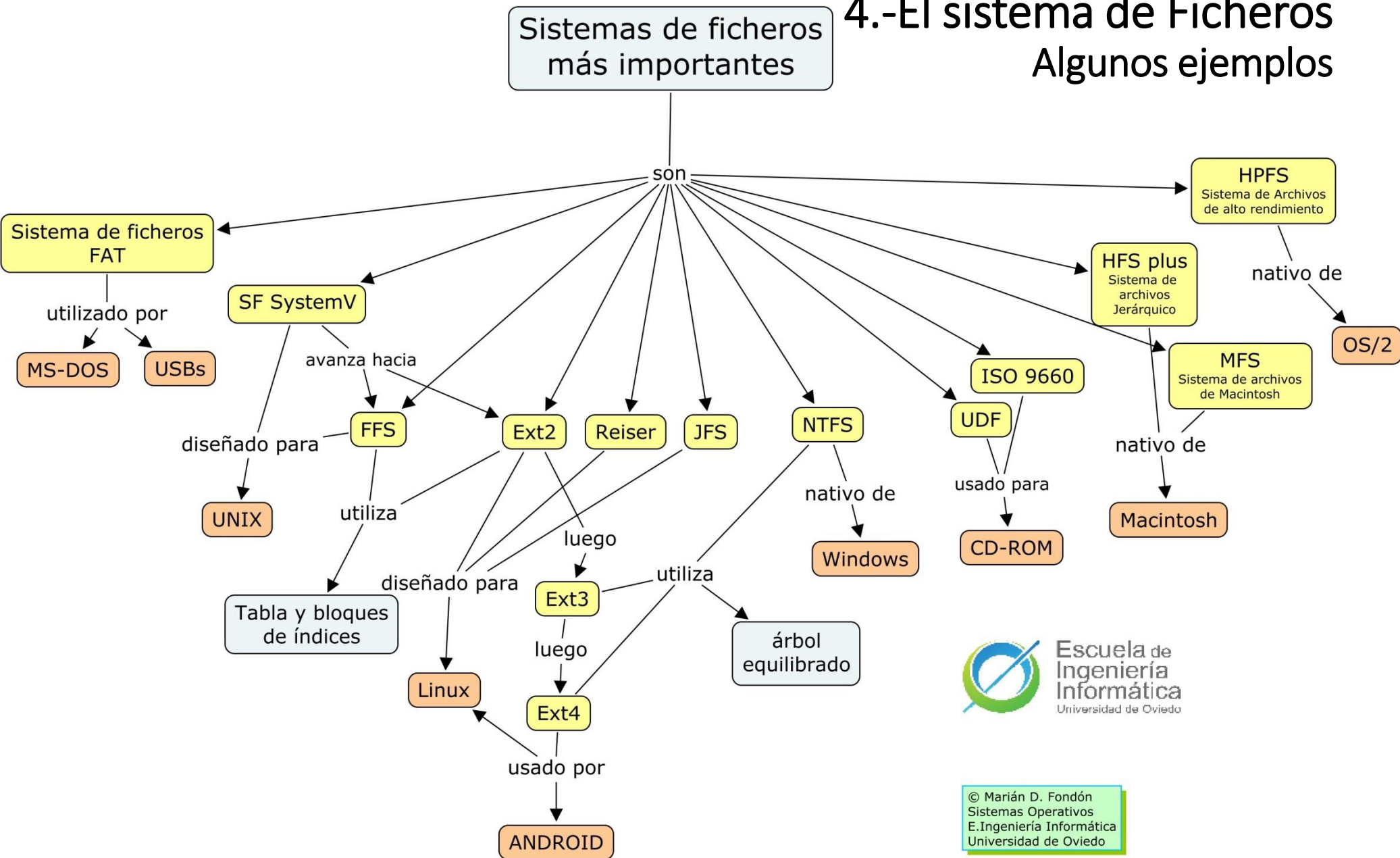
Descripción del sistema de archivos

- Tamaño de bloque, número máximo de ficheros, etc
- Información de estado de ocupación de los bloques de datos
- Descriptores de ficheros

