FACULTAD DE INFORMATICA

SISTEMAS OPERATIVOS

3º de Informática.

PROBLEMAS SOBRE SISTEMAS DE FICHEROS

1. Calcular el número de accesos a disco necesarios para leer 20 bloques lógicos consecutivos (no necesariamente los 20 primeros bloques) de un fichero en un sistema con:
   a) Asignación contigua
   b) Asignación no contigua encadenada
   c) Asignación no contigua indexada
   Explicar las diferencias entre los accesos a bloques lógicos y físicos.

2. 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.

   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.

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

FAT

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

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.

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.

3. Considerar un fichero que consta de 100 bloques. ¿Cuántas operaciones de disco son necesarias para cada una de las tres estrategias de asignación (contigua, encadenada 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.

4. 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.

5. Tenemos un sistema de ficheros sobre un disco con bloques de 512 palabras (1 palabrea = 4 bytes). Para cada una de las tres estrategias de asignación, las entradas de un directorio del sistema de ficheros son las siguientes:

Asignación contigua	Nom. Fichero	Pos. 1er bloque datos		Long. Fichero
Asignación encadenada	Nom. Fichero	Pos.1er bloque datos	Pos. últ. bloque	Long. Fichero
Asignación indexada	Nom. Fichero	Pos. bloque de índices		Long. Fichero

   Para la asignación indexada, una entrada de un directorio apunta al primer bloque de índices del fichero correspondiente, el cual a su vez apunta a 511 bloques y un puntero al siguiente bloque de índices. Para cada una de las tres estrategias de asignación:
   a. Explicar cómo se realiza la traducción de la posición del bloque lógico a la posición del bloque físico en este sistema.
   b. Si estamos actualmente en el bloque lógico 10 (el último bloque accedido fue el bloque 10) y queremos acceder al bloque lógico 4, ¿cuántos bloques físicos deben ser leídos de disco?

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?

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) si 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?.

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, enn un sistema de ficheros de tipo UNIX.

12. Dos estudiantes de informática, AA y BB, sostienen una discusión respecto a los i-nodos. AA 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. BB discrepa. ¿Quién tiene razón?

13. La llamada al sistema **int link(char *_nombre1_, char *_nombre2_)** crea un enlace rígido (*hard link*) entre el fichero existente ***nombre1*** y el nuevo apelativo ***nombre2***. Describir en seudo-código el trabajo del sistema operativo para dar cumplimiento a la llamada **link**.

14. Sea un servicio de ficheros, similar al de UNIX, con las siguientes características:
    - Representación de ficheros mediante nodos-i con 10 bloques directos y direccionamiento de bloques indirectos mediante números de 4 bytes.
    - Cache para 16 bloques con política de gestión LRU.

    Sobre un sistema de ficheros que maneja tamaños de bloque de 4 Kbytes y cuyo primer bloque de datos es el 10, se quiere programar una aplicación que realice la siguiente secuencia de operaciones:
    1.- creación de un fichero de nombre "prueba",
    2.- escritura completa de 34 bloques de dicho fichero,
    3.- lectura directa, con posicionamiento al principio del fichero y lectura bloque a bloque hasta el final del mismo y, por último,
    4.- lectura inversa bloque a bloque desde el final al principio del fichero.
    Se pide:
    a) Escribir el programa descrito indicando claramente las llamadas al sistema de ficheros.
    b) Para cada una de las funciones escritura completa, lectura directa y lectura inversa, escriba una traza de los bloques que se acceden
    c) Para cada una de las funciones escritura completa, lectura directa y lectura inversa, calcule el número de fallos de bloque en la cache.

15. Un sistema de ficheros similar al de UNIX, presenta las siguientes características:
    - Representación de ficheros mediante nodos-i con 12 direcciones directas a bloque, un indirecto simple, un indirecto doble y un indirecto triple y direcciones de bloques de 4 bytes. El tamaño del bloque del sistema de ficheros es de 8 KB.
    - El sistema de ficheros emplea un cache de 4 MB con una política de reemplazo LRU.

    Sobre este sistema de ficheros se ejecuta el siguiente fragmento de código:

```
#define DATA_SIZE     2048
#define N             5 * 1024
char buffer[DATA_SIZE];
int fd, i;

fd = open("fichero", O_RDWR);
for (i = 0 ; i < N; i++)
    write(fd, buffer, DATA_SIZE);
lseek(fd, 0, SEEK_SET);
for (i = 0 ; i < N; i++)
    read(fd, buffer, DATA_SIZE);
close(fd);
```

    Teniendo en cuenta que el fichero tiene un tamaño inicial de 16 MB, que la cache se encuentra inicialmente vacía y que no se realiza ninguna operación de volcado (flush) durante el mencionado programa, se pide:
    a) ¿Cuántos accesos a disco se producen durante los bucles de lectura y escritura utilizando una política de actualización write-through (escritura inmediata)?
    b) ¿Cuántos accesos se producirían si se utilizase una política de actualización write-back (escritura diferida)?
    c) Calcule la tasa de aciertos a la cache que se produce durante el bucle de lectura.
    d) Calcule en función de N, siendo N la constante definida en el programa, la tasa de aciertos a la cache que se produce durante el bucle de lectura. Supóngase que como mucho se accede los 16 MB que ocupa el fichero.
    NOTA: Considere que no se actualiza ninguna de las fechas que aparecen en el nodo-i del fichero hasta que se realiza la operación close().

