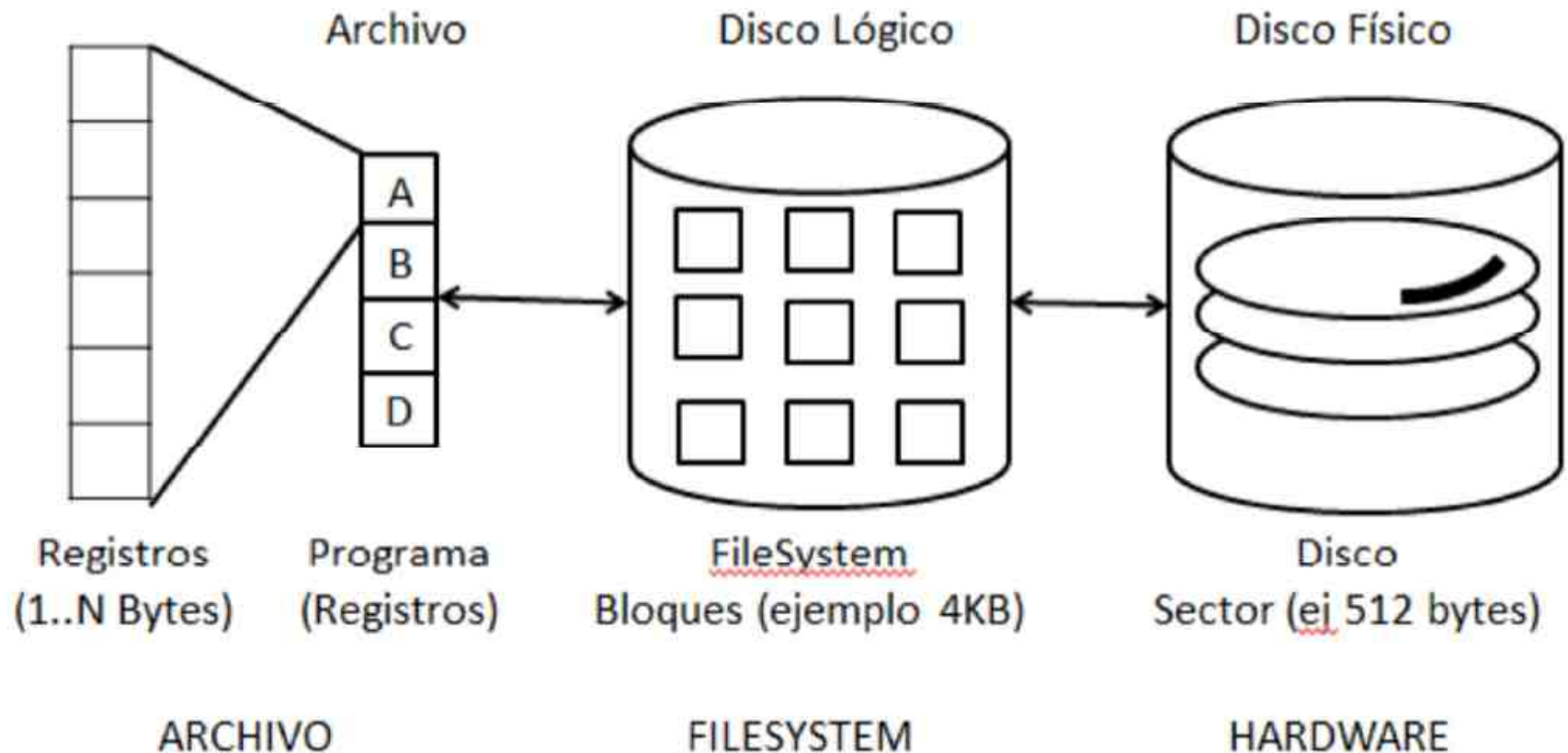


File System

FAT / UFS (Ext2)



Repaso clase anterior: File System



Archivos / Bloques

Vista del usuario main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>

void mifuncion(int *numero);

int VARIABLE_GLOBAL;
int main(void){
    int i;
    int hilos = 10;
    VARIABLE_GLOBAL = 0;
    pthread_t tid[hilos][2];
    for (i = 0; i < hilos; i++){
        tid[i][1] = i+1;
        pthread_create(&tid[i][0], NULL, (void*)
mifuncion, &tid[i][1]);
    }
    // Espero finalizacion de todos los hilos para
continuar
    for (i = 0; i < hilos; i++){
        pthread_join(tid[i][0], NULL);
    }
    return EXIT_SUCCESS;
}
```

Vista del FileSystem main.c

1er bloque

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>

void mifuncion(int *numero);

int VARIABLE
```

2do bloque

```
LE_GLOBAL;
int main(void){
    int i;
    int hilos = 10;
    VARIABLE_GLOBAL =
0;
    pthread_t tid[hilos][2];
    for (i = 0; i < hilo
```

3er bloque

```
s; i++){
        tid[i][1] = i+1;

pthread_create(&tid[i][0],
NULL, (void*) mifuncion,
&tid[i][1]);
    }
    // Espero finalizacion
de todos los hilo
```

4to bloque

```
s para continuar
    for (i = 0; i < hilos; i++){
pthread_join(tid[i][0], NULL);
    }
    return EXIT_SUCCESS;
}00000000000000000000
```

Ejemplo de escritura de un archivo nuevo

1) Crear el archivo:

- Verificar que exista FCB disponible.
- Crear Entrada de Directorio.

