

Boletín de Ejercicios de Sistemas de Ficheros

2º curso del Grado en Informática
8 de septiembre de 2017

1. Un sistema de ficheros UNIX tiene bloques de 1 KiB y direcciones de disco de 4 bytes. Se pide:
 - a) ¿Cuál es el tamaño máximo de fichero si los nodos-i contienen 10 entradas directas y 3 indirectas: una sencilla, una doble y una triplemente indirecta?
 - b) Calcula el tiempo necesario para obtener el nodo-i del fichero `/usr/juan/curso/primer.txt`. Supón que el nodo-i del directorio raíz está en memoria, pero ninguna otra cosa está en memoria a lo largo de la ruta. Supón también que cada directorio cabe en un bloque del disco, y que cada operación de disco consume 13,3 ms.
2. Un fichero se compone actualmente de 12 bloques (del 0 al 11). Los metadatos del sistema de ficheros ya están en memoria, por tanto se encuentra en memoria toda la información para saber qué bloques hay libres y cuales están ocupados, y la información del bloque que hay que añadir. Calcula cuántas lecturas y escrituras **de bloques de datos** en disco se necesitan para añadir un bloque al principio, a la mitad (delante del bloque 6) y al final del fichero, respectivamente, en cada una de las tres estrategias de implantación de ficheros que se indican en a), b) y c). Calcula después lo que se pide en el apartado d).
 - a) Asignación adyacente suponiendo que no hay espacio para crecer al principio pero sí al final.
 - b) Lista ligada sin índice.
 - c) Nodos-i de un nivel (suponiendo que los punteros caben en el nodo-i directamente, sin bloques simplemente indirectos ni doblemente indirectos).
 - d) Suponiendo que la longitud del fichero es de 6096 bytes, el tamaño de bloque 512 bytes y que son necesarios 4 bytes para guardar la dirección de un bloque del disco, calcula el número de accesos a disco que provocaría añadir 1 byte al final del fichero en cada una de las tres estrategias anteriores.
3. Un disco de 12 cabezas, 256 sectores (de 512 bytes cada uno) por pista y 2048 cilindros, se formatea con un sistema de ficheros UNIX estándar con un bloque de arranque, un superbloque, mapa de bits de nodos-i, nodos-i y zona de datos de 3 129 308 KiB. Parte de la zona de datos de 3 129 308 KiB está ocupada por el mapa de bits de datos que indica qué bloques de datos están libres. Los nodos-i son estándares de 64 bytes cada uno, 10 punteros directos, un BSI, un BDI y un BTI. El bloque es de 2 KiB y las direcciones de disco de 4 bytes. Se pide:
 - a) ¿Cuántos ficheros como máximo admite el disco?
 - b) ¿Cuál es el tamaño máximo de un fichero con esta estructura de nodo-i, sin tener en cuenta el límite impuesto por el tamaño del disco?
 - c) ¿Cuántos bloques de metadatos (BSI, BDI y BTI) se utilizan en ese fichero?
4. Un sistema de ficheros utiliza bloques de 1 KiB, punteros a bloque de 4 bytes y una caché de disco que suponemos que mantiene en memoria cualquier bloque que hayamos leído previamente. Queremos comparar el número de bloques de disco a los que habría que acceder en varias organizaciones si quisiéramos leer secuencialmente, es decir, cada una a partir de la lectura anterior, los bytes 10 230, 10 240, 266 239, 1 003 519, 1 003 520 y 1 047 552 de un fichero. Indica

razonadamente y con el apoyo gráfico que precises qué bloques del fichero se irán trayendo de disco en cada uno de los casos siguientes. Al final de cada apartado remarca el resultado del número de accesos a disco que han sido necesarios. Se supone que la caché mantiene los bloques ya leídos y no nos supone un acceso volver a leerlos.

- a) Asignación contigua.
- b) Lista ligada.
- c) Nodo-i estándar de 64 bytes, con 10 punteros directos a bloques de datos, un BSI, un BDI y un BTI. El nodo-i se supone que se encuentra ya en memoria.

5. Tenemos un sistema de ficheros UNIX estándar con nodo-i de 64 bytes, 10 entradas directas, BSI, BDI y BTI. Tamaño de bloque 1024 bytes y tamaño del número de bloque de 4 bytes. El sistema operativo mantiene una caché de bloques suficientemente grande.

Queremos leer 5000 KiB a partir del byte 201 533 000, inclusive. Se pide responder a las siguientes cuestiones:

- a) ¿Qué cantidad de bloques de datos tendremos que leer del fichero?
- b) ¿Dónde se encuentran en el nodo-i las dos entrada de metadatos que contienen los números del primer y último bloque de datos a leer, respectivamente? Un ejemplo de respuesta sería: en la entrada 120 del BSI 3 del BDI independiente. La explicación debe ser razonada.
- c) ¿Cuántos bloques BSI, BDI y BTI hemos tenido que leer para leer secuencialmente todos los bloques de datos que nos piden?
- d) Si el sistema de ficheros hubiese sido implementado por lista ligada sin índice, ¿cuántos bloques habríamos tenido que leer?

