Diseño de sistemas operativos. Febrero de 2001. Ejercicio 2

Enunciado

Los dispositivos de almacenamiento óptico de tipo ***CD*** suelen tener una capacidad de 650 ó 700Mb. Pueden ser ***CDROM*** (si vienen escritos de fábrica), ***CD-R****ecordable* (si pueden ser escritos un única vez) o ***CD-ReWrite*** (si se pueden reescribir múltiples veces, previo borrado de su contenido).
Con cada bloque de datos que se escribe, sucede automáticamente el formateo a bajo nivel del medio: se graba una etiqueta que identifica el número de bloque, luego se graba el bloque de datos en sí (2Kb) y finalmente un código corrector de errores.

```
[Etiqueta] [...2Kb de datos...] [Redundancia]
```

De esta manera, cada vez que se lee un bloque de datos, sucede automáticamente la verificación de que se está leyendo dicho bloque y de que su contenido es correcto.

Estos dispositivos se caracterizan porque no permiten la sobre-escritura, no se pueden "machacar" los datos de un bloque con nueva información. El espacio del disco que no ha sido usado todavía sigue virgen, y puede ser escrito, pero ningún espacio ya escrito puede ser sobre-escrito. Se empieza a escribir por el primer bloque virgen y se sigue en orden ascendente. Son dispositivos de escritura incremental (*append-only*) y de lectura aleatoria.

Para crear un sistema de ficheros sobre uno de estos dispositivos se usa un programa *creador*:

1ª) Le indicaremos un directorio, cuyo contenido (conjunto de ficheros y/o subdirectorios), deseamos grabar en el dispositivo.
2ª) Recolecta la información necesaria para determinar la imagen que el sistema de ficheros deberá tener en el dispositivo.
3ª) Finalmente escribe dicha imagen en dispositivo, como una secuencia de bloques.

Se desea que diseñe una versión del Sistema de Ficheros UNIX (UFS), que denominaremos CDFS, adaptada a su uso sobre dispositivos de escritura incremental y lectura aleatoria.

Se pide:

A) Dibuje la estructura del UFS convencional. Para cada una de sus partes explique: cuál es su contenido y utilidad y de qué factores depende su tamaño.
B) Para cada una de las partes del UFS convencional, razone la necesidad de que exista o no en el CDFS.
C) Dibuje la estructura del CDFS. Para cada una de sus partes explique: cuál es su contenido y de qué factores depende su tamaño.
D) Razone qué información necesita recolectar el programa *creador* en su 2ª fase para poder determinar la imagen del sistema de ficheros a grabar.
E) ¿Cuál es la política de ubicación más óptima cabe aplicar? ¿Cómo se verá afectada la estructura del nodo-i?

Apartado EXTRA. ¡Sólo si le sobra tiempo!

Imagine que queremos poder volver a usar el programa *creador* sobre un dispositivo en el que ya anteriormente creamos un CDFS, para grabar en él sólo las alteraciones (borrados y/o añadidos) que haya sufrido la información en él contenida. A esto se le denomina **multisesión**. El dispositivo contendrá todas las sesiones, pero al ser montado para su uso normal sólo será visible la última.

F) Sugiera qué modificaciones realizaría al CDFS para que soportase multisesión.

Solución

A) En la estructura de un Sistema de Ficheros UNIX convencional se pueden distinguir las siguientes partes:

[BB] [SB] [MBB] [MBI] [Inodos] [Bloques de datos]

BB, Bloque de *boot*.- En todo dispositivo orientado a bloque y en toda partición de dispositivo orientado a bloque que se utilice para contener un sistema de ficheros, aparecerá este primer campo. Está destinado a contener código ejecutable para, si se desea, poder arrancar la máquina desde esta partición. Su tamaño es de un bloque.

SB, Superbloque.- Todo sistema de ficheros tipo UNIX tiene esta estructura como segundo bloque (después del bloque de *boot*). Esta estructura es la que, entre otras cosas, identifica al sistema de ficheros tipo UNIX como tal, a través de un número mágico o similar. Asimismo contiene información de detalle sobre la disposición y tamaño del resto de las partes (campos) del sistema de ficheros. Sin el superbloque, el sistema de ficheros no puede ser reconocido como tal, ni puede ser manejado convenientemente. Por esta razón es común que existan varías copias del mismo dispersas por la partición en puntos preestablecidos. Su tamaño es, como su nombre indica, un bloque, que en su mayor parte estará vacío.