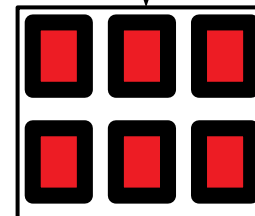
FCB: 405	/home :		
Atributos			
Ptr a bloques			
...			
	nombre	Tipo	FCB
	hola.txt	regular	402
	nuevo.txt	regular	405

3) Escribir en el archivo:

- Escribir los bloques.
- Actualizar atributos (fechas, tamaño, permisos).

2) Asignarle bloques requeridos:

- Abrir el archivo y agregarlo en las listas de archivos abiertos.
- Obtener bloques libres.
- Asignar los bloques al archivo.



4) Cerrar el archivo:

- Actualizar lista de archivos abiertos

FAT (File Allocation Table)

- Desarrollado en 1977 utilizado en diskettes.
- Actualmente se utiliza en algunas memorias Flash.
- Versiones 12 / 16 / 32 / VFAT / ExFat
- DOS / Windows

FAT (File Allocation Table)

- **Partición**



FAT (File Allocation Table)

- Instalación de Volumen (Tipo FAT) en Partición.



FAT (File Allocation Table)

- **Directorios:** contiene un listado de archivos y directorios que “contiene”. Entradas de Directorio:

Tipo de archivo	Nombre y extensión	Primer Cluster Archivo	Tamaño	...
-----------------	--------------------	------------------------	--------	-----

FAT (File Allocation Table)

- **Directorios:** contiene un listado de archivos y directorios que “contiene”. Entradas de Directorio:

Tipo de archivo	Nombre y extensión	Primer Cluster Archivo	Tamaño	...
-----------------	--------------------	------------------------	--------	-----

Cluster con
archivo tipo
directorio

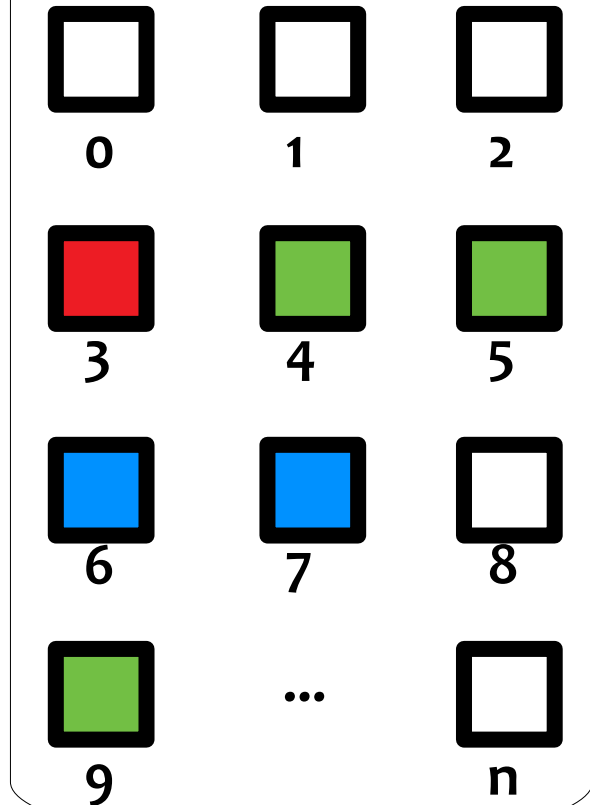
a	arc.txt	7	2000b
a	tp.c	5	3000b
d	utnso	3	ob

FAT (File Allocation Table)

A	arc.txt	7	2000 bytes	...
A	tp.c	5	3000 bytes	...
D	utnso	3	0 bytes	...

0	libre
1	libre
2	libre
3	fin
4	9
5	4
6	fin
7	6
8	error
9	fin
	...
n	libre

Tamaño de Cluster: 1 KiB



FAT (File Allocation Table)

FAT12: Punteros de 12 bits.

FAT16: Punteros de 16 bits.

FAT32: Punteros de 32 bits.

