

Diseño de Sistemas Operativos. Febrero de 2000.

Ejercicio 2.

Enunciado

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 bloque indirecto simple, uno doble y uno triple.
- El tamaño de bloque es de 4 Kbytes. Las direcciones de bloques se especifican con 4 bytes.
- Cache de bloques de 2 Mbytes y política de reemplazo LRU. Inicialmente está vacía.
- Disco con política CSCAN. Tiempo de búsqueda entre pistas 4 ms. Tiempo de latencia rotacional 4 ms. Velocidad de transferencia 4 Mbytes/segundo. Capacidad 4 Gbytes.

Suponiendo que en este sistema, existe un archivo, de nombre **pepe**, ya creado con 16 Mbytes de datos, se ejecuta el siguiente fragmento de código:

```
#define DATASIZE 4096
#define N 2048
char buffer[DATASIZE];
int fd, i;

fd = open ("pepe", O_RDWR);
for (i=0; i < N; i ++){
    read (fd, buffer, DATASIZE);
    for (i=0; i < N; i ++){
        read (fd, buffer, DATASIZE);
        lseek (fd, 0, SEEK_SET); /* Al principio */
        for (i=0; i < N; i ++){
            write (fd, buffer, DATASIZE);
        }
    }
}
```

Se pide contestar a las siguientes preguntas:

- a) Calcule la tasa de aciertos en la cache durante la ejecución del programa.
- b) Si los datos del archivo **pepe** están contiguos en el disco, cada pista del disco almacena 64 Kbytes, las cabezas del disco está inicialmente en la pista 0 y el primer bloque del archivo está en la pista 12. ¿Cuál será el tiempo de acceso a disco para este programa? Incluya también la transferencia.

Solución.

Antes de hacer la solución vamos a hacer dos suposiciones que son lógicas:

- El tiempo de volver a la pista inicial es 4 mseg.
- Los bloques de la pista están intercalados de forma que se pueda leer la pista completa sin esperar la latencia para cada bloque.

- a) ¿Cuántos accesos a disco se ejecutan en el código anterior, suponiendo que se usa una política de escritura diferida (write back) en la cache?

Los accesos necesarios son los siguientes:

open

- 1 para traer el bloque del directorio en que está pepe.
- 1 para traer su nodo-i.

Total accesos para open = 2

read

- 12 para los bloques directos del nodo-i.
- 1 para el bloque indirecto simple.
- 1024 bloques de datos.
- 1 indirecto doble.
- 1 para el bloque indirecto simple de segundo nivel.
- 1024 bloques de datos.
- 1 para el bloque indirecto simple de segundo nivel.
- 1024 bloques de datos.
- 1 para el bloque indirecto simple de segundo nivel.
- 1012 bloques de datos.

Total accesos para lectura = 4101

write

- 1536 bloques de datos
- 1 bloque indirecto simple
- 1 bloque indirecto doble y 1 bloque indirecto simple
- 2 acceso para escribir victimas reemplazadas
- 1 para escribir el nodo-i en el close o el bloque víctima sustituido.

Total accesos para escritura = 1541

Total accesos a disco= 5642

b) **Calcule la tasa de aciertos en la cache durante la ejecución del programa.**

Accesos totales a la cache = $1+12+1024*2+3060*3+1024*2+12+1012*3 = 16.337$

Donde:

- * 3 se debe a los accesos a bloque doble indirecto, simple indirecto y bloque de datos.
- * 2 se debe a los accesos a bloque simple indirecto y datos

Accesos con éxito en la cache = $16.337 - \text{Total accesos a disco} = 16.337 - 5.642 = 10.695$

Tasa de aciertos = $10695 / 16337 = 0,66$ Es decir 66 %

c) **Si los datos del archivo pepe están contiguos en el disco, cada pista del disco almacena 64 Kbytes, las cabezas del disco está inicialmente en la pista 0 y el primer bloque del archivo está en la pista 12. ¿Cuál será el tiempo de acceso a disco para este programa? Incluya también la transferencia.**

Cada pista del disco tiene 64 Kbytes. Puesto que el archivo origina 5655 accesos a disco para leer el archivo y cada bloque es de 4 Kbytes, el número de pistas a leer es de:

$\text{Pistas} = 5655 * 4 \text{ Kbytes} / 64 \text{ Kbytes} = 354$

Puesto que el tiempo de movimiento entre pistas es de 4 mseg., el tiempo de acceso se descompone de la siguiente forma:

- $4 * 12 = 48$ mseg. Para ir a la pista inicial del archivo (12)
- $4 + 4 + 64 \text{ Kbytes} / 4096 \text{ Kbytes} = 4 + 15,6 \text{ mseg.} = 19,6 \text{ mseg.}$ Para leer cada pista durante el bucle de lectura y moverse a la siguiente. Como hay que leer 256 pistas, el total es de 6041,6 mseg.
- 4 mseg. Para ir al principio del archivo
- $4 * 12 = 48$ mseg. Para ir a la pista inicial del archivo (12)
- $4 + 4 + 15,6$ por pista para transferencia de escritura, latencia y movimiento. Por 98 pistas que se escriben, el total de escritura es de 2312,8

Luego el tiempo total de acceso al disco es de 8, 6 segundos aproximadamente.