

# Relación Tema 4.pdf



**LosCocos**



**Sistemas Operativos**



**2º Grado en Ingeniería Informática**



**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación  
Universidad de Granada**

**MARVEL STUDIOS**  
*Bruja Escarlata  
y Visión*



**Disney+**

Serie Original ya disponible  
en exclusiva





MARVEL STUDIOS

# Bruja Escarlata y Visión

Disney+

Serie Original ya disponible en exclusiva



# RELACION EJERCICIOS TEMA 4

3

- 1) Si es posible, gracias a que podemos jugar con el nombre de los archivos. Todos los que pertenecan al mismo nivel llevarán un número delante del nombre y, si tuviéramos dos archivos del mismo nombre, en un mismo nivel, al final del nombre se irán numerando con el alfabeto. Diferenciamos con "-" el nombre de archivo.  
Ej: 1-archivo, 2-archivo, 2-archivo-a

- 2) Se podría acudir a la estructura que controla el directorio inicial un campo que establezca el tamaño. Si alguna vez se sobrepasa el tamaño libre, el sistema lanzará una excepción para que se libere espacio.

3)

Nombre	Tipo	Fecha	Nº Bloque
DATOS	F	8-2-90	3
DATOS 1	F	1-3-90	1
DATOS 2	F	2-3-90	2
D	D	3-3-90	10
Cartera	F	13-3-90	6

	FAT		
1	*	10	*
2	4	11	*
3	15	12	
4	5	13	
5	*	14	
6	7	15	* 16
7	8	16	17
8	9	17	*
9	*	18	

1 1 1 1 / 1 1 1 1 / 1 1 0 0 / 0 0 1 1 / 1 0

- 5) Si está organizada como una lista enlazada, bastaría con buscar otro bloque libre, recorrer la secuencia de bloques libres enlazados y ver que regencia una lista.

- 6) Lista enlazada → Tamaño & bloques libres enlazados  
Mapa de bits → Tamaño bloques llenos (bit a uno)  
Cuando los bloques llenos sean más numerosos que los llenos, la lista enlazada ocupará menos, pues al tener más unos, aumenta el tamaño del mapa de bits.

7

Dentro de un programa, los archivos van a abrirse ~~de~~ solo durante el transcurso de ese programa, a través de un descriptor, y cuando el programa acaba, se eliminan. Es por esto que es necesaria la existencia de un bit de temporalidad.

8

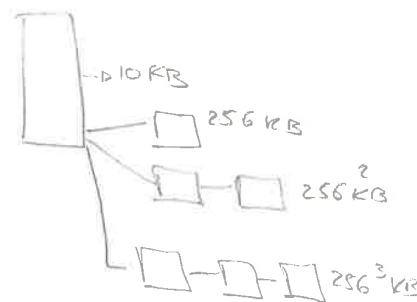
Con la instrucción **RENAME** solo cambiamos un campo de la estructura interna (i-nodo) asociada al archivo. Al copiar y borrar estamos creando una estructura nueva con campos nuevos y eliminando la anterior.

9

$$10 \text{ KB} + 256 \text{ KB} + 256^2 \text{ KB} + 256^3 \text{ KB}$$

$$= 5 \cdot 2^{11} + 2^{18} + 2^{16} \cdot 2^{10} + 2^{24} \cdot 2^{10} =$$

$$5 \cdot 2^{11} + 2^{18} + 2^{26} + 2^{34} \approx \underline{16 \text{ GB}}$$



10

a) Modificados *frecuentemente*, y accedidos *frecuentemente aleatoriamente*

→ **CONTIGUO**, no es necesario un método para modificar

b) Modificados *con frecuencia*, y accedidos *en su totalidad con*

*cierta frecuencia* → **NO CONTIGUO, ENLAZADO (Tabla FAT)**  
necesitamos un sistema para acceder y con *velocidad*