- **Sólo se usan 28 bits para el puntero.**

FAT	
0	libre
1	libre
2	libre
3	fin
4	9
5	4
6	fin
7	6
8	error
9	fin
	...
n	libre

FAT (File Allocation Table)

Ejemplo FAT32:

- Punteros de 32 bits.
- Tamaño de Cluster 2 KiB.
- Tamaño máximo teórico del filesystem:
 - $2^{28} * 2^{11} = 2^{39} = 512 \text{ GiB}$
- Tamaño Máximo Real (Disco de 500GiB)
 - $512 \text{ GiB} > 500 \text{ GiB}$ entonces es 500 GiB
- Tamaño Máximo Real (Disco de 1000GiB)
 - $512 \text{ GiB} < 1000 \text{ GiB}$ entonces es 512 GiB

0	libre
1	libre
2	libre
3	fin
4	9
5	4
6	fin
7	6
8	error
9	fin
	...
n	libre

UFS (EXT₂ / EXT₃)

- Desarrollado en 1992.
- Actualmente se utiliza en sistema tipo Linux / MAC
- EXTFS / EXT₂ / EXT₃ / EXT₄ (2008)

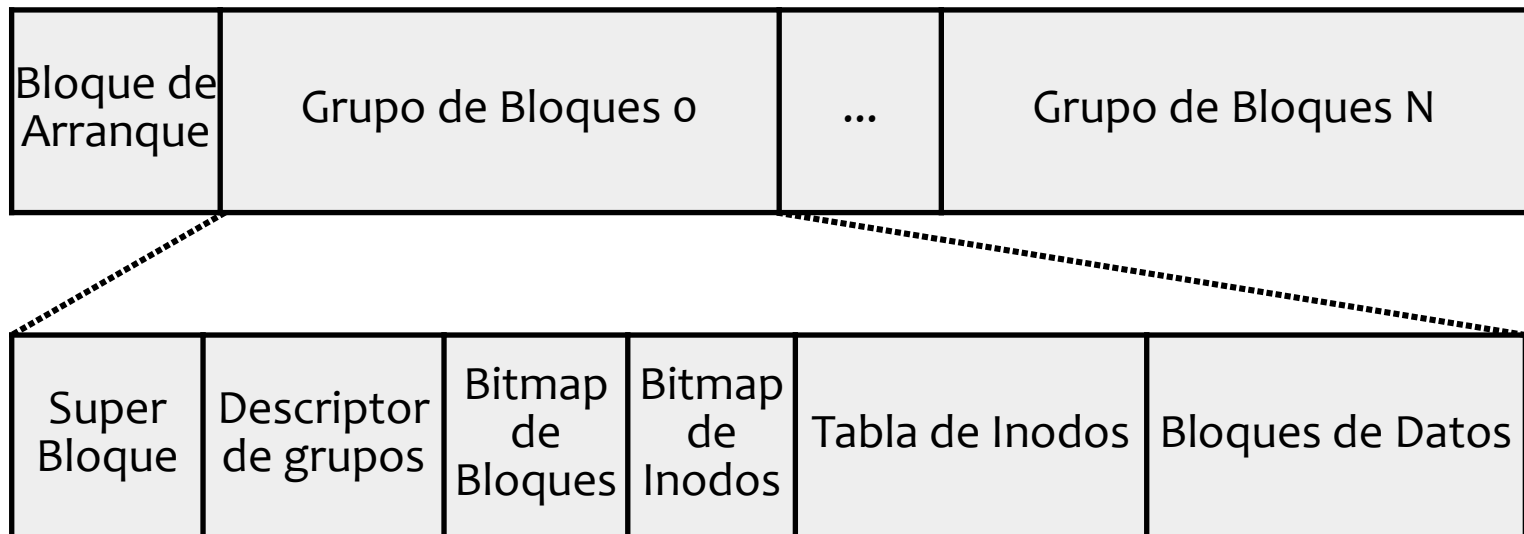
UFS (EXT₂ / EXT₃)

- **Partición**



UFS (EXT2 / EXT3)

- Instalación de Volumen (Tipo UFS) en Partición.



UFS (EXT₂ / EXT₃)

- **Directorios:** contiene un listado de archivos y directorios que “contiene”. Entradas de Directorio:

Número de Inodo	Nombre del archivo	Tipo de Archivo
-----------------	--------------------	-----------------

Bloque de datos
de archivo tipo
directorio

10	arc.txt	archivo
20	tp.c	archivo
25	utnso	directorio

UFS (EXT2 / EXT3)

▪ Inodos:

```

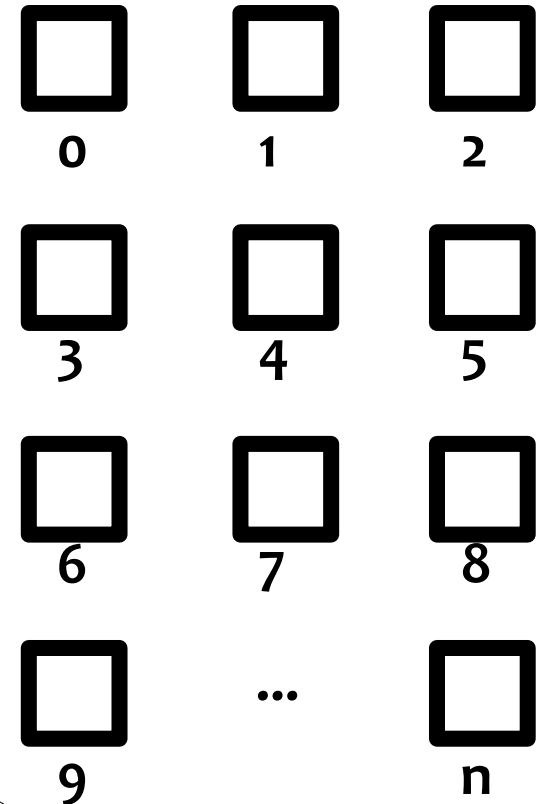
10 arc.txt  archivo
20 tp.c    archivo
25 utnso   directorio

