

1.- Calcular el número de accesos a disco necesarios (para el caso peor y para el caso mejor) para leer 20 bloques lógicos consecutivos (no necesariamente los 20 primeros) de un fichero en un sistema con:

- a) Asignación contigua
- b) Asignación no contigua mediante índice enlazado (FAT)
- c) Asignación no contigua indexada directa

Considere además que:

- I No hay ningún dato relacionado con el sistema de ficheros en memoria RAM.
- II El fichero se encuentra en el directorio raíz, cuya ubicación en disco es conocida, y que dicho directorio ocupa 2 bloques de datos.
- III La tabla FAT ocupa 4 bloques.

a) Hay que acceder al disco porque no lo hay en RAM

1 ó 2 + 20 accesos.

(porque la dirección ocupa 2 bloques).

b) 1º al directorio (1-2 bloques)

2º Recorrer la tabla FAT hasta encontrar el 1er Bloque que nos interesa (1-4 Bloques)



c) 1º al directorio (1-2 Bloques)

2º Acceso a la tabl de nodos-i 1 Bloque.

(Si sólo usamos enlaces directos porque necesitamos un acceso por el bloque)

Mejor Caso: 1+1+20

Peor Caso: 2+1+2

2.- Un dispositivo de memoria flash de 64 MiB de capacidad y bloques de 1KiB, contiene un sistema de ficheros FAT.

a. ¿Cuántos bytes son necesarios para almacenar la tabla FAT?

I 64 KiB

II 128 KiB

III 1 MiB

IV 512 KiB

V Ninguna de las respuestas anteriores es correcta

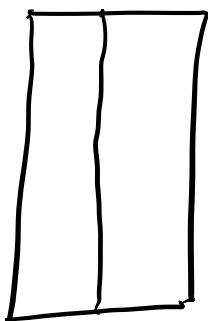
b. ¿Es posible realizar enlaces rígidos en un sistema de ficheros tipo FAT?

c. ¿Por qué no se puede establecer enlaces rígidos a ficheros de volúmenes distintos en sistemas de ficheros tipo ext2?

$$\text{Tam. Disco} = 64 \text{ MiB} = 2^6 \cdot 2^{20} = 2^{26} \text{ B}$$

$$\text{Tam. Bloque} = 1 \text{ KiB} = 2^{10} \text{ B}$$

FAT

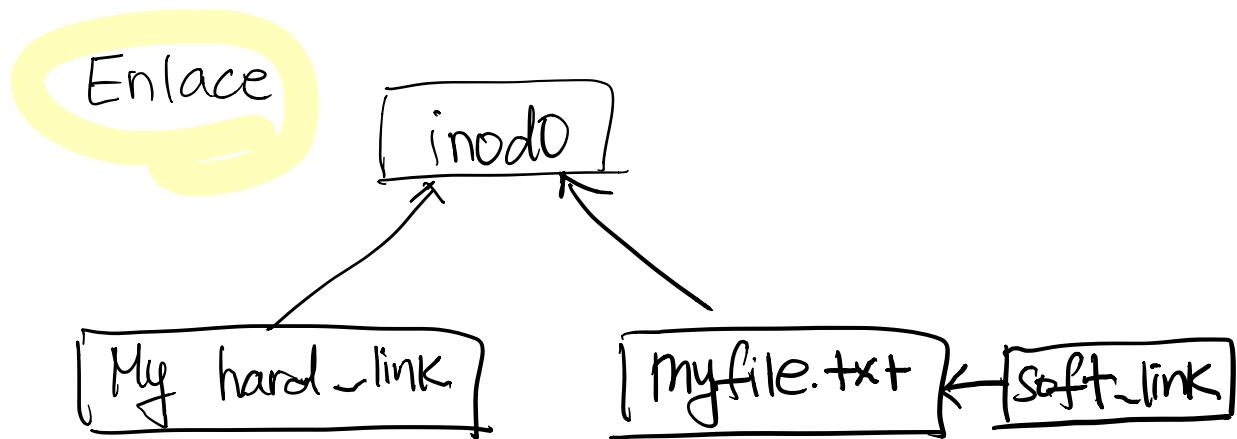


$$\frac{2^{26}}{2^{10}} = 2^{16} \text{ Bloques}$$

Para posicionar bloques necesitamos 16 bits (2 Bytes)

