

## Tema 4: Sistema de Archivos

1. Sea un Sistema Operativo que sólo soporta un directorio (es decir, todos los archivos existentes estarán al mismo nivel), pero permite que los nombres de archivo sean de longitud variable. Apoyándonos únicamente en los servicios proporcionados por este Sistema Operativo, deseamos construir una "utilidad" que "simule" un sistema jerárquico de archivos. ¿Es esto posible? ¿Cómo?
2. En un entorno multiusuario, cada usuario tiene un directorio inicial al entrar en el sistema a partir del cual puede crear archivos y subdirectorios. Surge, entonces, la necesidad de limitar el tamaño de este directorio para impedir que el usuario consuma un espacio de disco excesivo. ¿De qué forma el Sistema Operativo podría implementar la limitación de tamaño de un directorio?
3. 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.

Nombre      Tipo    Fecha    N°Bloque

DATOS	F	8-2-90	3

FAT

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

Tenga en cuenta que:

- el tamaño de bloque es de 512 bytes
- el asterisco indica último bloque
- todo lo que está en blanco en la figura está libre.

Rellene la figura para representar lo siguiente:

- 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.
4. Si usamos un Mapa de Bits para la gestión del espacio libre, especifique la sucesión de bits que contendría respecto a los 18 bloques del ejercicio anterior.
  5. Si se pierde el primer puntero de la lista de espacio libre, ¿podría el Sistema Operativo reconstruirla? ¿Cómo?
  6. El espacio libre en disco puede ser implementado usando una lista encadenada con agrupación o un mapa de bits. La dirección en disco requiere D bits. Sea un disco con B bloques, en que F están libres. ¿En qué condición la lista usa menos espacio que el mapa de bits?

7. Entre los posibles atributos de un archivo, existe un bit que marca un archivo como temporal y por lo tanto esta sujeto a destrucción automática cuando el proceso acaba ¿Cuál es la razón de esto? Después de todo un proceso siempre puede destruir sus archivos, si así lo decide.
8. Algunos SO proporcionan una llamada al sistema (RENAME) para dar un nombre nuevo a un archivo existente ¿Existe alguna diferencia entre utilizar esta llamada para renombrar un archivo y copiar el archivo a uno nuevo, con el nuevo nombre y destruyendo el antiguo?
9. Un i-nodo de UNIX tiene 10 direcciones de disco para los diez primeros bloques de datos, y tres direcciones más para realizar una indexación a uno, dos y tres niveles. Si cada bloque índice tiene 256 direcciones de bloques de disco, cuál es el tamaño del mayor archivo que puede ser manejado, suponiendo que 1 bloque de disco es de 1KByte?
10. Sobre conversión de direcciones lógicas dentro de un archivo a direcciones físicas de disco. Estamos utilizando la estrategia de indexación a tres niveles para asignar espacio en disco. Tenemos que el tamaño de bloque es igual a 512 bytes, y el tamaño de puntero es de 4 bytes. Se recibe la solicitud por parte de un proceso de usuario de leer el carácter número N de determinado archivo. Suponemos que ya hemos leído la entrada del directorio asociada a este archivo, es decir, tenemos en memoria los datos PRIMER-BLOQUE y TAMAÑO. Calcule la sucesión de direcciones de bloque que se leen hasta llegar al bloque de datos que posee el citado carácter.
11. ¿Qué método de asignación de espacio en un sistema de archivos elegiría para maximizar la eficiencia en términos de velocidad de acceso, uso del espacio de almacenamiento y facilidad de modificación (añadir/borrar /modificar), cuando los datos son:
  - a) modificados infrecuentemente, y accedidos frecuentemente de forma aleatoria
  - b) modificados con frecuencia, y accedidos en su totalidad con cierta frecuencia
  - c) modificados frecuentemente y accedidos aleatoriamente y frecuentemente
12. ¿Cuál es el tamaño que ocupan todas las entradas de una tabla FAT32 que son necesarias para referenciar los cluster de datos, cuyo tamaño es de 16 KB, de una partición de 20 GB ocupada exclusivamente por la propia FAT32 y dichos cluster de datos?
13. Cuando en un sistema Unix/Linux se abre el archivo /usr/ast/work/f, se necesitan varios accesos a disco. Calcule el número de accesos a disco requeridos (como máximo) bajo la suposición de que el i-nodo raíz ya se encuentra en memoria y que todos los directorios necesitan como máximo 1 bloque para almacenar los datos de sus archivos.
14. Suponiendo una ejecución correcta de las siguientes órdenes en el sistema operativo Linux: /home/jgarcia/prog

```
$ ls -i          //lista los archivos y sus números de i-nodos del
                  directorio prog
18020          fich1.c
18071          fich2.c
18001          pract1.c

/home/jgarcia/prog > cd ../tmp
/home/jgarcia/tmp > ln -s ../prog/pract1.c p1.c //crea enlace simbólico
/home/jgarcia/tmp > ln ../prog/pract1.c p2.c   //crea enlace absoluto
```

represente gráficamente cómo y dónde quedaría reflejada y almacenada toda la información referente a la creación anterior de un enlace simbólico y absoluto ("hard") a un mismo archivo, pract1.c.

- 15.** En un sistema de archivos ext2 (Linux), ¿qué espacio total (en bytes) se requiere para almacenar la información sobre la localización física de un archivo que ocupa 3 Mbytes?. Suponga que el tamaño de un bloque lógico es de 1 Kbytes y se utilizan direcciones de 4 bytes. Justifique la solución detalladamente.
- 16.** En la mayoría de los sistemas operativos, el modelo para manejar un archivo es el siguiente:
- Abrir el archivo, que nos devuelve un descriptor de archivo asociado a él.
  - Acceder al archivo a través de ese descriptor devuelto por el sistema.
- ¿Cuáles son las razones de hacerlo así? ¿Por qué no, por ejemplo, se especifica el archivo a manipular en cada operación que se realice sobre él?
- 17.** Sea un directorio cualquiera en un sistema de archivos ext2 de Linux, por ejemplo, DirB. De él cuelgan algunos archivos que están en uso por uno o más procesos. ¿Es posible usar este directorio como punto de montaje? Justifíquelo.