3. Bloques de datos

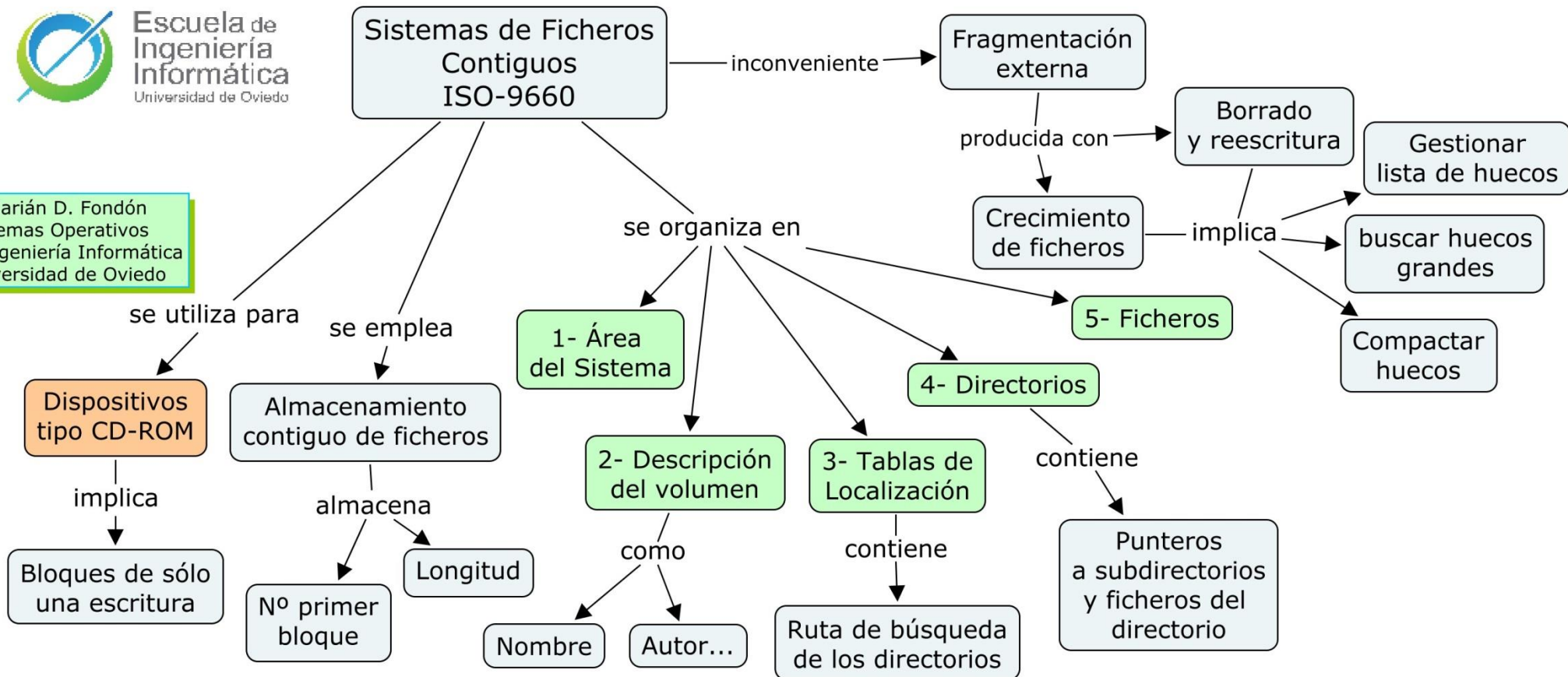
- Para guardar datos de los ficheros

4.-El sistema de Ficheros Algunos ejemplos