$$\text{Tam. FAT} = n^{\text{*}} \cdot \text{Bytes necesario para posicionar}$$

$$= 2^{16} \cdot 2 = 2^{17} = 2^7 \text{ KiB}$$



b) Enlace duro puede tener varios nombres para un mismo fichero. Se debería buscar en todo el sistema de ficheros sus atributos.

Al no tener una estructura cerrada → mucho coste
→ No se puede tener.

c) Se mira en tablas distintas.

3.- Un sistema de ficheros UNIX utiliza bloques de 1024 bytes, para cuyo direccionamiento se emplean direcciones de 16 bits. Los nodos-i (entradas en una tabla que contiene la información descriptiva de los ficheros) contienen 8 direcciones de disco para bloques de datos, una dirección de bloque índice indirecto simple y una dirección de bloque índice indirecto doble. Para indicar el tamaño del fichero y el desplazamiento, (offset) de la posición en bytes en las operaciones read y write, se utilizan números de 64 bits. Conteste razonadamente a las siguientes cuestiones:

1. ¿Cuál es el tamaño máximo de un fichero en este sistema? ¿y el de la partición?
2. ¿Qué deberíamos cambiar para aumentar el tamaño de fichero (cambio de tamaño de bloque, punteros, incluir indirecto triple)? ¿Cuál sería el nuevo tamaño máximo de fichero?

Tam. Bloque : 1024 B
direcciónamiento: 16 Bits

Los nodos-i: Directo 8
indirecto Simple 1
indirecto doble 1.

a) Tam. max Fichero = min (T_{ESNI} , $T_{maxN^{\#} bloq}$, T_{offset})

Tamaño máximo de fichero



- Tamaño máximo de archivo según organización:

$$T_B \times (E_d + T_B/B_{id} + (T_B/B_{id})^2 + (T_B/B_{id})^3)$$

- T_B : tamaño de bloque
- E_d : número de enlaces directos en el nodo-i
- B_{id} : número de bytes usados para el identificador de bloque físico.

- Tamaño máximo de archivo según dirección:

$$T_B \times \frac{2^{8B_{id}}}{2} \text{ bits usados}$$

- Es una aproximación que desprecia el espacio usado para almacenar el propio índice (usa todos los bloques para los datos del fichero)

Tam máx según organización

$$= 2^{10} \cdot \left(8 + \frac{2^{10}}{2} + \left(\frac{2^{10}}{2} \right)^2 \right) = 2^{10} \cdot (2^3 + 2^9 + 2^{18}) \text{ bytes}$$

Tam máx por nº Bloq / de partición / según dirección

$$1024 \cdot 2^{\frac{8(2)}{2}} = 2^{10} \cdot 2^{16} = \underline{\underline{2^{26} \text{ Bytes}}}$$

Tam según offset = 2^{64} Bytes

b) Para aumentar el tamaño del fichero, hay que aumentar el tamaño del bloque, el nº de punteros y añadir el enlace de indirecto simple.

4.- Un programa UNIX crea un fichero e inmediatamente se posiciona (con lseek) en el byte 55 millones. Luego escribe un byte. ¿Cuántos bloques de disco ocupa ahora el fichero (incluyendo bloques indirectos)? Asumase que los nodos-i contienen 10 índices directos, 1 índice indirecto simple, 1 índice indirecto doble y 1 índice indirecto triple, los bloques tienen un tamaño de 2 Kbytes y el tamaño de los índices (direcciones de bloques de disco) es de 32 bits. ¿Qué sucesión de índices lógicos nos lleva al byte posicionado por lseek?