6. Un disco de 4 cabezas, 2048 cilindros y 256 sectores de 512 bytes por pista, se formatea con un sistema de ficheros UNIX estándar con bloque de 1024 bytes, nodo-i de 64 bytes, 10 punteros directos, 1 BSI, 1 BDI y 1 BTI. Hemos reservado espacio para un bloque de arranque, un superbloque, 262 144 nodos-i, el mapa de bits de nodos-i, el mapa de bits de bloques de datos y los propios bloques de datos. Indica los bloques totales del disco, dedicados a nodos-i, mapas de bits y datos.

7. En un sistema de archivos UNIX estándar tenemos bloques de 1K y direcciones de disco de 4 bytes, nodos-i con 10 bloques directos, un BSI, un BDI y un BTI. ¿Qué bloques tenemos que leer para llegar a leer el byte 12.345.678 de un fichero, y en qué puntero de cada bloque de metadatos se encuentra el siguiente bloque a buscar?

8. Tenemos un disco duro con 8 cabezas, 8192 cilindros y 128 sectores de 512 bytes por pista, formateado con un sistema de ficheros con bloque de 2048 bytes y número de bloque de 4 bytes, con bloque de arranque, superbloque, mapa de bits de nodos-i, espacio para 2 097 152 nodos-i de 64 bytes, mapa de bits de bloques de datos, y zona de bloques de datos. El SF está basado en nodo-i pero sin BSI, BDI y BTI. En su lugar vamos a usar una lista simplemente enlazada de BSI's donde el último número de bloque se usa para apuntar al siguiente BSI. De esta manera en el nodo-i nos caben 12 números de bloque, en lugar de los 10 habituales y solo se usa un último puntero para indicar dónde se encuentra el primer BSI de la lista. Queremos comparar su funcionamiento con el clásico (nodo-i, BSI, BDI y BTI), para lo cual se pide lo siguiente:

- a) Calcula el tamaño de la zona de datos del disco.

- b) Calcula cuántos accesos a disco hacen falta para traer a memoria el bloque 262 666 de un fichero en cada una de las dos configuraciones. Recuerda que los bloques se numeran desde 0.
 - c) Calcula cuánto ocupa realmente en disco un fichero cuyo último bloque sea el 262 666 usando el nodo-i clásico.
 - d) Calcula cuánto ocupa realmente en disco el mismo fichero usando el nuevo tipo de nodo-i.
9. Tenemos tres sistemas de ficheros: organización contigua, lista ligada y nodo-i estándar con 10 punteros directos, un BSI, un BDI y un BTI. El tamaño del bloque es de 1.024 bytes. Se pide calcular el número de bloques que hay que escribir para crear un fichero escribiendo un único byte en la posición 2 097 152 en cada uno de los sistemas de ficheros indicados. ¡OJO! No olvides que las posiciones empiezan por cero, y ten en cuenta que un bloque de un fichero que no se usa en determinados sistemas de ficheros puede que no haga falta que se almacene.
10. Dado un disco duro con un tamaño total de 8 GiB, que se formatea con un sistema de fichero UNIX estándar, con un bloque de arranque maestro (MBR), un superbloque, bloques de mapa de bits de nodos-i, bloques para nodos-i, bloques para mapa de bits de datos, y bloques de datos. Cada nodo-i tiene un tamaño total de 64 bytes y cuenta con 10 punteros directos, un puntero a un BSI, otro a un BDI, y finalmente, otro a un BTI. Los bloques lógicos son de 4 KiB y las direcciones de disco de 4 bytes. Hemos visto que dicho sistema de ficheros puede albergar hasta 94 949 ficheros de 84 KiB cada uno de ellos, sin que sobre ningún bloque en la zona de datos. Se pide:
- a) Calcula el número total de bloques de datos que se requerirían para cada uno de esos ficheros. ¿Cuántos de esos bloques irían destinados a metadatos? Justifica tu respuesta.
 - b) Calcula el tamaño, en bloques, de todas las zonas del disco. En cada caso razona la respuesta que des.
 - c) Dado que para almacenar un fichero se han requerido un total de 257 bloques de datos del disco para guardar sus metadatos (BSI, BDI y BTI), ¿qué tamaño mínimo y máximo podría tener dicho fichero? Razona tu respuesta.
 - d) Indica, justificando adecuadamente tu respuesta, dónde se encuentra exactamente la entrada en los metadatos que contiene el número de bloque correspondiente al byte 268 435 456 de un fichero (empezando la cuenta de los bytes en 0). Ejemplo de respuesta: se encuentra en el puntero 123 del BSI 34 del BDI 3 del BTI de la estructura de nodos-i.

Soluciones a ejercicios seleccionados

■ Ejercicio 1:

- a) $16,06 \text{ GiB} = 1 \text{ KiB} \times (10 + 256 + 256 \times 256 + 256 \times 256 \times 256)$
- b) Tenemos que leer el primer bloque del directorio raíz, después el nodo-i de “usr”, el primer bloque de dicho directorio, el nodo-i de “juan”, el primer bloque de dicho directorio, el nodo-i de “curso”, su primer bloque y el nodo-i de “primero.txt”. En total 8 bloques, a 13,3 ms, 106,4 ms en total.

