

Relación Problemas Tema 4. Sistemas Operativos

- 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	NºBloque
DATOS	F	8-2-90	3

FAT

1		10	
2		11	
3	15	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:

- Creación del archivo DATOS1 con fecha 1-3-90, y tamaño de 10 bytes.
 - Creación del archivo DATOS2 con fecha 2-3-90, y tamaño 1200 bytes.
 - El archivo DATOS aumenta de tamaño, necesitando 2 bloques más.
 - Creación del directorio D, con fecha 3-3-90, y tamaño 1 bloque.
 - Creación del archivo CARTAS con fecha 13-3-90 y tamaño 2 kBytes.
- Si se pierde el primer puntero de la lista de espacio libre, ¿podría el Sistema Operativo reconstruirla? ¿Cómo?
 - 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?
 - 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.
 - 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?
6. Un i-nodo del sistemas de archivos *s5fs* 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?
 7. 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.
 8. ¿Qué organización 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
 9. Un controlador en software de disco duro recibe peticiones para los cilindros 10, 22, 20, 2, 40, 6 y 38 en ese orden. El brazo tarda 6ms en moverse de un cilindro al siguiente Suponiendo que el brazo está inicialmente en el cilindro 20 ¿cuánto tiempo de posicionamiento se requiere siguiendo los algoritmos
 1. FCFS ?
 2. SSTF ?
 3. SCAN ?
 10. ¿Tendría sentido combinar en un mismo sistema de archivos dos métodos de asignación de espacio en disco diferentes? Por ejemplo, el método contiguo combinado con el enlazado. Ponga un ejemplo y razónelo.
 11. ¿Por qué la FAT en MS-DOS y Windows tiene un tamaño fijo en disco para cada sistema de archivos?
 12. Para comprobar la consistencia de un sistema de archivos de Unix, el comprobador de consistencia (*fsck*) construye dos listas de contadores (cada contador mantiene información de un bloque de disco), la primera lista registra si el bloque está asignado a algún archivo y la segunda, si está libre. Según se muestra en la siguiente figura:

en uso: 1 0 1 0 0 1 0 1 1 0 1 0 0 1 0
libres: 0 0 0 1 1 1 0 0 0 1 0 1 1 0 1

¿Existen errores? ¿Son serios estos errores? ¿por qué? ¿qué acciones correctivas sería necesario realizar sobre la información del sistema de archivos?

13. Cuando 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. ¿El planificador de Unix favorece a los procesos limitados por E/S (cortos) frente a los procesos limitados por CPU (largos)? Explique cómo lo hace. ¿Y el planificador de Windows 2000 o NT?
15. Supongamos que un proceso, **P1**, abre el archivo "**datos**" en modo **lectura/escritura** y otro proceso, **P2**, abre el mismo archivo y con el mismo modo, y a continuación crea un proceso hijo que abre el archivo `"/usr/pepe/doc"` en modo **lectura/escritura**. Represente toda la información relevante sobre el estado de las tablas de descriptores de archivos, tabla de archivos y tabla de i-nodos después de dichas operaciones.
16. Suponiendo una ejecución correcta de las siguientes órdenes en el sistema operativo Unix:
- ```
/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 un enlace simbólico *)
/home/jgarcia/tmp > ln ../prog/pract1.c p2.c (* crea un enlace absoluto o duro*)
```
- 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**.
17. En Unix, ¿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.
18. ¿Por qué en el sistema de archivos DOS (FAT) está limitado el número de archivos y/o directorios que descienden directamente del directorio raíz y en Unix no lo está?
19. ¿Cuál es el problema que se plantea cuando un proceso no realiza la llamada al sistema **wait** para cada uno de sus procesos hijos que han terminado su ejecución? ¿Qué efecto puede producir esto en el sistema?

20. Tenemos un archivo de 4 MBytes, calcule el espacio en disco necesario para almacenar el archivo (incluidos sus metadatos) en un Sistema de Archivos FAT32 y en otro s5fs. Suponga que el tamaño del cluster en FAT32 y el de bloque lógico en s5fs es de 4KBytes.
21. 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?
22. ¿Cómo consigue optimizar FFS operaciones del tipo “ls -l” sobre un directorio? Explíquelo y ponga un ejemplo.
23. Sea un directorio cualquiera en un sistema de archivos de Unix, 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.
24. ¿Es correcto el siguiente programa en UNIX, en el sentido de que no genera inconsistencias en las estructuras de datos que representan el estado de los archivos en el sistema? Represente con un esquema gráfico la información almacenada en las estructuras (tablas de descriptors de archivos, tabla de archivos y tabla de i-nodos) que representan el estado real referente a los archivos utilizados por los siguientes procesos en el instante de tiempo en que: el *PROCESO HIJO DE A* ha ejecutado la instrucción marcada con (1), el *PADRE (PROCESO A)* ha ejecutado la (2), y el *PROCESO B* ha ejecutado la (3). Para ello suponga que son los únicos procesos ejecutándose en el sistema en ese momento.

```
/* PROCESO A */
main(){
 int archivos[2], pid;
 archivos[0]=open("/users/prog.c",O_RDONLY);
 archivos[1]=open("/users/prog.c",O_RDWR);
 if ((pid=fork())==-1)
 perror("Ha ocurrido un error en la creación del proceso);
 else if (pid==0)
 execlp ("ls", "ls", NULL); /*(1)*/
 else
 close(archivos[0]); /*(2)*/
}
/* PROCESO B */
main(){
 int archivos[2];
 archivos[0]=open("/users/prog.c",O_RDONLY);
 archivos[1]=open("/users/texto.txt",O_RDWR);
 dup (archivos[0]); /* (3) */
}
```