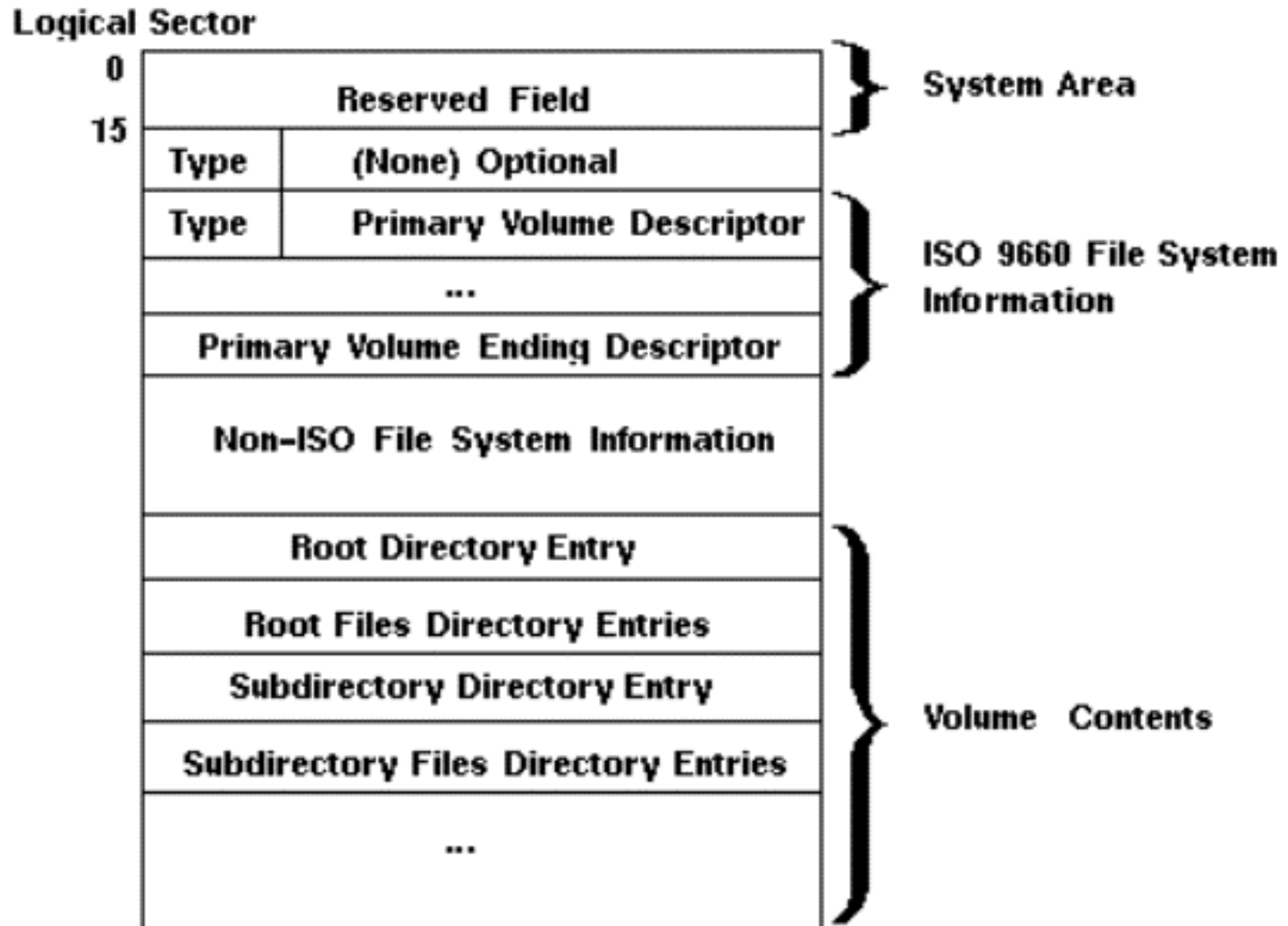
4.-El sistema de Ficheros

Algunos ejemplos: ISO-9660



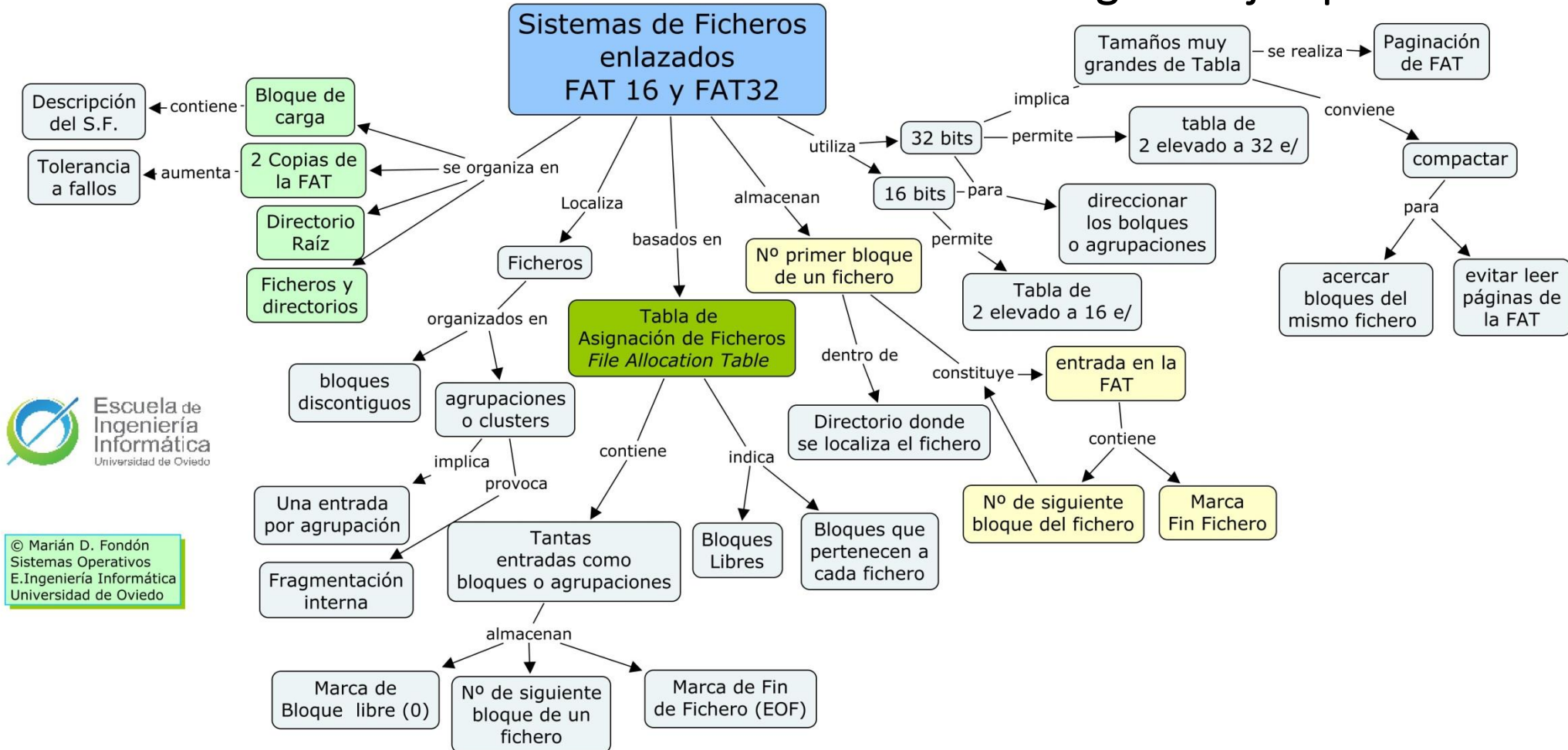
4.-El sistema de Ficheros