offset 55000000 + 1

enlaces: 10 directos

1 ind. simple

1 ind. doble

1 ind. triple

tam. Bloq = 2 KiB = $2 \cdot 2^{10} = 2^6 B$

Direccionalmiento = 32 bits

↓

4Bytes

$$\frac{55000001}{2''} = 26855.47$$

26855 bloques enteros que me he saltado

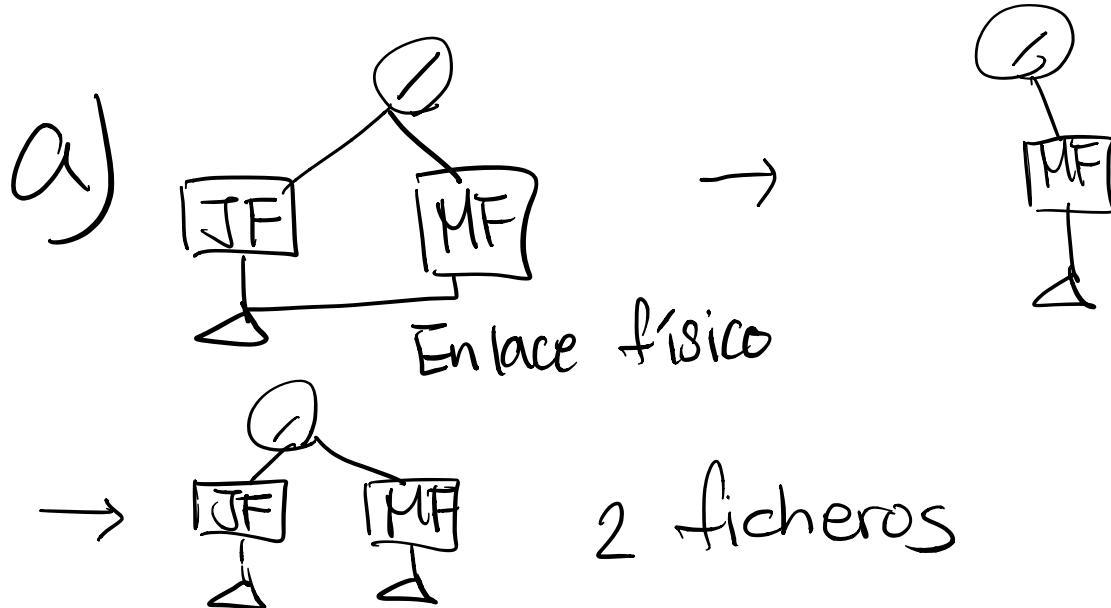
26856 es el bloque lógico donde está la posición del $55 \cdot 10^6$

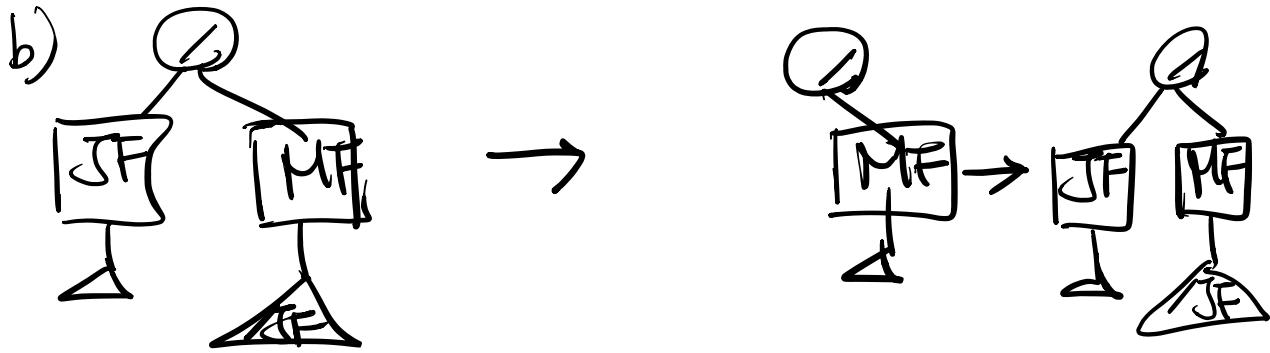
$$10 + \frac{\frac{2''}{2^2}}{2^2} + \left(\frac{\frac{2''}{2^2}}{2^2} \right)^2 = 262.666$$

En disco ocupa 3 bloques
1 de datos
2 para el índice

5.- Juan crea un fichero de nombre JFichero. María crea un enlace físico a JFichero y le da el nombre MFichero. Juan elimina JFichero. Luego Juan crea un nuevo fichero y también le llama JFichero.

