

Sistemas de archivos

Sistemas Operativos

Franco Frizzo

Departamento de Computación, FCEyN, UBA

9 de mayo de 2017

Primer cuatrimestre de 2017

Ejercicio 1

- 1** Se tiene un disco con bloques de 8 KB, y un sistema de archivos donde los bloques de un archivo se almacenan en forma contigua. Considerar un archivo de 10 MB, para el que se conoce en qué posición del disco se encuentra su primer bloque. Suponer que hay bloques libres luego del final del archivo, pero no antes del comienzo.
- ¿Cuántas operaciones de disco son necesarias para...
 - (a) agregar un bloque al principio del archivo?
 - (b) agregar un bloque en la mitad del archivo?
 - (c) agregar un bloque al final del archivo?
 - (d) eliminar el primer bloque del archivo?
 - (e) eliminar el bloque en la mitad del archivo?
 - (f) eliminar el último bloque del archivo?

Ejercicio 1

- 2** Se tiene un disco con bloques de 8 KB y un sistema de archivos con una FAT, que está cargada en memoria. Considerar un archivo de 10 MB, para el que se conoce la dirección de su primer bloque.

■ ¿Cuántas operaciones de disco son necesarias para...

- (a) agregar un bloque al principio del archivo?
- (b) agregar un bloque en la mitad del archivo?
- (c) agregar un bloque al final del archivo?
- (d) eliminar el primer bloque del archivo?
- (e) eliminar el bloque en la mitad del archivo?
- (f) eliminar el último bloque del archivo?

¿Cuántas posiciones de la FAT deben consultarse y cuántas deben modificarse en cada caso?

- Comparar con los resultados obtenidos para la asignación contigua.

Ejercicio 1

- 3 Se tiene un disco con bloques de 8 KB y un sistema de archivos con inodos, sin direcciones. Considerar un archivo de 10 MB, cuyo inodo está cargado en memoria.

■ ¿Cuántas operaciones de disco son necesarias para...

- (a) agregar un bloque al principio del archivo?
- (b) agregar un bloque en la mitad del archivo?
- (c) agregar un bloque al final del archivo?
- (d) eliminar el primer bloque del archivo?
- (e) eliminar el bloque en la mitad del archivo?
- (f) eliminar el último bloque del archivo?

¿Cuántas posiciones del inodo deben consultarse y cuántas deben modificarse en cada caso?

- Comparar con los resultados obtenidos para la asignación contigua y con FAT.

Ejercicio 2

Se tiene un disco con capacidad de 128 GB, y bloques de 8 KB. Supongamos un sistema de archivos similar a FAT, donde la tabla se ubica desde la posición 0.

- 1 ¿Cuál es el tamaño que ocupará la tabla?
- 2 ¿Cuál es la cantidad máxima de archivos de 10 MB que es posible almacenar?
- 3 Se sabe que un archivo comienza en el bloque 20. Dada la siguiente FAT, indicar el tamaño de dicho archivo.

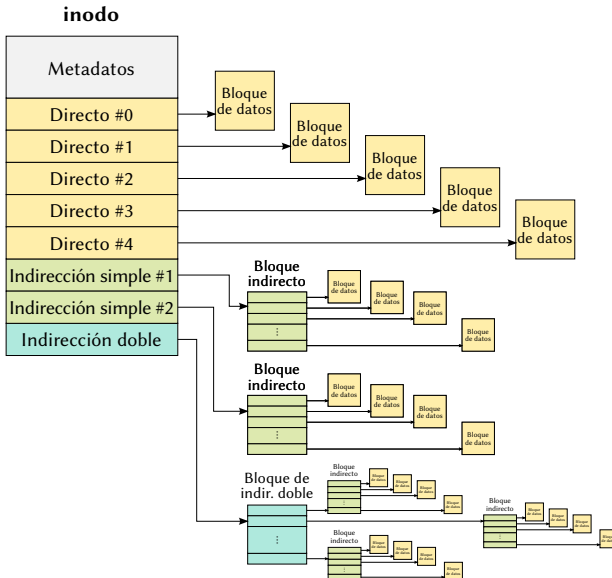
Bloque	0	1	2	3	4	5	6	...	20	21	22	...
Siguiente	EOF	2	23	4	5	0	7	...	21	22	3	...

Ejercicio 3

Un sistema de archivos de UNIX, tipo ext2, tiene bloques de tamaño 4 KB y el direccionamiento a bloques de disco (LBA) es de 8 bytes. A su vez, cada inodo cuenta con 5 entradas directas, dos indirectas y una doblemente indirecta.

- 1 ¿Cuál es el tamaño máximo de archivo que soporta?
- 2 Si el 50% del espacio en disco es ocupado por archivos de 2 KB, el 25% por archivos de 4 KB y el 25% restante por archivos de 8 KB, ¿qué porcentaje del espacio en disco es desperdiciado? (Considere solo el espacio utilizado en los bloques de datos).
- 3 ¿Cuántos bloques hace falta acceder para procesar completo un archivo de 5 MB?

Ejercicio 3



Ejercicio 4

- ¿Qué ventajas tiene un sistema que utiliza inodos sobre uno de tipo FAT?
- ¿Qué ventajas tiene un sistema de tipo FAT sobre uno que utiliza inodos?

Ejercicio 5

Escribir el pseudocódigo de la función

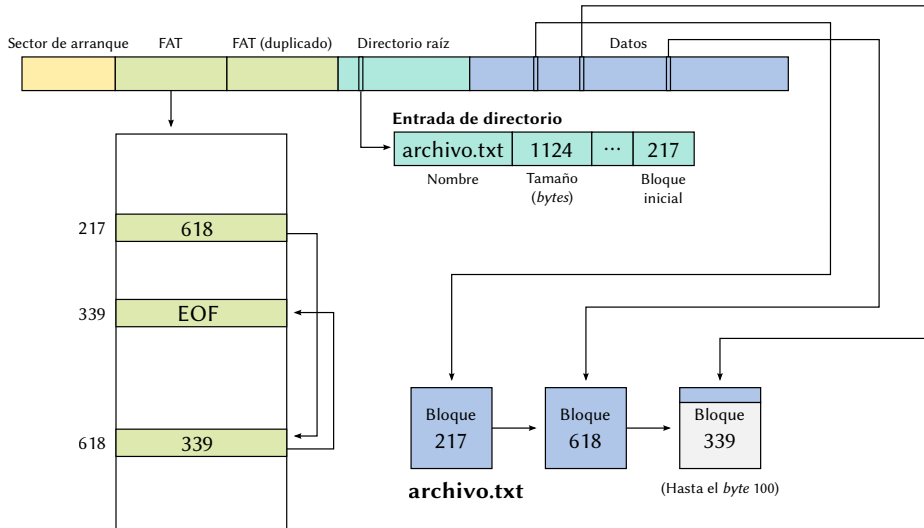
```
char* read_file(string name)
```

que devuelve el puntero a un *buffer* con el contenido de un archivo ubicado en el directorio *root* de un sistema de archivos FAT.

Se puede asumir que se dispone de:

- las funciones:
 - `lba_t fat_lookup(lba_t block_index).`
 - `char* allocate_buffer(int size).`
 - `void copy_block(lba_t orig_lba, char* dst_buffer).`
- una estructura, `fat_dir_entry`, que representa una entrada de directorio de un sistema de archivos FAT.
- las constantes `BLOCK_SIZE`, `ROOT_START_BLOCK` y `ROOT_SIZE`.

Recuperación de un archivo en FAT*



*Suponiendo bloques de 512 bytes.