c) Modificados *frecuentemente* y accedidos *aleatoriamente y frecuentemente*

→ **NO CONTIGUO, INDEXADO**, ofrece la mejor velocidad de acceso *aleatorio*

11

FAT 32 → 32 bits/entrada = 4 bytes/entrada

Cluster datos = 16 KB

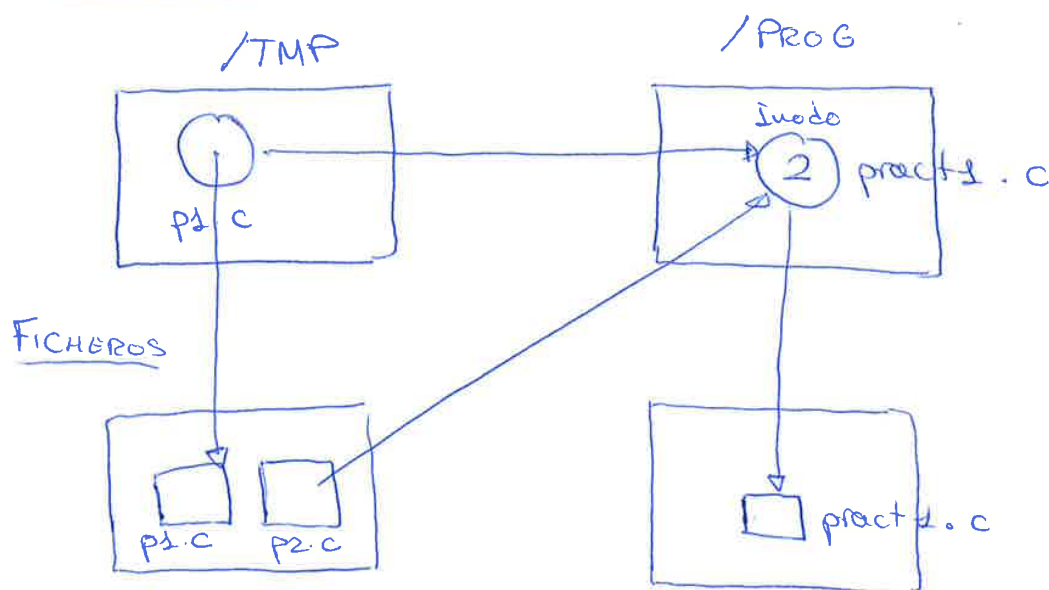
Partición = 20 GB → 4B x + 16 KB ≠ X

$$x = \frac{20 \text{ GB}}{4 \text{ B} + 16 \text{ KB}} = \frac{20 \cdot 2^{30} \text{ B}}{4 \text{ B} + 2^{14} \text{ B}} = 1310400 \text{ entradas}$$

$$\text{TAMAÑO FAT} = \text{Entrada} \times \text{Tam} = 1310400 \times 4 \text{ B} = \underline{5 \text{ MB}}$$

14

## DIRECTORIOS



Explicación → Hemos creado dentro del directorio tmp un enlace simbólico p1.c y otro absoluto p2.c.

- El enlace simbólico crea su propio nodo y este apunta a la estructura correspondiente del archivo con el que enlaza pract1.c. Es por eso que el contador de enlaces aumenta a 2. El mismo y el enlace simbólico.
- El enlace absoluto no crea nodo, solo se copia el puntero a la estructura con que se relaciona pract1.c. Es por eso que el contador de enlaces no aumenta.