16. Sea un sistema de ficheros similar al de UNIX con un tamaño de bloque de 4K y un nodo-i con 10 punteros directos, 1 indirecto simple, 1 doble y 1 triple. Sin embargo, a diferencia del sistema de ficheros de UNIX, este sistema usa write-through en todas las operaciones de escritura en el disco, tanto para la metainformación como para los propios datos. Se pretende analizar en este sistema qué zonas de una partición son actualizadas por las distintas llamadas al sistema. Para ello se considerarán las siguientes zonas:

- Mapa de bloques libres (**MB**)
- Mapa de nodos-i libres (**MN**)
- Bloques con nodos-i (**BN**)
- Bloques de datos (**BD**)

Dado el siguiente fragmento de programa:

```
mkdir("/dir", 0755);              /* llamada 1 */
fd=creat("/dir/f1", 0666);        /* llamada 2 */
write(fd, buf, 4096);             /* llamada 3 */
lseek(fd, 40960, SEEK_SET);       /* llamada 4 */
write(fd, buf, 4096);             /* llamada 5 */
close(fd);                        /* llamada 6 */
symlink("/dir/f1","/dir/f2");     /* llamada 7 */
unlink("/dir/f1");                /* llamada 8 */
```

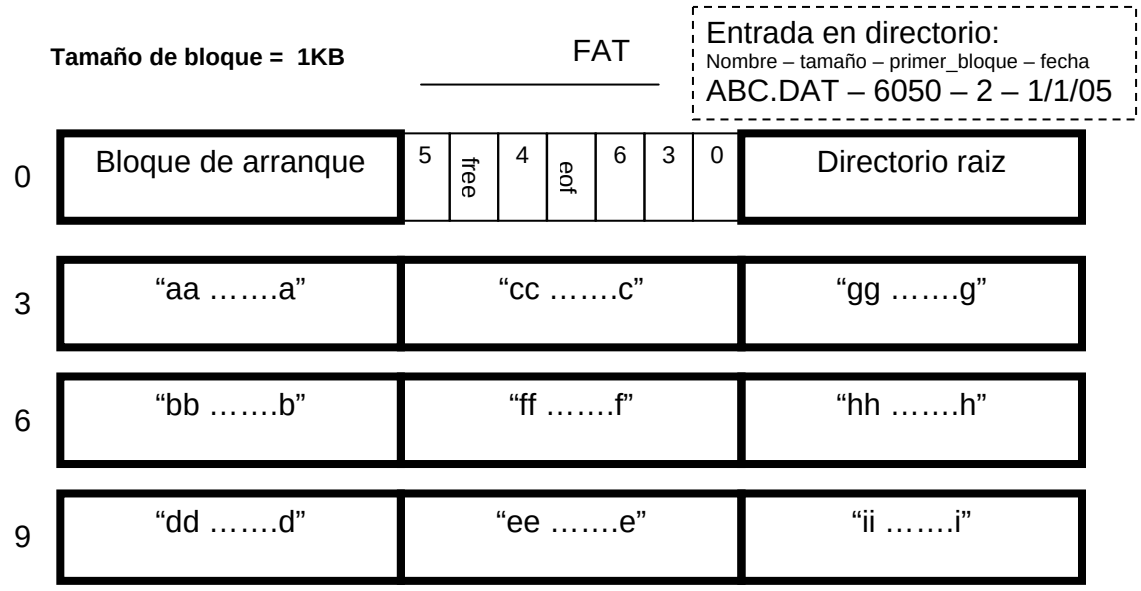
Suponiendo que no se produce ningún error en la ejecución de dicho fragmento, se pide especificar qué zonas son actualizadas en cada llamada explicando razonadamente el motivo de dicha actualización.

17. Dado un disco de 10 MB con sectores de 512B y geometría de 512 cilindros, 4 cabezas y 10 sectores por pista, determinar las direcciones CHS de los sectores involucrados en las operaciones de acceso al disco que resultan de la ejecución de las siguientes sentencias,

```
lseek(fd, 20, PRINCIPIO);
read(fd, buffer, 15);
```

cuando el fichero de descriptor `fd` está organizado como registros de longitud fija de 80B, y los argumentos de las llamadas están expresados en unidades de registro. La dirección LBA del primer bloque del fichero es 356 y se usa asignación contigua.

18. Para las siguientes operaciones **read()** y **write()** sobre el fichero *ABC.DAT*, determinar qué datos se obtienen (leen) y cuáles se modifican (escriben). Reseñar todos los cambios, incluyendo aquéllos que puedan ocurrir algo más tarde si se usa escritura diferida. Usar el sistema de ficheros FAT de la figura.
    i) read() 55 bytes desde la posición 1010
    ii) read() 200 bytes desde la posición 4000
    iii) read() 2000 bytes desde la posición 4400
    iv) write() 100 bytes con “xx…x” a partir de la posición 5060
    v) write() 200 bytes con “yy…y” a partir de la posición 6080



Repetir el ejercicio aplicándolo al fichero cuyo inodo es 5 en el sistema de ficheros de la figura siguiente: