

EJERCICIO 1 [4puntos]. Supondremos que todas las entradas de un directorio caben en un solo bloque. Estamos situados en un determinado directorio como directorio de trabajo actual, ya tenemos su inodo y bloque de datos en memoria debido a operaciones anteriores.

a) En el intérprete de órdenes ejecutamos **ls**. ¿Cuántos accesos a disco se necesitarán para realizar esta orden?

No se necesita ningún acceso adicional a disco puesto que tenemos el bloque de datos del directorio, y en él ya están los nombres de sus hijos que es lo que muestra la orden **ls**.

b) Ejecutamos **ls -l**. ¿Cuántos accesos a disco se necesitarán para realizar esta orden?

Se necesita un acceso a disco por cada hijo del directorio actual para leer el inodo, pues con **-l** la orden **ls** muestra metadatos de cada hijo

EJERCICIO 2 [4puntos]. De dos razones por las cuales en el sistema de archivos ext2 se minimiza el movimiento de cabezas de lectura/escritura en el acceso a un archivo en disco:

Porque los datos de un archivo se almacenan en el mismo grupo de bloques, y por otro lado, también el inodo está en el mismo grupo de bloques por tanto metadatos y datos están próximos.

EJERCICIO 3. [4puntos]. Considere un sistema de archivos cuyos bloques son de 1KBytes y cada puntero a bloque requiere 4B. Sea el archivo A1 de 3GB. (B=Byte)

Calcular cuántos accesos a disco serán necesarios para acceder únicamente a cierto byte que se ubica en el último bloque de este archivo en los siguientes métodos de asignación de espacio, incluyendo la lectura del bloque de datos en sí, y todas las lecturas necesarias de metadatos del archivo.

(considerando que hemos leído previamente la entrada de directorio del archivo A1, y la tabla Fat o Inodo del archivo si fuera el caso).

a) Asignación contigua. Respuesta: N° total de lecturas=1 Justificación:

Se nos dice que hemos leído previamente la entrada de directorio del archivo A1, donde se almacena n° bloque comienzo

Tamaño de archivo= 3GBytes= $3 \cdot 2^{30}$ B

n° bloques del archivo = $3 \cdot 2^{30} / 2^{10} = 3 \cdot 2^{20}$

N° de bloque a acceder = n° Bloque de comienzo + $3 \cdot 2^{20}$

Ese es el único bloque que hay que leer

b) Lista encadenada de bloques. Respuesta:

Cada bloque almacena un puntero, luego el tamaño para datos que almacena cada bloque es $(2^{10}-4)$ Bytes

n° bloques del archivo = $3 \cdot 2^{30} / (2^{10}-4)$

Se desea acceder al ultimo bloque

Habra que leer el bloque de comienzo, y recorrer la secuencia de punteros hasta haber leído un n° de bloques igual a $3 \cdot 2^{30} / (2^{10}-4)$

N° total de lecturas = $3 \cdot 2^{30} / (2^{10}-4)$

c) Se usa una tabla FAT

Respuesta:

Puesto que se nos dice que la tabla FAT ya está en memoria, hay que seguir la cadena de punteros en ella contenida hasta llegar al último bloque del archivo, luego solo hay que realizar una lectura de disco (el bloque de datos)

EJERCICIO 4. Estamos en un sistema que utiliza **inodos**, en que un inodo en este sistema contiene 10 punteros directos a bloques de datos, el primer puntero en una indexación a 1 nivel, idem para 2 niveles, idem para 3 niveles.

El tamaño de bloque es de 1KBytes y cada puntero a bloque requiere 4 Bytes. En todas las preguntas siguientes consideramos que tenemos ya leídos en memoria los inodos de los archivos que se procesen. En todos los ítem siguientes, cuando se habla del nº de accesos a disco requeridos para acceder a un determinado bloque o byte de un archivo, **no se incluye** la lectura del inodo de disco cuando se habla del nº de accesos a disco requeridos, y **sí hemos de incluir** la lectura o escritura del bloque de datos en sí.