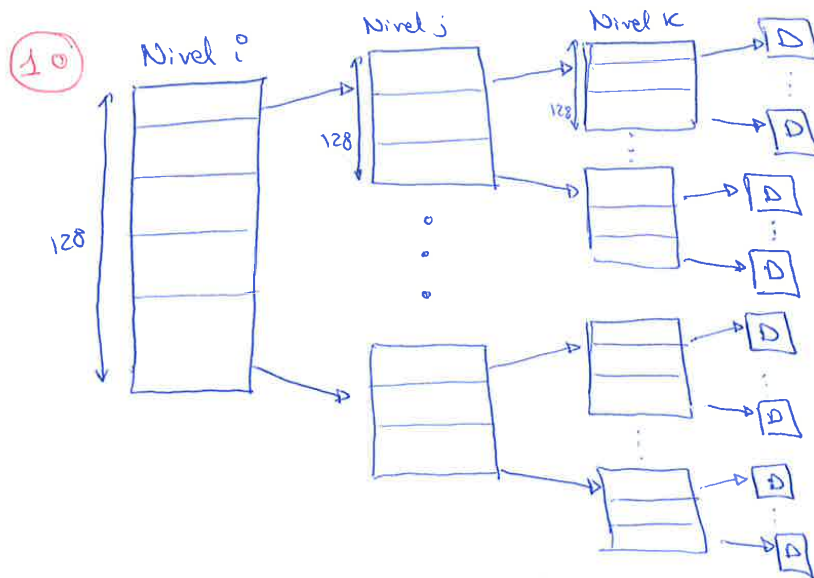
16

Porque podríamos tener varios nombres para el mismo archivo, dado por el mismo o por distintos usuarios.

Gracias a su descriptor de archivo, accedemos a él de forma inequívoca.

17

Sí se puede. No se podrá solo en caso de que el directorio fuese utilizado por un proceso.



TAM BLOQUE  $\Rightarrow$  512 B/bloque

TAM PUNTERO = TAM DIRECCIONES  $\rightarrow$  4 B/dir }  $\frac{512 \text{ B/bloque}}{4 \text{ B/dir}} = 128 \text{ dir/bloque}$