- ¿Cuántos ficheros diferentes existen después de las acciones anteriores?
- ¿Sería diferente la respuesta si el enlace que ha creado María fuese un enlace simbólico de nombre MFichero que apuntara a JFichero?





6.- En la siguiente figura se representa una tabla FAT. Al borde de sus entradas se ha escrito, como ayuda de referencia, el número correspondiente al bloque en cuestión. También se ha representado la entrada de cierto directorio. Como simplificación del ejemplo, suponemos que en cada entrada del directorio se almacena: Nombre de archivo/directorio, el tipo (F=archivo, D=directorio), la fecha de creación y el número del bloque inicial.

#Bloque	Índice	#Bloque	Índice
1	<EOF>	10	11
2	4	11	R
3	15	12	<EOF>
4	5	13	
5	<EOF>	14	
6	7	15	<EOF> 6
7	<EOF>	16	
8	<EOF>	17	
9	10	18	

Nombre	Tipo	Fecha	#Bloque
DATOS	F	08/02/2015	3
DATOS1	F	01/03/2017	1
DATOS2	F	02/03/2017	2
P	D	03/03/2017	8
CARTAS	F	13/03/2017	9

- a. Rellene la figura para representar lo siguiente (Nota: supóngase tamaño de bloque 512 Bytes y que siempre se elige el primer bloque vacío como política de asignación):

- I Creación del archivo DATOS1 con fecha 1-3-2017, y tamaño de 10 Bytes.
- II Creación del archivo DATOS2 con fecha 2-3-2017, y tamaño 1200 Bytes. → 3 Bloques
- III El archivo DATOS aumenta de tamaño, necesitando 2 bloques más.
- IV Creación del directorio D, con fecha 3-3-2017, y tamaño 1 bloque.
- V Creación del archivo CARTAS con fecha 13-3-2017 y tamaño 2 KiBytes. → 4 Bloq.

- b. Si usamos un Mapa de Bits para la gestión del espacio libre, especifique la sucesión de bits que contendría respecto a los 18 bloques de la tabla anterior.

1 1 1 1 1 1 1 1 1 0 0 1 0 0 0

7.- Un sistema de ficheros basado en i-nodos y mapa de bits contiene la siguiente información:

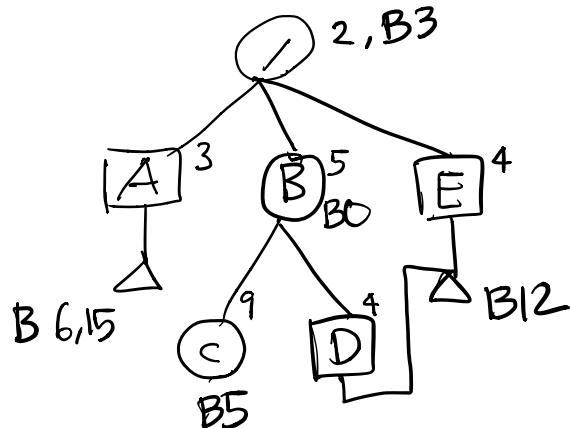
Mapa de bits: 1 0 0 1 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0

siempre raíz

i-nodo 2	i-nodo 3	i-nodo 4	i-nodo 5	i-nodo 9
Tamaño 1	Tamaño 2	Tamaño 1	Tamaño	Tamaño 1
#Enlaces NA	#Enlaces 1	#Enlaces 2	#Enlaces NA	#Enlaces NA
Tipo F/D D	Tipo F/D F	Tipo F/D F	Tipo F/D D	Tipo F/D D
Directo 3	Directo 6	Directo 12	Directo 0	Directo 5
Indirecto Nil	Indirecto 7	Indirecto Nil	Indirecto Nil	Indirecto Nil

