Diseño de Sistemas Operativos
Febrero de 2003
2º Problema

ENUNCIADO

Recientemente, los científicos de IBM han demostrado una nueva tecnología de almacenamiento no volátil (semejante a la memoria flash) denominada **Millipede** que, gracias a la nanotecnología, es capaz de almacenar un Terabit en un componente del tamaño de un sello postal. Esto supone una densidad de almacenamiento 20 veces superior a la máxima alcanzada con tecnología de almacenamiento magnético. Además, las partes móviles de este componente son tan diminutas que permite considerar que el tiempo de acceso al medio será uniforme. Sin embargo, las velocidades de transferencia previstas son del orden de 2 Megabits por segundo, esto es, muy inferiores a los 160 Megabytes por segundo de un disco duro actual.

No es descabellado imaginar que dentro de poco tiempo un PC doméstico pueda contener un dispositivo (basado en la tecnología descrita) de dimensiones e interfaz equivalentes a un disco duro actual, pero con una capacidad de 8 Terabytes y una velocidad de transferencia de 8 Megabytes por segundo.

Suponga que estudiamos la viabilidad de utilizar un Sistema de Archivos convencional (tipo UFS) con nodos-i de 128 bytes y direcciones de bloque de 32 bits, sobre un dispositivo como el presentado.

1. Calcule cuánto espacio ocuparían los metadatos (bitmaps, nodos-i, etc.) de este Sistema de Archivos. Haga las suposiciones que considere necesarias. Exprese el resultado en la magnitud más significativa (kilo, mega, giga, tera,...) de bytes.

2. El uso de los sistemas de almacenamiento sigue la ley de los gases nobles: tiende ocuparse todo el espacio disponible. Suponiendo que para una ocupación del 80%, el espacio ocupado por directorios y bloques indirectos de distintos niveles equivale a 7 veces el espacio ocupado por los metadatos, ¿cuánto tiempo se tardaría en leer del dispositivo toda esta información (metadatos, directorios y bloques indirectos)?

3. En vista de los resultados obtenidos en los apartados anteriores, razone porqué un Sistema de Archivos convencional (tipo UFS) **no** es adecuado para un dispositivo como éste.

4. ¿Qué técnicas debería utilizar un Sistema de Archivos para ser adecuado para un dispositivo como éste? Describa en qué consisten las mismas.

5. Suponiendo un Sistema de Archivos con las características del punto anterior, describa cómo se realizarían las acciones siguientes indicando claramente las diferencias respecto al caso de un Sistema de Archivos convencional:
   - Borrado de un directorio vacío.
   - Montaje del dispositivo y verificación si procede.

SOLUCIÓN

1. (Peso: **3**/10)

Tamaño del bloque

El dispositivo es de 8 TB = 2^{43} bytes. Dado que las direcciones a bloque son de 32 bits, el número **máximo** de bloques en el dispositivo es de 2^{32}. Por lo tanto el tamaño **mínimo** de bloque será:

$$2^{43}\ _{(\text{bytes})} / 2^{32}\ _{(\text{bloques})} = 2^{11}\ _{(\text{bytes/bloque})} = 2\ \text{KB}$$

Por otro lado, con bloques de 2 KB y con direcciones de 4 bytes (32 bits), caben:

$$2^{11}\ _{(\text{bytes/bloque})} / 2^{2}\ _{(\text{direcciones/byte})} = 2^{9}\ _{(\text{direcciones/bloque})}$$

Y si consideramos un nodo-i estándar, con hasta triple indirecto, entonces el mayor fichero direccionable ocuparía

algo más de la treintaidosava parte del dispositivo:

$(2^{3*9} + 2^{2*9} + 2^{1*9} + 10)_{(bloques)} >= 2^{27}_{(bloques)} = 2^{38}_{(bytes)}$

De forma semejante y para bloques de 4 KB, se podrá direccionar ficheros de más de $2^{30}_{(bloques)} * 2^{12}_{(bytes/bloque)} = 2^{42}_{(bytes)}$ o más de la mitad del dispositivo.

Para que un fichero pudiese ocupar la totalidad del dispositivo sería preciso usar bloques de 8 KB.

Cabría argumentar que, dadas las dimensiones del dispositivo, puede resultar más efectivo que el bloque sea de tamaño mayor, ya que aunque aumenta la fragmentación interna, el espacio ocupado por los metadatos se reducirá proporcionalmente.

Para el resto del problema supondremos bloques de 8 KB.