$$i = \frac{N}{128 \times 128 \times 512}$$

$$N' = N \% (128 \times 128 \times 512)$$

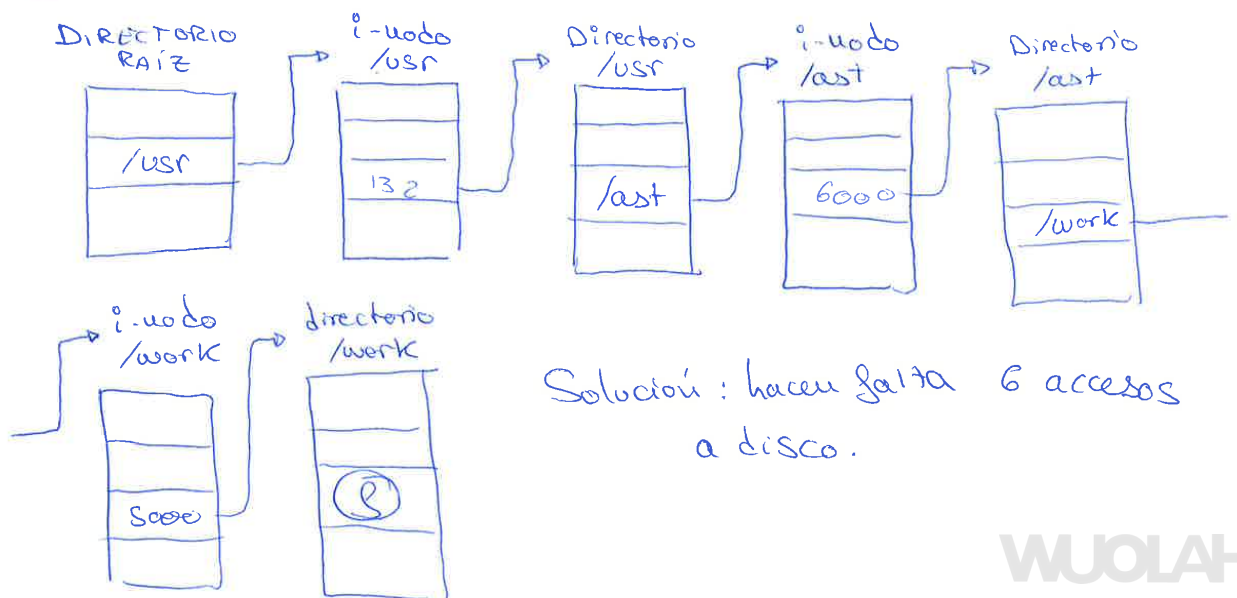
$$j = \frac{N'}{128 \times 512}$$

$$N'' = N' \% (128 \times 512)$$

$$K = \frac{N''}{512}$$

$$N''' = N'' \% 512 \quad N \text{ que necesitamos}$$

13 /usr/ast/work/8

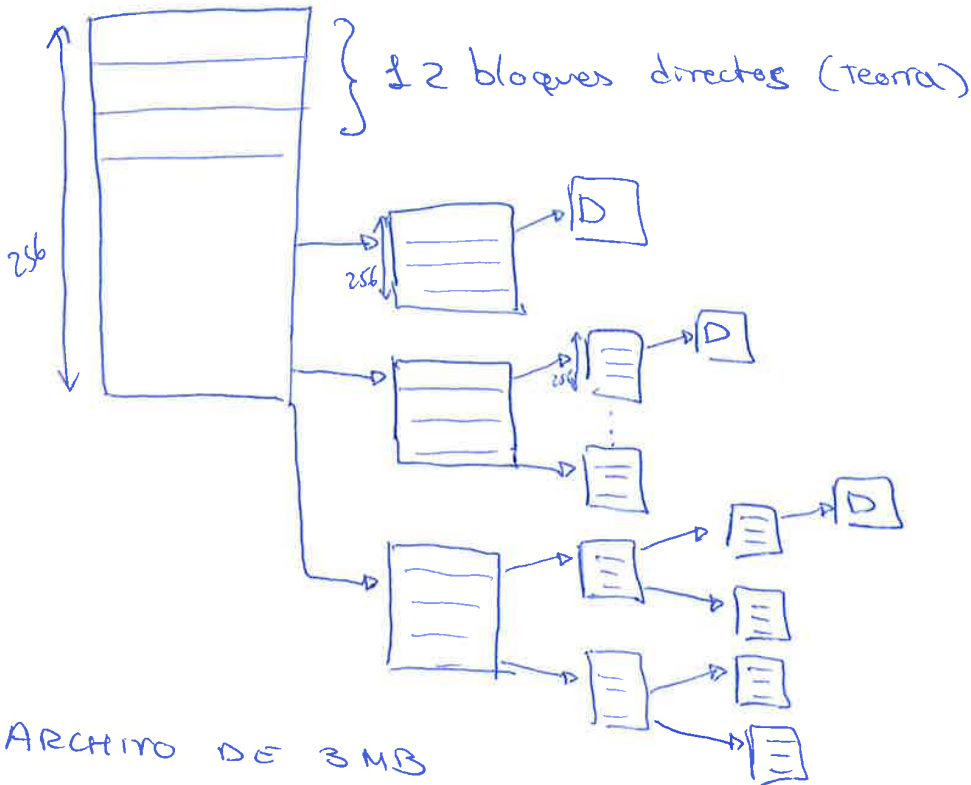


Solución: hacen falta 6 accesos a disco.



15

LINUX



ARCHIVO DE 3MB

TAM. BLOQUE  $\Rightarrow$  1KB/bloque

TAM. DIR  $\Rightarrow$  4B/dir

$$\frac{2^{10} \text{ B/bloque}}{2^2 \text{ B/dir}} = 256 \text{ dir/bloque}$$

Archivo de 3MB = 3072

3072 - 12 bloques directos = 3060

¿Cabe en el primer nivel? NO  $\rightarrow$  3060 - 256 = 2804 a direccionar

¿Cabe en el segundo nivel? Si ¿Cuántos bloques necesita?

$$2804 / 256 = 11 \text{ bloques}$$

BLOQUES TOTALES = 11 + 1 + 1 = 13 bloques

Los cocos



WUOLAH