Algunos ejemplos: ISO-9660



4.-El sistema de Ficheros

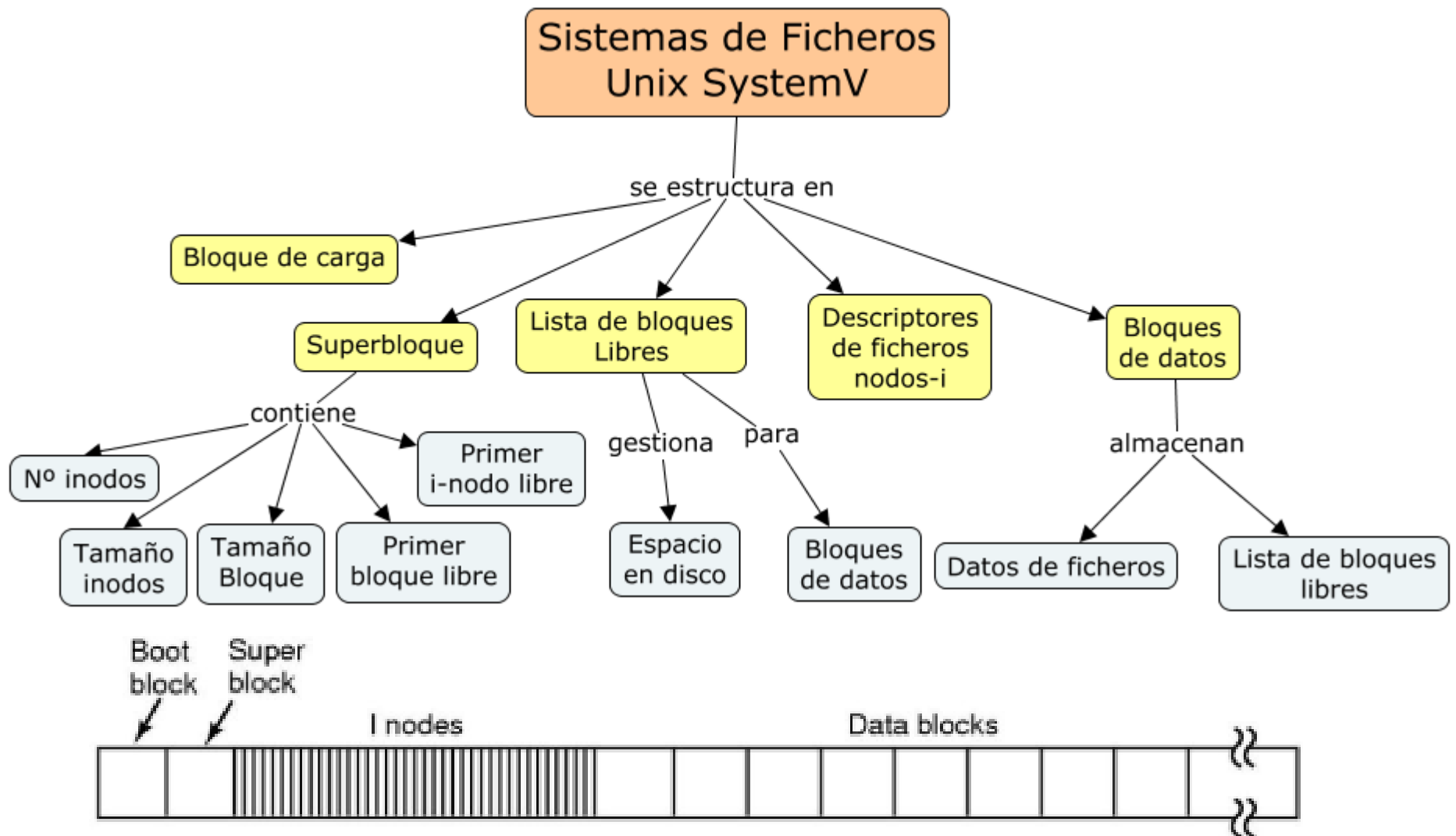
Algunos ejemplos: FAT



Boot Sector	FAT	FAT	Directory Entries	...	file	...	file	Directory Entries	...
-------------	-----	-----	-------------------	-----	------	-----	------	-------------------	-----

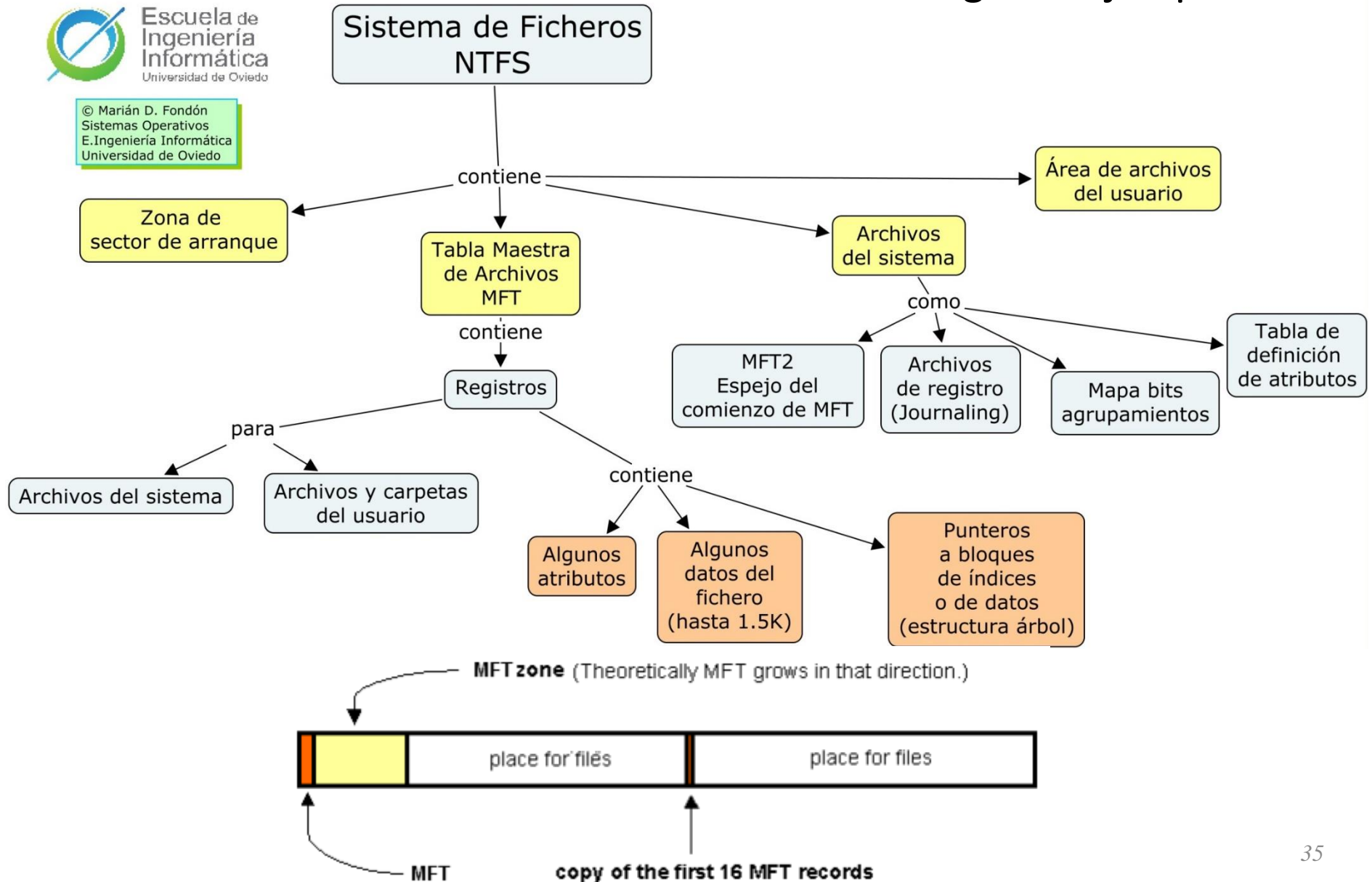
4.-El sistema de Ficheros

Algunos ejemplos: Unix System V



4.-El sistema de Ficheros

Algunos ejemplos: NTFS



4.-El sistema de Ficheros

Estructura

Problemas de los sistemas de ficheros anteriores

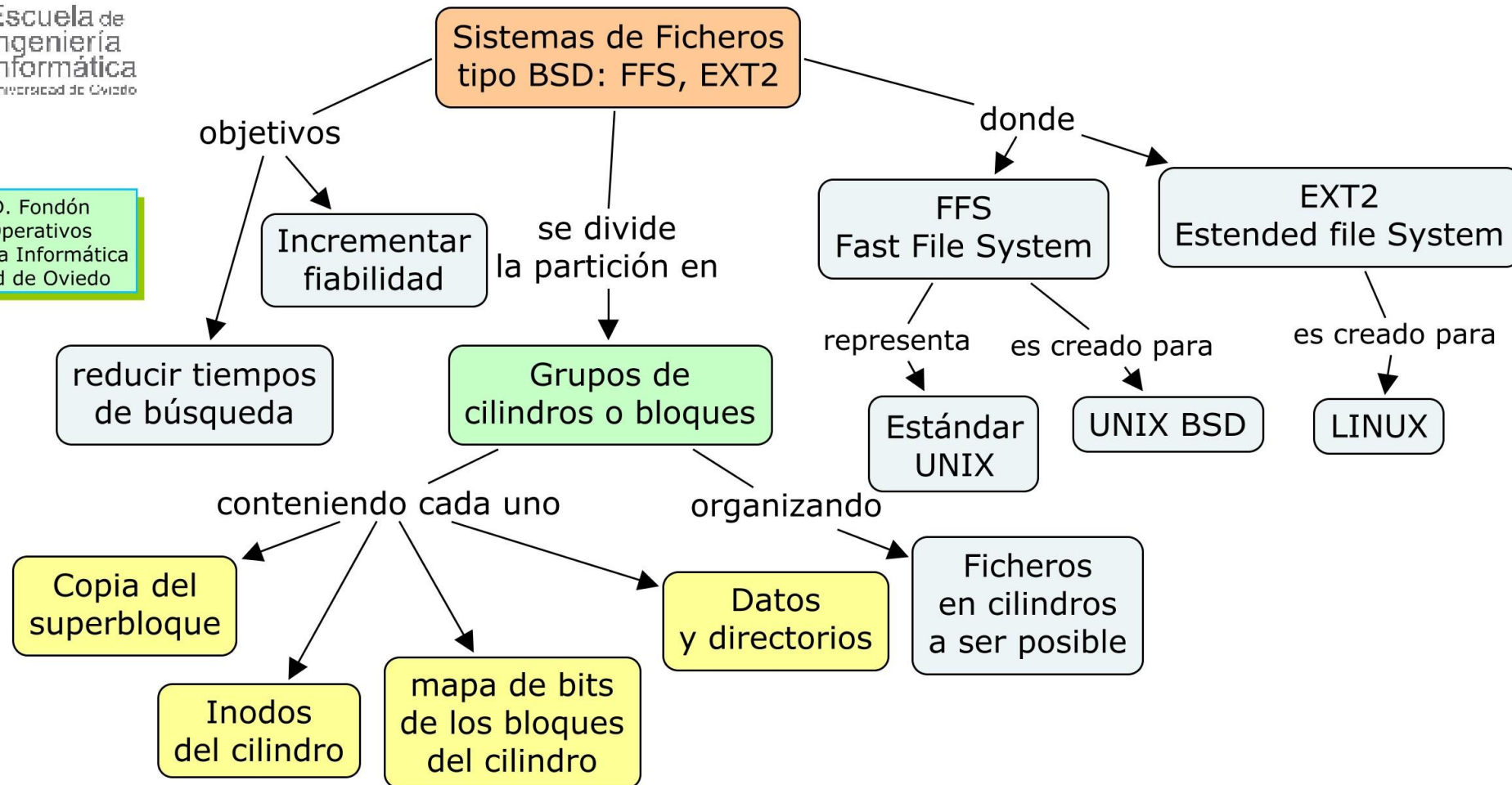
- La metainformación está agrupada al principio del disco
- El **tiempo de búsqueda es grande** porque los bloques están dispersos por el disco
- **Si se corrompe** la metainformación, el sistema se inutiliza

Para resolverlo aparecen

- FFS (Fast File System) de Unix BSD y
- EXT2 (extended file system) de Linux

4.-El sistema de Ficheros

Algunos ejemplos



4.-El sistema de Ficheros

Sistemas de ficheros con Journaling

- Diseñados para resolver el **problema de la recuperación de la consistencia** del SF si se producen fallos.
- Se añade al modelo de Sistema de Ficheros tipo FFS o ext2 un **registro** en cada grupo de bloques
- La idea principal es guardar todas las operaciones de escritura de forma secuencial en un fichero de registro (log o journal) intermedio del disco y posteriormente escribirlas en cualquier otro formato.
- **Se aplican técnicas transaccionales**
- Los datos se almacenan consecutivamente en el registro
- No se reescriben los datos (si se modifica dos veces un directorio se guarda 2 veces)

Ejemplos de sistemas de ficheros

JFS de IBM, ext3 de Linux, Reiser FS de Namesys, XFS de Silicon Graphics

1. El sistema de ficheros
2. Ficheros
3. Directorios
4. El Sistema de Gestión de Ficheros
5. Respaldo y recuperación

5.-Respaldo y recuperación

Importancia de la seguridad ante problemas de pérdida de datos

- Daño fortuito (rayos, incendios, fallos Hw)
- Daño intencionado (virus)
- Errores (Borrado accidental)