```

inodo 20

	id
	propietario
	grupo
	permisos
	tamaño
	...
1	puntero
2	puntero
3	puntero
4	puntero
	...
p	puntero

Tamaño de Cluster: 1 KiB



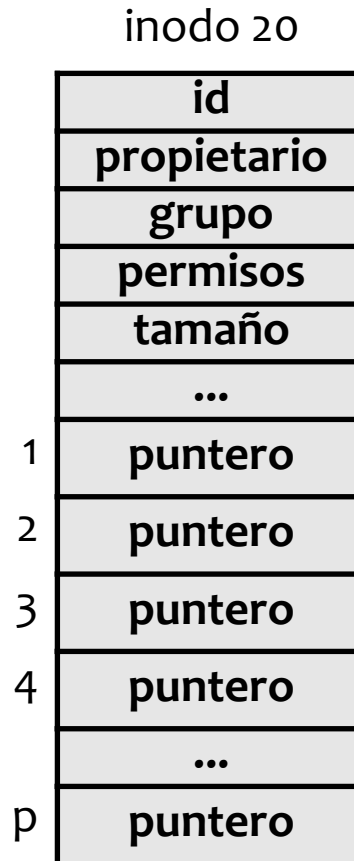
UFS (EXT₂ / EXT₃)

- **Inodos:**
 - Es el FCB del file system.
 - Hay uno por cada archivo.
 - Contiene los atributos del archivo.
 - Punteros que permiten acceder a los datos.
 - Tamaño fijo de 128 bytes.

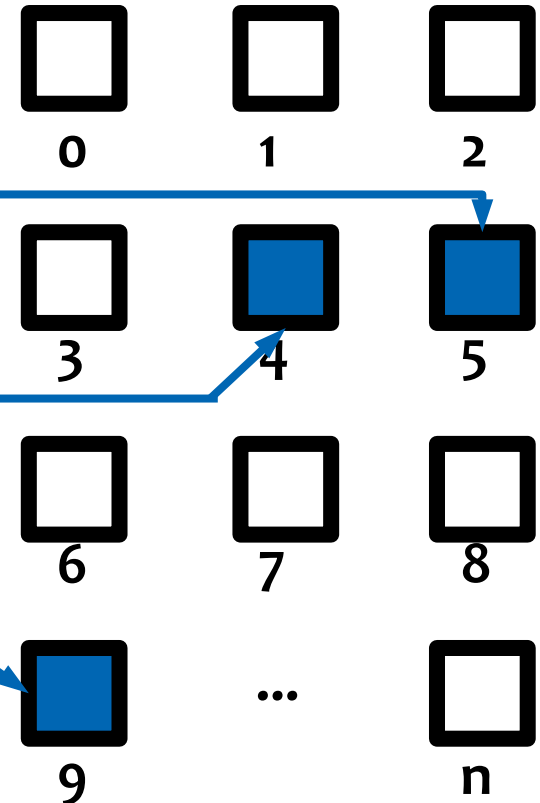
UFS (EXT2 / EXT3)