Responda en la hoja de respuestas asignando A=VERDADERO; B=FALSO

| | |
|---|--|
| 1) | F*Para procesar el contenido de archivos inferiores a 10KB no es necesario ningún acceso adicional a disco |
| 2) | V*El nº de punteros que caben en un bloque es de 256 punteros |
| 3) | F*El nº de punteros que caben en un bloque es de 512 punteros |
| 4) | V*Con la indexación a 1 nivel se tiene únicamente 1 bloque de índices |
| 5) | V*Con la indexación a 1 nivel se describe el almacenamiento de 1K*256B de datos del archivo |
| 6) | F*Con la indexación a 1 nivel se describe el almacenamiento de 1K*512B de datos del archivo |
| 7) | V*Un archivo de 256KB de tamaño no necesita la parte de indexación a 2 niveles |
| 8) | F*Un archivo de 266KB de tamaño sí necesita la parte de indexación a 2 niveles |
| 9) | V*Un archivo de tamaño > 266KB de tamaño sí necesita la parte de indexación a 2 niveles |
| 10) | V*Para acceder al byte nº N de un archivo solo se necesita 1 acceso adicional a disco si $N < 10KB$ Para acceder al byte nºN de un archivo, si $N > 10KB$ y $N \leq 1K * 256B$ |
| 11) | F*no hay que utilizar el bloque de índices a 1 nivel |
| 12) | V*se necesitan 2 accesos a disco |
| 13) | F*la respuesta depende del tamaño total del archivo |
| 14) | V*El espacio empleado en metadatos (aparte del inodo) para archivos que no usen ningún tipo de indexación es 0 |
| 15) | F*El espacio empleado en metadatos (aparte del inodo) es independiente del tamaño del archivo |
| 16) | V*El espacio empleado en metadatos (aparte del inodo) aumenta al aumentar el tamaño del archivo |
| Sea el caso de un archivo cuyo tamaño es el máximo permitido en este sistema; el nº de bloques de disco ocupados en metadatos es... | |
| 17) | V* $1 + 1 + 256 + 1 + 256 + 256^2$ |
| 18) | F* $256 + 256^2 + 256^3$ |
| 19) | F* 256^3 |
| 20) | F*El nº de accesos a disco requeridos para acceder al byte N de un archivo es independiente del valor |

| |
|--|
| de N |
| 21) V*El nº de accesos a disco requeridos para procesar un archivo crece al aumentar el tamaño del archivo |
| Sea el caso de un archivo cuyo tamaño es el máximo permitido en este sistema; el nº de accesos a disco requeridos para acceder al último bloque del archivo es |
| 22) F*1 |
| 23) V*4 |
| 24) F*Ninguno de los anteriores |

EJERCICIO 7 Responda en la hoja de respuestas asignando A=VERDADERO; B=FALSO

| |
|--|
| <p>Sobre el buffer caché de Unix:</p> <p>25) V*Cuando un proceso deja de usar un bloque éste pasa al final de la lista de bloques libres</p> <p>26) V*Cuando se necesita un buffer para traer a MP un bloque nuevo, se toma el primero de la lista de bloques libres</p> <p>27) V*Si el nº de marcos de página utilizados para la buffer caché disminuye, entonces disminuirá la velocidad de realización de las solicitudes de E/S de disco</p> <p>28) F*Es imposible que se de el caso de que ante la solicitud de lectura de un bloque, no haya bloques libres.</p> <p>29) V*Buffer sucio significa que el kernel ha de escribir el contenido del buffer en disco antes de reasignar el espacio en MP que ocupa</p> <p>30) F*Buffer sucio significa que no contiene datos válidos.</p> <p>31) F*Si un buffer está en la lista de bloques libres entonces no tiene datos válidos</p> <p>32) V*Mientras haya un proceso utilizando un buffer, éste no está en la lista de bloques libres.</p> <p>33) V*Cuando hay que elegir un bloque libre para se reemplazado por otro que se va a leer desde disco, se elige el bloque libre que lleva más tiempo sin utilizarse.</p> <p>34) F*Cuando un proceso solicita la lectura de un bloque que no está en la caché de buffers y la lista de bloques libres está vacía, se utiliza como bloque víctima el bloque del depósito de buffers que lleva más tiempo sin utilizarse.</p> |
| <p>Sobre el sistema de archivos VFS:</p> <p>35) F*El objeto <code>dentry</code> representa a un directorio</p> <p>36) V*El objeto <code>dentry</code> representa a una entrada de un directorio</p> <p>37) V*El objeto <code>file</code> representa a un archivo abierto</p> <p>38) F*El objeto <code>file</code> puede representar a un directorio si está en uso</p> <p>39) F*El objeto <code>file</code> contiene un flag “dirty” para indicar si la copia en disco está actualizada</p> <p>40) V*El objeto <code>superblock</code> contiene un flag “dirty” para indicar si la copia en disco está actualizada</p> <p>41) V*El objeto <code>inode</code> contiene un flag “dirty” para indicar si la copia en disco está actualizada</p> |
| <p>Sobre el tema 5 (gestión de entradas/salidas):</p> <p>42) V*En una operación de E/S no bloqueante, el proceso de usuario continua ejecutandose tras la petición de E/S aunque ésta no haya terminado</p> <p>43) V*Un manejador de dispositivo es la única parte del SO que conoce la estructura de registros del controlador de dispositivo</p> <p>44) F*Un manejador de dispositivo tiene la responsabilidad de asegurar que las operaciones que un proceso realice sobre un archivo sean solo las autorizadas</p> <p>45) F*El manejador de dispositivo activa a la rutina de tratamiento de interrupción asociada al dispositivo</p> <p>46) V*El manejador de dispositivo, en caso de dispositivos rápidos, tras enviar la petición de E/S al controlador de dispositivo espera una respuesta sin bloquearse</p> <p>47) V*En la planificación de disco, el algoritmo “Menor tiempo de búsqueda (SSF)” tiene el problema de que pueden darse casos de inanición</p> |

- 48)** F*En la planificación de disco, el algoritmo “Política del ascensor o Scan” tiene el problema de que pueden darse casos de inanición
- 49)** V*En la planificación de disco deberá dar preferencia a solicitudes de E/S que provengan de la gestión de memoria virtual frente a las que provengan de procesos de usuario.