Tamaño de los metadatos

Para calcular el espacio ocupado por los metadatos, consideraremos despreciable el espacio ocupado por los dos primeros bloques: boot (**BB**) y superbloque (**SB**).

Así, el dispositivo consistirá en:

$2^{43}_{(bytes)} / 2^{13}_{(bytes/bloque)} = 2^{30}_{(bloques\ de\ 8KB)}$

Por lo tanto, el bitmap de bloques (**BMB**) ocupará:

$2^{30}_{(bits)} / 2^{3}_{(bits/byte)} = 2^{27}_{(bytes)} = 128\ MB$

Suponiendo un nodo-i por bloque (máximo número de ficheros de tamaño mínimo), el bitmap de nodos-i (**BMI**) ocuparía lo mismo, 128 MB.

Por último el vector de nodos-i (**VI**) ocupará:

$2^{30}_{(nodos\text{-}i)} * 2^{7}_{(bytes/nodo\text{-}i)} = 2^{37}_{(bytes)} = 128\ GB$

Por lo tanto el espacio ocupado por metadatos será algo más de 128 Gigabytes.

2. (Peso: **1**/10)

Según se indica en el enunciado, el espacio ocupado por directorios y bloques indirectos equivale a 7 veces el ocupado por los metadatos, luego el tiempo necesario para leer esta información nunca será menor de:

$(7+1) * 2^{37}_{(bytes)} / 2^{23}_{(bytes/segundo)} = 2^{(3+37-23)} = 2^{17}_{(segundos)} >= 1{,}5\ días$

3. (Peso: **2**/10)

El tiempo obtenido en el apartado anterior parece enorme, pero, ¿cuándo es preciso gastar este tiempo en leer toda esta información?, o lo que es igual, ¿cuándo se hace necesario leer del sistema de archivos todo excepto los datos de los ficheros?

La respuesta es: **durante la verificación del sistema de archivos**.

El sistema de archivos UFS es frágil porque es propenso a corromperse si no es desmontado correctamente, y en tal caso, es imprescindible realizar una verificación completa de la integridad de toda la estructura del sistema de archivos.

Por lo tanto, el UFS no resulta apropiado para un dispositivo tan grande pero tan lento, porque no es razonable que tras una caída del sistema, nuestro PC tarde más de un día y medio en recuperar la integridad del sistema de archivos antes de arrancar.

4. (Peso: **2**/10)

La técnica denominada ***journaling*** utilizada en diversos sistemas de archivos modernos: JFS, ext3fs o Reiser, y llevada al extremo en el caso del Log-Structured File System, permite conseguir sistemas de archivos memos frágiles, y mucho más rápidos a la hora de la verificación de integridad.

Esta técnica se basa en una escritura ordenada y cuidadosa de todas las alteraciones de la estructura del sistema de archivos. Para ello se utiliza un archivo (almacenado también en el disco) como *journal* o *log*, donde se anotan todas las operaciones antes de ser realizadas. De esta forma, esta lista de intenciones sirve de soporte parra conseguir que el sistema pase de forma transaccional (todo o nada) de un estado coherente a otro.

5. (Peso: **2**/10)

El borrado de un directorio vacío, en un sistema de archivos convencional, exige:

1. Decodificar el nombre de directorio indicado, hasta alcanzar su nodo-i.
2. Verificar que, efectivamente, es un directorio y está vacío.
3. Liberar sus bloques de datos, marcándolos como libres en el BMB.
4. Liberar el nodo-i que ocupa, marcándolo como libre en el BMI.
5. Eliminar la entrada de su directorio padre que le apuntaba.
6. Reducir en uno el número de enlaces del nodo-i de su directorio padre.
7. Actualizar el tiempo de última modificación del nodo-i del directorio padre.

En un sistema con *journaling*, todas estas operaciones se realizarían igualmente, pero como si de una transacción se tratase, esto es:

- `Transaction_Begin(&tid)`
- `...sub operaciones...`
- `Transaction_Commit(tid)`
- `Transaction_End(tid)`

En el montaje de un sistema de archivos convencional, se comprueba que el sistema fue desmontado correctamente, y si no fue así, se realiza una verificación exhaustiva de integridad. En el caso de un sistema con *journaling* si el sistema se cae por cualquier causa sin que el sistema de archivos haya sido desmontado, para devolver el sistema a un estado coherente, no habrá que verificar el sistema de archivos completo, sino sólo habrá que comprobar las operaciones indicadas en el fichero *journal* que contiene la lista de intenciones. Para cada operación apuntada y no completada, si llegó al *commit*, se completará, y si no se descartará, deshaciendo cualquier acción parcial que se haya realizado.