MBB, Mapa de bits de bloques.- Esta estructura de datos es tratada como un vector de bits donde cada bit informa sobre el estado de ocupación (libre u ocupado) de cada bloque del dispositivo. Realmente sólo tendría que informar sobre los bloque que componen el campo "Bloques de datos" y no sobre todo el dispositivo, pero esta optimización no suele realizarse.
Precisa un bit por bloque, luego su tamaño dependerá del número total de bloques del dispositivo, o lo que es lo mismo, del tamaño del dispositivo y del tamaño del bloque. Finalmente su tamaño se redondeará por exceso a un número entero de bloques.

MBI, Mapa de bits de inodos.- Al igual que la estructura anterior, se trata de un vector de bits, donde cada bit informa sobre el estado de

ocupación (libre u ocupado) de cada inodo del sistema de ficheros.
Precisa un bit por inodo, luego su tamaño dependerá del número total de inodos del sistema de ficheros (véase el apartado siguiente). Finalmente su tamaño se redondeará por exceso a un número entero de bloques.

Inodos.- Este campo equivale a un vector de inodos, es decir, contiene todos los inodos del sistema de ficheros uno detrás de otro.
Cada **inodo** contiene toda la información relativa a un objeto (fichero, directorio, fichero especial, etc., etc.) del sistema de ficheros, exceptuando su nombre. En general en el inodo se pueden distinguir dos partes, atributos y localización.

Atributos.- Son todos los que caracterizan al objeto en cuestión (tipo de objeto, permisos, propietario, grupo, tamaño, número de nombres, fechas de creación, modificación y acceso).

Localización.- Son las direcciones de los bloques ocupados por el fichero (objeto en general) en el sistema de ficheros. Es una tabla indexada, cuyo nivel de indexación aumenta con el tamaño del objeto. Dispone de una serie de direcciones directas (alrededor de 10) y de otras tres entradas de nivel de indexación creciente (simple, doble y triple indirecta) que apuntan a otros bloques de datos (que denominamos bloques indirectos) usados para contener direcciones de bloque.

El tamaño del campo de inodos dependerá evidentemente del tamaño del inodo (64 o 128 bytes) y del número de inodos del sistema de ficheros. Este parámetro se establece en el momento de la creación del sistemas de ficheros, y normalmente se escoge un inodo por bloque, considerando este caso como el de máximo número de ficheros posible.

Bloques de datos.- Este campo abarca el resto del espacio disponible. Está destinado a contener los bloques que componen el contenido de los ficheros y directorios (objetos en general) así como aquellos bloques indirectos usados para contener direcciones de bloque de que permiten la localización del contenido de los ficheros de gran tamaño.

B) En un Sistema de Ficheros de tipo CDFS, dada su naturaleza de escritura incremental, es preciso preestablecer la imagen del sistema de ficheros (2ª fase del *creador*) antes de estamparla en el dispositivo (3ª fase), con lo cuál el espacio del disco va a ir siendo consumido en estricto orden creciente (no de forma aleatoria) y por lo tanto, no todas las partes mencionadas en el punto A tendrán porqué existir.

BB, Bloque de *boot*.- Como se mencionó anteriormente, debe aparecer (aunque no vaya a usarse) en todo dispositivo orientado a bloque y en toda partición de dispositivo orientado a bloque que se utilice para contener un sistema de ficheros.

SB, Superbloque.- Es imprescindible para identificar al sistema de ficheros tipo UNIX como tal. En nuestro caso, indicará que se trata de un CDFS. Asimismo contendrá información de detalle sobre la disposición y tamaño del resto de las partes (campos) del sistema de ficheros. También podría pensarse en mantener varias copias del mismo dispersas por la partición en puntos preestablecidos.

MBB, Mapa de bits de bloques.- No existirá en el CDFS. Su objetivo es conocer qué bloques de datos están libres y cuáles ocupados. Esta información quedará resumida en nuestro caso indicando en un campo del superbloque, el número del primer bloque todavía virgen del dispositivo tras estampar la imagen del sistema de ficheros en disco.

MBI, Mapa de bits de inodos.- No existirá en el CDFS. No la necesitaremos para conocer el estado de ocupación de los inodos del sistema de ficheros. Todos los inodos que hayamos precisado estarán en uso y no dejarán de estarlo bajo ninguna circunstancia dado que no se puede reescribir información. Esta información quedará resumida en nuestro caso indicando en un campo del superbloque, el número de inodos usados en el sistema de ficheros.

Inodos.- Este campo seguirá existiendo. Seguimos necesitando la información sobre atributos y localización relativa a cada objeto (fichero, directorio, fichero especial, etc., etc.) existente en el sistema de ficheros.

Atributos.- Puede que alguno de sus campos carezca ahora de utilidad: fechas de modificación y acceso, pero en general se mantendrá inalterado.