Soluciones

- Copias de respaldo (backups)
- Almacenamiento redundante (dispositivos RAID)

5.-Respaldo y recuperación

Copias de respaldo

- Consiste en copiar los datos en otro dispositivo.
- Puede hacerse de varias maneras. Las más habituales son:
 - 1. Respaldo total**
Copiar periódicamente los ficheros a otros dispositivos
 - 2. Respaldo incremental**
Sólo se copian los datos modificados desde la última copia

Estas dos soluciones tienen inconvenientes

- Pueden llevar mucho tiempo, pueden obligar a parar el sistema

5.-Respaldo y recuperación

Dispositivos RAID

Los dispositivos RAID (*Redundant Array of Independent Disks*) utilizan varios discos para mejorar el rendimiento y aumentar la seguridad

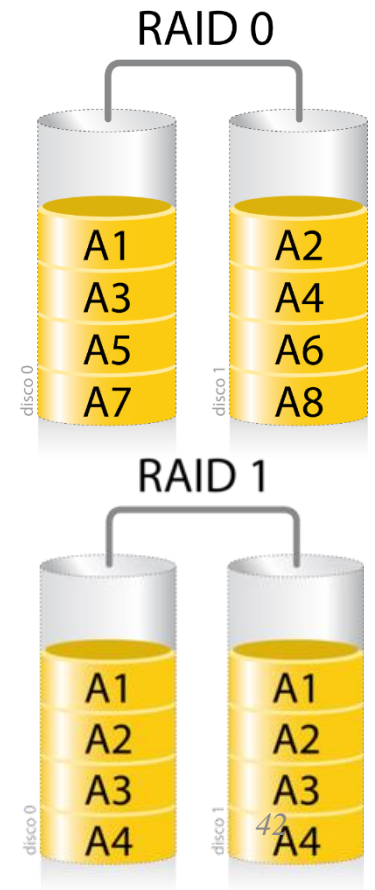
Hay varios niveles del RAID (desde 0 a 7, más múltiples niveles anidados).

RAID 0 (Data Striping). Reparte en 2 ó más discos el almacenamiento, apareciendo como una única unidad lógica.

- Aumenta el rendimiento.
- No proporciona redundancia

RAID 1 (Mirroring). Copia los datos en 2 ó más discos, tratándose como una única unidad.

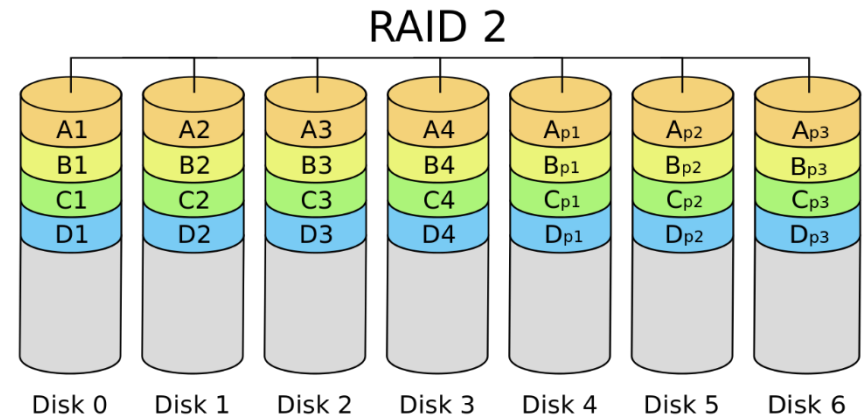
- No aumenta el rendimiento.
- Proporciona redundancia



5.-Respaldo y recuperación Dispositivos RAID

RAID 3. Distribuye los datos entre varios discos, a nivel de bit. Incluye códigos de paridad y de recuperación de errores.

- Aumenta el rendimiento.
- Proporciona redundancia
(No usado en la actualidad)



RAID 5. Distribuye los datos entre varios discos, a nivel de bloque. Incluye códigos de paridad y de recuperación de errores.

- Aumenta el rendimiento.
- Proporciona redundancia