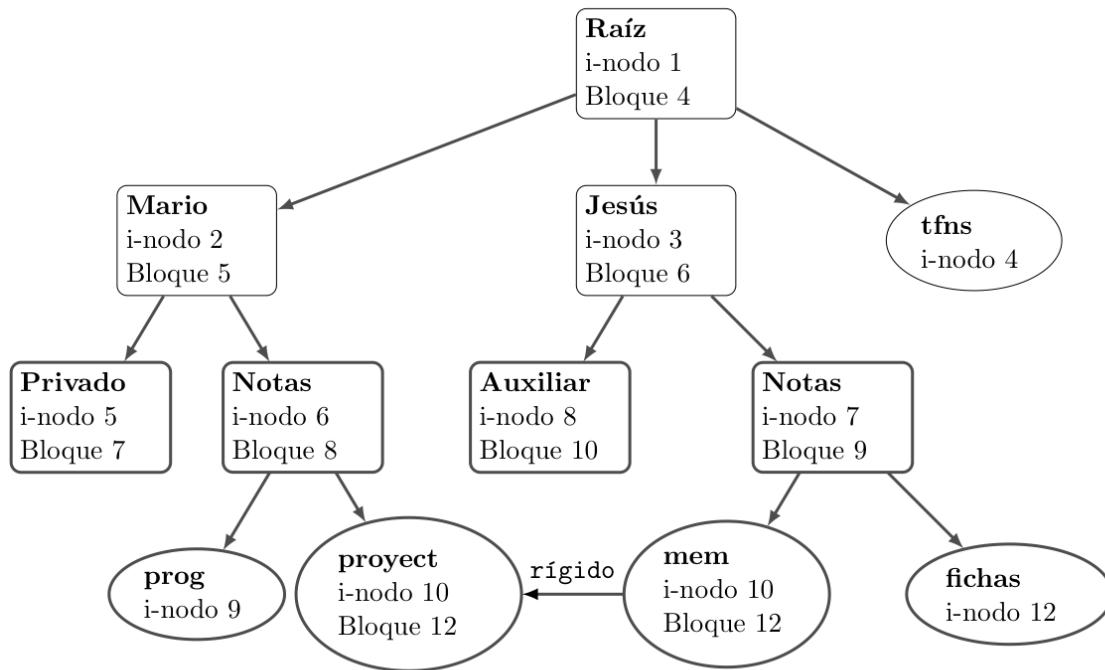
Bloque 0	Bloque 3	Bloque 5	Bloque 6	Bloque 7	Bloque 12	Bloque 15
. 5	. 2	. 9	Datos sin formato	15 NULL NULL	Datos sin formato	Datos sin formato
.. 2	.. 2	.. 5				
C 9	A 3					
D 4	B 5					
E 4						

- Rellene los huecos para que el sistema sea consistente. Asuma para ello que el tamaño se expresa en bloques.
- Dibuje el árbol del directorio empleando óvalos para los directorios, rectángulos para los ficheros y triángulos para los datos



8.- El sistema de ficheros de un SO diseñado a partir de UNIX utiliza bloques de disco de 1024 bytes de capacidad. Para el direccionamiento de estos bloques se utilizan punteros de 16 bits. Para indicar el tamaño del fichero y el desplazamiento (*offset*) de la posición en bytes en las operaciones **read** y **write**, se utilizan números de 64 bits. Cada nodo-i tiene 8 punteros de direccionamiento directo, 1 puntero indirecto simple y 1 puntero indirecto doble.

- ¿Cuál será el tamaño máximo de un fichero suponiendo despreciable el espacio ocupado por el superbloque y la tabla de nodos-i?
- Si se modifica el tamaño de puntero pasándolo a 32 bits, ¿cuál será el nuevo tamaño máximo?
- Dada la siguiente estructura de directorio, en el que Jesús comparte el fichero **proyect** de Mario mediante un enlace rígido de nombre **mem**, indicar los contenidos de los directorios y nodos-i que se encontrará el sistema (y el orden en que se los encontrará) al hacer la búsqueda del fichero memoria desde el directorio raíz.