▪ Inodos:

```
10 arc.txt  archivo
20 tp.c    archivo
25 utnso   directorio
```



Tamaño de Cluster: 1 KiB

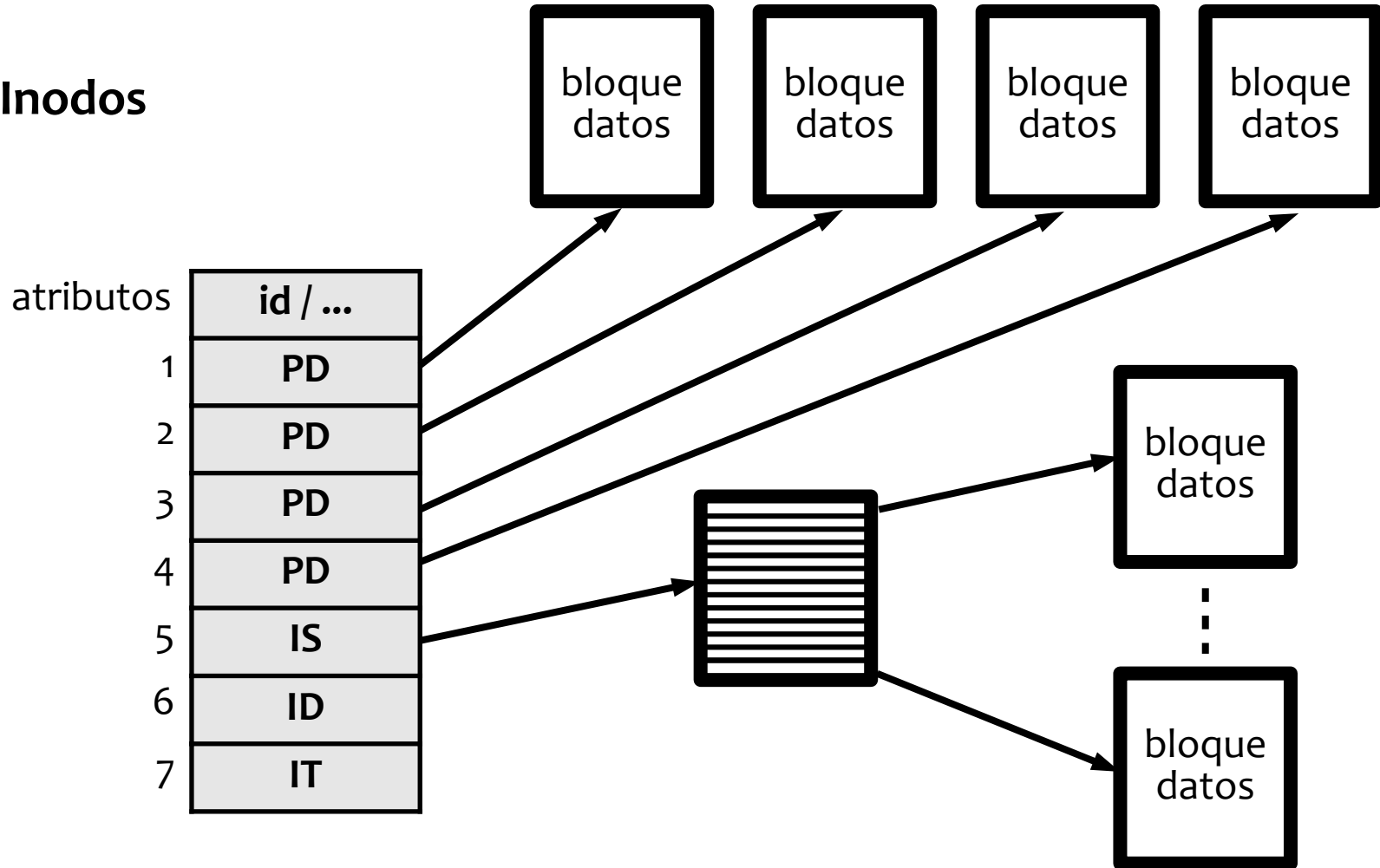


UFS (EXT₂ / EXT₃)

- **Inodos:**
 - Se crean punteros a demanda utilizando bloques de datos
 - Tipos de Punteros en el inodo:
 - Punteros Directos: apuntan a un bloque de datos.
 - Punteros Indirectos: apuntan a un bloque de punteros
 - Simples
 - Dobles
 - Triples

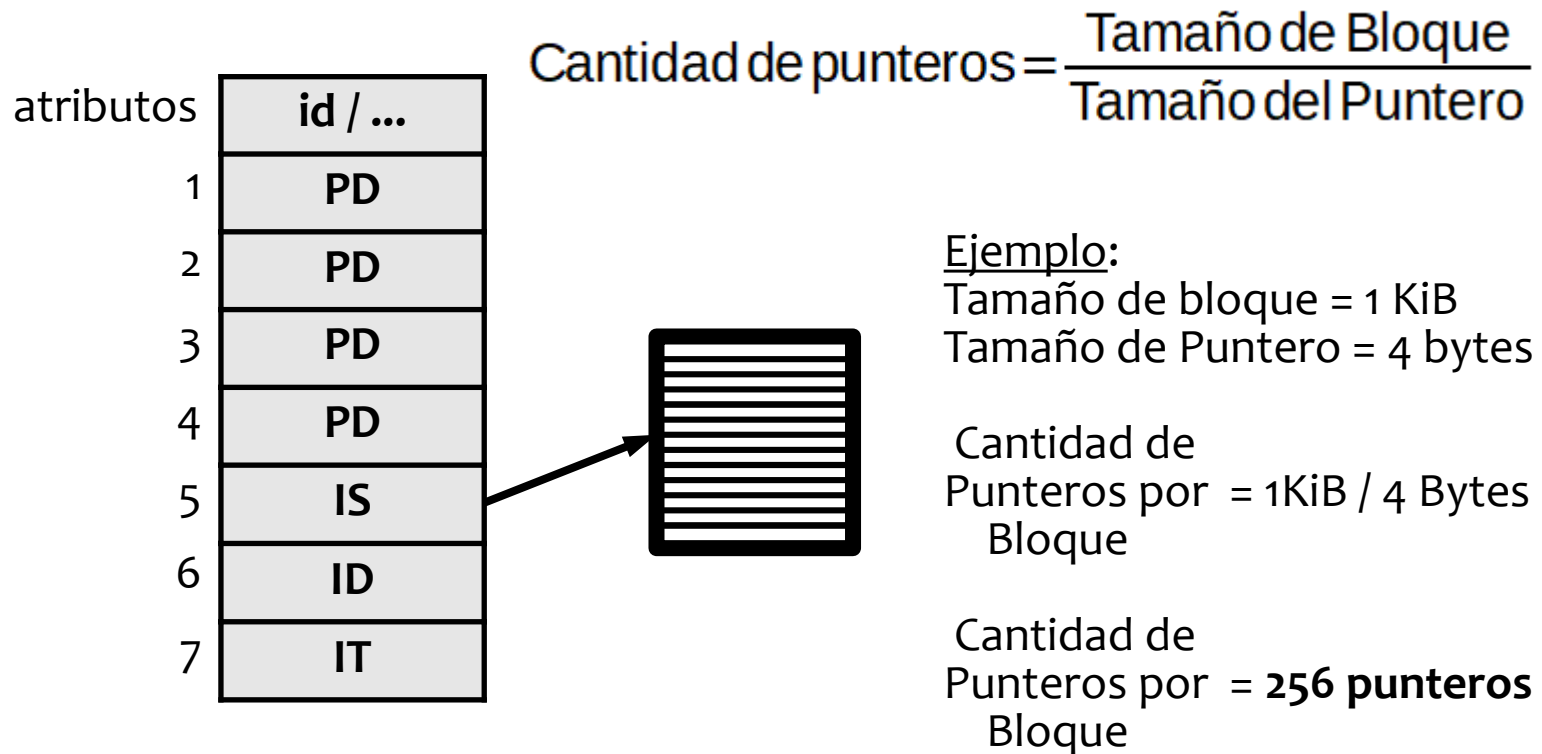
UFS (EXT2 / EXT3)