Localización.- Dado que la ubicación de cada fichero se preestablece en la 2º fase del *creador*, la disposición natural de los bloques de datos de cada fichero será aquella en que se hallen contiguos uno detrás de otro, en orden. Por lo tanto, no necesitaremos el conjunto de direcciones de bloque directas e indirectas. Bastará con conocer la dirección del primer bloque asignado al fichero y su tamaño (que forma parte de los atributos del inodo) para poder localizar cualquier bloque. Esto contesta a la pregunta E.

Bloques de datos.- Evidentemente sigue siendo necesario el espacio destinado a contener los bloques que componen el contenido de los ficheros y directorios (objetos en general).

C) La estructura de un Sistema de Ficheros CDFS se distinguirán las siguientes partes:

[BB] [SB] [Inodos] [Bloques de datos] [Resto virgen...]

BB, Bloque de *boot*.- Contenido y tamaño idénticos a los del apartado A.

SB, Superbloque.- Indicará que se trata de un CDFS y contendrá información de detalle sobre la disposición y tamaño del resto de las partes (campos) del sistema de ficheros. En concreto, contendrá la información que nos permite prescindir de los bitmap de bloques e inodos: un campo con el número del primer bloque todavía virgen y otro campo con el número de inodos usados en el sistema de ficheros. Su tamaño será un bloque que en su mayor parte estará vacío.

Inodos.- El contenido de este campo se ha descrito en el apartado B. El tamaño del campo de inodos depende del tamaño del inodo (que ahora podrá ser menor) y del número de inodos, que en el caso del CDFS no es un máximo prefijado, sino el número concreto de objetos que **existan** en el sistema de ficheros.

Bloques de datos.- Este campo contendrá los bloques que componen el contenido de los ficheros y directorios (objetos en general) que **existan** en el sistema de ficheros. Por lo tanto no abarcará el resto del espacio disponible sino sólo lo estrictamente necesario.

Resto virgen.- A diferencia de un UFS convencional, en este caso, la parte del dispositivo que no haya sido usada todavía aparecerá como tal, y se corresponderá con el espacio todavía virgen del dispositivo.

Los dispositivos de almacenamiento óptico de tipo CD suelen tener una capacidad de 650 ó 700Mb

D) El programa *creador* en su 2ª fase debe determinar la imagen del sistema de ficheros a grabar. Para ello deberá recolectar información sobre todos los objetos a copiar. Por lo tanto deberá recorrer recursivamente, en profundidad, el directorio que se le indica como argumento e ir tomando nota de los atributos de cada objeto que encuentre, así como del contenido de los directorios, esto es de los nombres de los objetos y de su jerarquía.
Simultáneamente a este recorrido deberá ir determinando la disposición que cada objeto tendrá en la imagen del sistema de ficheros que se está construyendo.
En resumen, se recoge toda la información, exceptuando la disposición física de la misma sobre el sistema de ficheros original (que nos da igual) y quizás también el contenido de los ficheros (no así de los directorios) que puede ser leído al vuelo posteriormente en la 3ª fase.

E) La política de ubicación es la que determina la disposición de la información sobre el medio. Lo más óptimo en nuestro caso, y también lo más natural dada la naturaleza de este tipo de dispositivos va a ser la ubicación secuencial (o contigua) de todos los bloques de cada fichero, esto es, los bloques lógicamente contiguos de cada fichero (objeto en general) serán bloques físicamente contiguos.
Como ya se indicó en el apartado B, esto afecta al contenido del inodo en lo que respecta a la información sobre ubicación. Nos bastará con tener un único puntero que nos indique el número del primer bloque del fichero, y su tamaño.

F) Como se deduce del enunciado, la imagen de cada sesión sería escrita a continuación de la imagen de la anterior. De esta manera cuando el sistema operativo se disponga a montar nuestro sistema de ficheros CDFS empezará por leer y analizar el primer superbloque (correspondiente a la primera sesión) pero no debería contentarse con esto ya que pueden existir otras sesiones escritas con posterioridad.
Deberá inspeccionar la zona indicada en cada superbloque como primer bloque virgen, para comprobar si efectivamente sigue virgen, en cuyo caso estaremos sobre la última sesión del dispositivo, o no está virgen, sino que contiene el superbloque de otra sesión más actual. La última sesión es la que debe ser montada.
Por lo demás, para poder soportar la modificación y borrado de ficheros, lo más cómodo es volver a repetir la información de todos los inodos del sistema de ficheros, reutilizando el espacio asignado de los objetos que continúen vivos e inalterados y escribir versiones nuevas completas de todo los que haya variado.