Nota: los directorios se muestran como rectángulos y los ficheros como óvalos.

Tam. Disco = 2^{10} B direcciónamiento: 16 bits
 offset: 64 bits enlaces: 8 directos
 1 ind. simple
 1 ind. doble

a) el tam. máx. Fichero
 $= \min(T_{organización}, T_{dirección}, T_{offset})$

Torganización

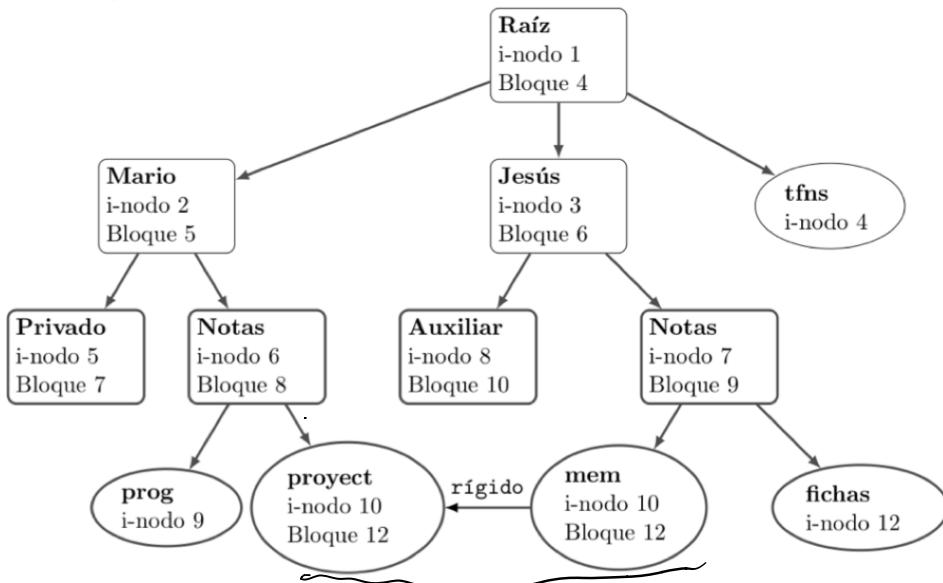
$$\begin{aligned} &= 2^{10} \cdot \left(8 + \frac{2^{10}}{2} + \left(\frac{2^{10}}{2} \right)^2 \right) \\ &= 2^{10} \cdot (2^3 + 2^9 + 2^{18}) = 2^{13} + 2^{19} + 2^{28} \\ &= 2^{13} (1 + 2^6 + 2^{15}) \end{aligned}$$

Tdirecccionamiento

$$= 2^{10} \cdot 2^{16} = 2^{26} \text{ B}$$

$$T_{\text{offset}} = 2^{64}$$

- c. Dada la siguiente estructura de directorio, en el que Jesús comparte el fichero **proyect** de Mario mediante un enlace rígido de nombre **mem**, indicar los contenidos de los directorios y nodos-i que se encontrará el sistema (y el orden en que se los encontrará) al hacer la búsqueda del fichero memoria desde el directorio raíz.



Nota: los directorios se muestran como rectángulos y los ficheros como óvalos.

/ Jesus / Notas / mem

/ Mario / Notas / project

1. Acceso a Nodo-i 2
2. Acceso al bloque 4
3. Acceso al nodo-i 3
4. Acceso al bloque 6
5. Acceso al nodo-i 7
6. Acceso al bloque 9
7. al nodo-i 10
8. al bloque 12

B4

.	1
..	2
Mario	2
Jesus	3
Ferns	4

B6

.	3
..	2
Auxiliar	8
Notas	7

B9

.	7
..	3
mem	10
fichas	12

B12

contenido
del
fichero

1 3 7 10

TAM	1	1	1	1
#enlace	1	1	1	2
Tipo	D	D	D	F
Directo	4	6	9	12