▪ Inodos



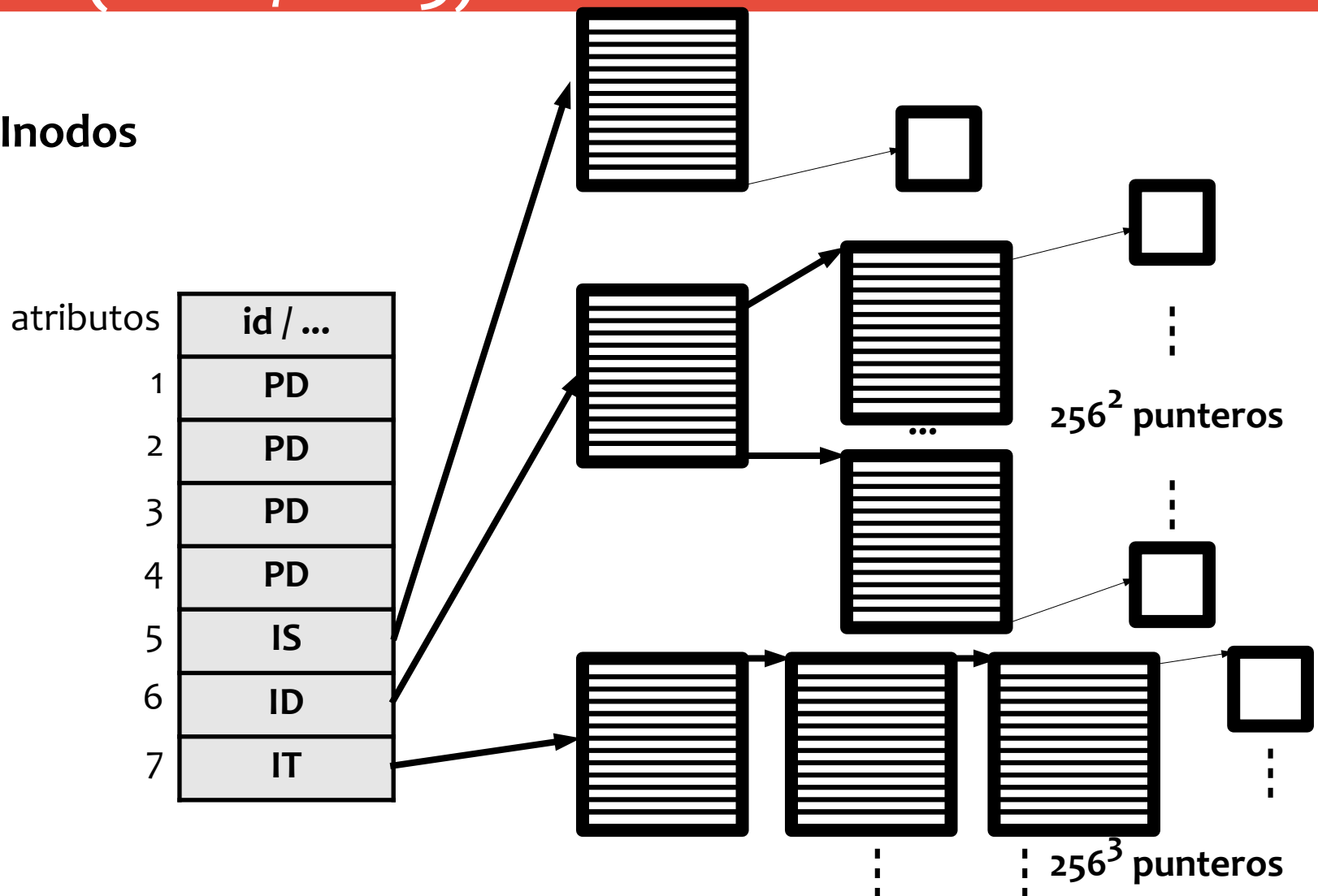
UFS (EXT2 / EXT3)

▪ Inodos



UFS (EXT2 / EXT3)

▪ Inodos



UFS (EXT2 / EXT3)

▪ Inodos

atributos	id / ...
1	PD
2	PD
3	PD
4	PD
5	IS
6	ID
7	IT

Tamaño Máximo Teórico de un archivo:

Tamaño de bloque = 1 KiB

Tamaño de Puntero = 4 bytes

Punteros por Bloque = 1 KiB / 4 bytes = 256

$$\left(\underset{\text{PD}}{4} + \underset{\text{IS}}{256} + \underset{\text{ID}}{256^2} + \underset{\text{IT}}{256^3} \right) * \underset{\text{bloque}}{1\text{KiB}} = 16 \text{ GiB}$$

Tamaño Máximo Real de un archivo:

Disco 8 GiB: Tamaño máximo real 8 GiB

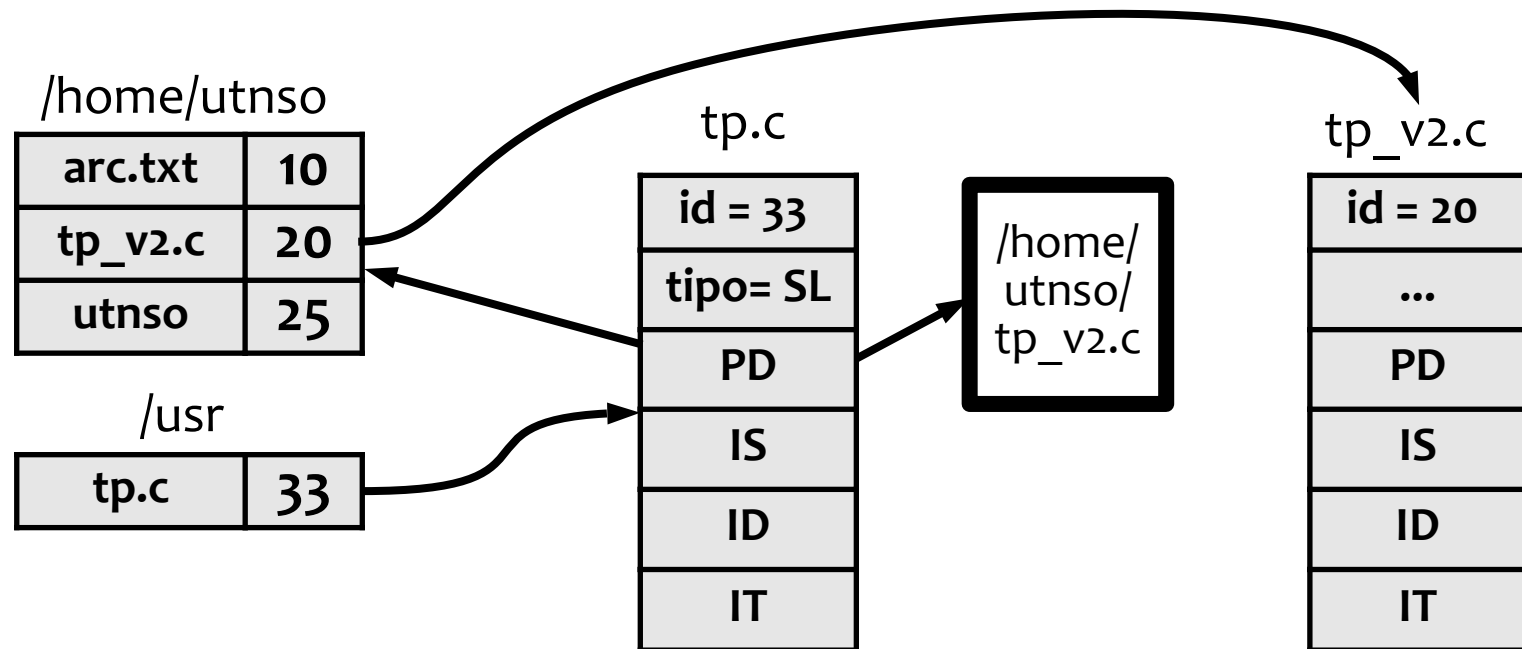
Disco 30 GiB: Tamaño máximo real 16 GiB

UFS (EXT₂ / EXT₃)

