

# Sistemas Operativos UCM 2012/2013

## Módulo 2 Gestión de Archivos y Directorios

1. Considerar un sistema donde el espacio libre se especifica en una lista de espacios libres.
  - a) Suponer que se pierde el puntero a la lista. ¿Puede el sistema reconstruirla?
  - b) Sugerir un esquema que asegure que el puntero nunca se pierda como resultado de un fallo de memoria.
2. 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

**Nota:** Asumir que no hay ningún dato relacionado con el sistema de ficheros en memoria RAM y que el fichero se encuentra en el directorio actual.

3. Un dispositivo de memoria flash de 64 MB de capacidad y bloques de 1KB, contiene un sistema de ficheros FAT. ¿Cuántos bytes son necesarios para almacenar la tabla FAT?
  - a. 64 KB
  - b. 128 KB
  - c. 1 MB
  - d. 512 KB
  - e. Ninguna de las respuestas anteriores es correcta

¿Es posible realizar enlaces rígidos en un sistema de ficheros tipo FAT? ¿Por qué no se puede establecer enlaces rígidos a ficheros de volúmenes distintos en sistemas de ficheros tipo EXT2?

4. 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		10	
2		11	
3	15	12	
4		13	
5		14	
6		15	<EOF>
7		16	
8		17	
9		18	

Nombre	Tipo	Fecha	Nº Bloque
DATOS	F	8-2-05	3

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):

- a) Creación del archivo *DATOS1* con fecha 1-3-90, y tamaño de 10 Bytes.
- b) Creación del archivo *DATOS2* con fecha 2-3-90, y tamaño 1200 Bytes.
- c) El archivo *DATOS* aumenta de tamaño, necesitando 2 bloques más.

- d) Creación del directorio *D*, con fecha 3-3-90, y tamaño 1 bloque.
- e) Creación del archivo *CARTAS* con fecha 13-3-90 y tamaño 2 KBytes.

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.

5. Considerar un fichero que consta de 100 bloques. ¿Cuántas operaciones de disco son necesarias para cada una de las tres estrategias de asignación (contigua, enlazada e indexada) al realizar las siguientes operaciones?:

- a) Añadir un bloque de información al comienzo
- b) Añadirlo a la mitad
- c) Añadirlo al final
- d) Suprimirlo del principio
- e) Suprimirlo de la mitad
- f) Suprimirlo del final.

6. Un sistema de ficheros *UNIX* utiliza bloques de 1024 bytes y direcciones de disco de 16 bits. Los *i-nodos* (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. ¿Cuál es el tamaño máximo de un fichero en este sistema? ¿Y el de la partición?

7. Dar 5 nombres de ruta diferentes para el fichero */etc/passwd*. (Sugerencia: considerar las entradas de directorio *.* y *..*)

8. ¿Cuántos accesos a disco son necesarios para abrir el fichero *games/chess* en *UNIX*?

9. 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)? Asíumase que los *i-nodos* 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*?

10. 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*?

11. Sugerir una razón por la cuál alguien pudiera desear construir un directorio sobre los cuales los demás usuarios tuvieran permiso de ejecución pero no de lectura, en un sistema de ficheros de tipo *UNIX*.

12. Dos estudiantes de informática, *Estudiante1* y *Estudiante2*, sostienen una discusión respecto a los *i-nodos*. *Estudiante1* argumenta que, dado que las memorias son cada vez más grandes y baratas, cuando se abre un fichero es más sencillo y rápido obtener una nueva copia del *i-nodo* para llevarla a la tabla de *i-nodos* en memoria, que buscar en la tabla entera para comprobar si ya está allí. *Estudiante2* discrepa. ¿Quién tiene razón?

13. 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

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		#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	
Indirecto	Null	Indirecto	7	Indirecto	Null	Indirecto	Null	Indirecto	Null

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

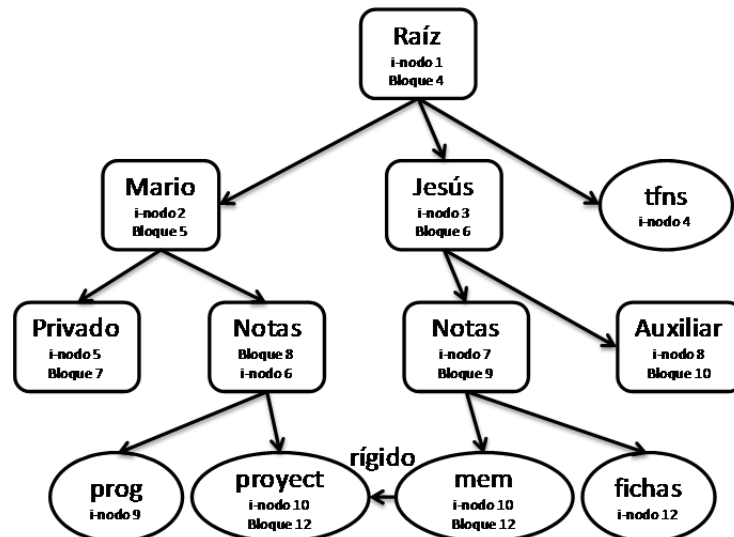
- Rellene los huecos para que el sistema sea consistente. Asuma para ello que el tamaño se expresa en bloques.
- Dibuje el árbol del directorio empleando óvalos para los directorios, rectángulos para los ficheros y triángulos para los datos

**14.** Un sistema de ficheros UNIX utiliza bloques de 512 bytes y direcciones de disco de 16 bits. Los **nodos-i** contienen 10 direcciones de disco para bloques de datos, una dirección de bloque índice indirecto simple y una dirección de bloque índice indirecto doble. Conteste de manera razonada a las siguientes cuestiones:

- ¿Cuál es el tamaño máximo de un fichero en este sistema?
- Un programa UNIX crea un fichero en este sistema e inmediatamente después escribe un byte de datos en la posición 1.000 y otro en la posición 10.000. ¿Cuántos bloques de datos ocupa este nuevo fichero en disco?

**15.** 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 i-nodo 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 i-nodos?
- 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 e i-nodos 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.

**16.** Dado un sistema de ficheros tipo UNIX, en el que los directorios son relativamente pequeños (cabén en un bloque de disco) y únicamente tiene una partición; razona qué operaciones de disco se necesitan para abrir el archivo `"/usr/curso3/SO/alumnos.txt"`. Suponer que el i-nodo del directorio raíz está ya en memoria y que no se ha cargado en memoria anteriormente ningún otro elemento de la ruta.