- **Accesos Directos (LINKS)**
 - **Soft Link / Symbolic Link**
 - **Hard Link**

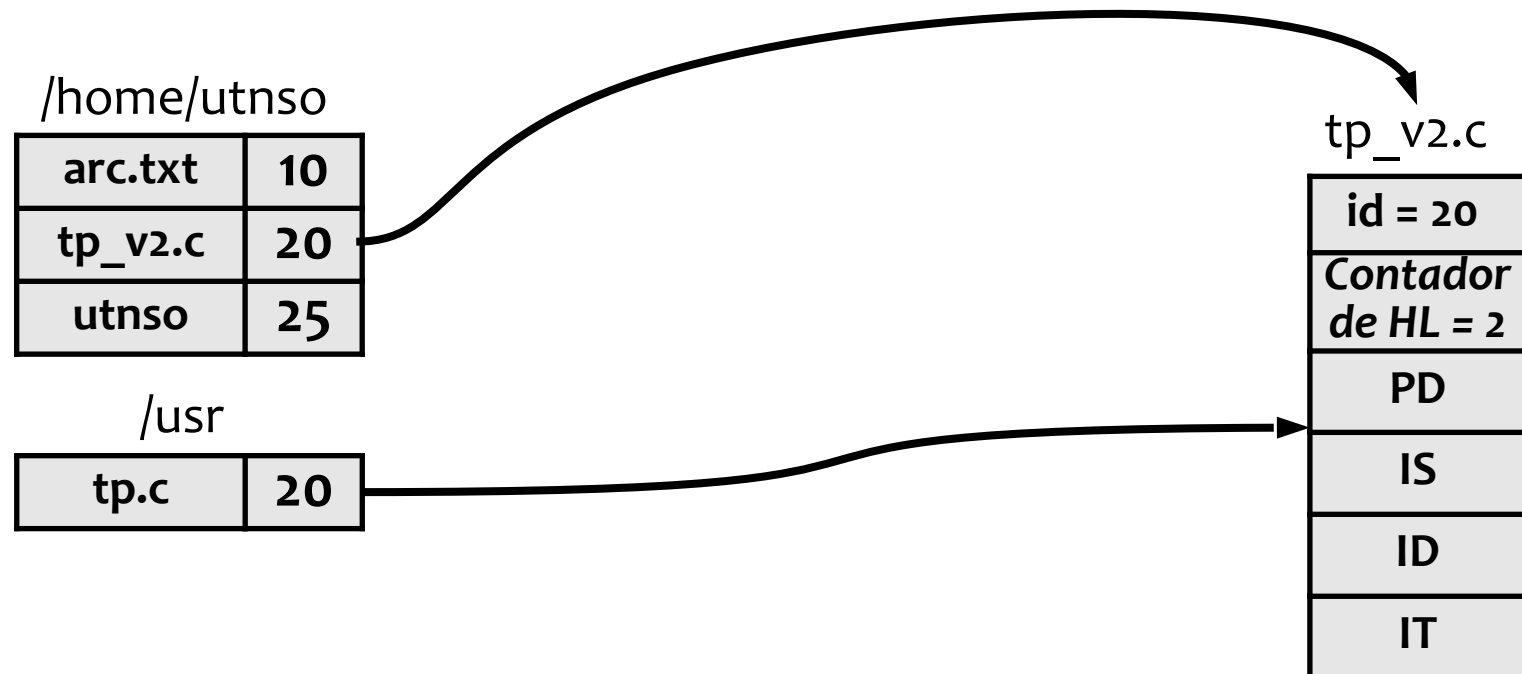
UFS (EXT2 / EXT3)

- Accesos Directos (LINKS)
 - Soft Link / Symbolic Link



UFS (EXT2 / EXT3)

- Accesos Directos (LINKS)
 - Hard Link



UFS (EXT₂ / EXT₃)

- Ejemplo:
- Punteros de 32 bits.
- Tamaño de Cluster 1 KiB.
- Tamaño máximo teórico del filesystem:
 - $2^{32} * 2^{10} = 2^{42} = 4 \text{ TiB}$
- Tamaño Máximo Real (Disco de 500 GiB)
 - $4 \text{ TiB} > 500 \text{ GiB}$ entonces es 500 GiB
- Tamaño Máximo Real (Disco de 8 TiB)
 - $4 \text{ TiB} < 8 \text{ TiB}$ entonces es 4 TiB

UFS (EXT₂ / EXT₃)

- Ejemplo: Tareas para agregar información en un archivo:
- Localizar el archivo.
 - Buscar a través de los directorios por medio de la ruta.
- Asignarle espacio.
 - Si es necesario asigna bloques libres al archivo.
 - Se modifica el bitmap de bloques y los punteros en inodo.
- Escribir los datos en el nuevo archivo.
 - Modificar atributos del archivo (tamaño, fechas)

FAT vs UFS

	FAT	UFS
Complejidad	Baja	Alta
FCB	No	Inodo
Espacio Libre	No	Bitmap
Journaling	No	Si
Links	No	Si
Seguridad	No	Si