■ Ejercicio 2:

- a) Asignación adyacente.
 - Añadir bloque al principio: Puesto que no hay espacio para crecer al principio, debemos mover los 12 bloques actuales una posición hacia delante (12L+12E) y luego escribir el nuevo bloque en disco (1E). Total: 25 accesos.
 - Añadir bloque a la mitad: Mover los 6 últimos bloques una posición hacia delante y escribir el nuevo 6L+7E. Total: 13 accesos.
 - Añadir bloque al final: 1E. Total: 1 acceso
- b) Lista ligada.
 - Añadir bloque al principio: Escribimos la dirección de primer bloque del fichero en el nuevo bloque, escribimos el nuevo bloque en disco y nos quedamos con la dirección de dicho bloque como nuevo primer bloque del fichero, 1E. Total: 1 acceso.
 - Añadir bloque a la mitad: Necesitamos saber la dirección del bloque 7 y para ello debemos recorrer los 6 bloques precedentes del fichero (6L) pues son los que contienen la información de la cadena de bloques. En el nuevo bloque guardamos el puntero al bloque 7 (que será el nuevo bloque 8) y lo escribimos (1E). El bloque 6 debe guardar un puntero al nuevo bloque (que será el nuevo 7) por lo que debemos modificar su información y escribirlo de nuevo al disco (1E) (se supone que se ha leído previamente en las 6L) total 8 accesos.
 - Añadir bloque al final: Escribimos el nuevo bloque en un bloque libre del disco (1E). La dirección del nuevo bloque se debe almacenar en el bloque 12 del fichero; para ello primero hay que leer su contenido (recorriendo los 12 bloques del fichero 12L) y después escribirlo de nuevo al disco (1E). Total 12L+2E
- c) Nodo-i.
 - En los tres casos solo hay 1E, la del bloque nuevo que se escribe en disco, ya que los cambios de direcciones de bloque se realizan en el nodo-i (se supone en memoria)
- d) En 12 bloques se pueden almacenar hasta $12 \cdot 512 = 6144$ bytes en las estrategias nodos-i y asignación adyacente. Sin embargo, solo se pueden almacenar 6096 bytes en la estrategia lista ligada sin índice ($12 \cdot 508$ ya que hay 4 bytes en cada bloque ocupado). El aumento en 1 byte de la longitud del fichero provoca el añadido de un nuevo bloque en esta última estrategia (caso ya estudiado). En las otras dos estrategias repercute exclusivamente en la modificación del contenido del último bloque: 1L+1E en ambas.

■ Ejercicio 3:

- a) Tendremos que calcular el número de nodos-i. Conociendo el tamaño de la zona de nodos-i y dividiendo por 64 (64 bytes es el tamaño de un nodo-i) sabremos el número de nodos-i

y por tanto el número de ficheros máximo que admite el disco. Primero calcularemos el tamaño de las zonas **mapa de bits de nodos-i** y **nodos-i**. Una vez calculado el tamaño, lo repartiremos entre las 2 zonas y así conoceremos el tamaño de la zona de nodos-i.

El tamaño total del disco es $12 \times 2048 \times 256 \times 512$ bytes = 1 572 864 bloques de 2 KiB

El tamaño de la zona de datos es de $(3\,129\,308 \text{ KiB} / 2 \text{ KiB por bloque}) = 1\,564\,654$ bloques. Por tanto, el tamaño de la zona restante (zona compartida por el mapa de bits de nodos-i y los nodos-i) es $1\,572\,864 - (1 + 1 + 1\,564\,654) = 8208$ bloques.

En un bloque de datos me caben 32 nodos-i y sabemos que cada bit del mapa de bits corresponde a un nodo-i, por tanto podemos establecer que cada bloque de mapa de bits contiene $2048 \text{ bytes} \times 8 \text{ bits} = 16384$ bits que corresponderían a 16384 nodos-i. Pero como hay 32 nodos-i por bloque, corresponden también a $16384/32 = 512$ bloques. Ahora podemos plantear la siguiente ecuación: $512x + x = 8208$ ya que los bloques de mapa de bits (x) más los bloques de nodos-i que representan ($512x$) tienen que sumar 8208 bloques en total. Despejamos la x y nos da 16 bloques para el mapa de bits. Si lo restamos de 8208, resulta 8192 bloques para nodos-i.

Estos 8192 bloques se traducen en (a 64 bytes por nodo-i, serían 32 nodos-i por bloque) 262 144 nodos-i.

- b) Para calcular el tamaño máximo, primero calculamos que en cada bloque caben $2048/4 = 512$ números de bloque. Por tanto ya el cálculo es sencillo: $(10 + 512 + 512 \times 512 + 512 \times 512 \times 512) \times 2 \text{ KiB} = 268\,960\,788 \text{ KiB}$, aprox. 256,5 GiB.
- c) Para saber la cantidad de bloques de metadatos usados, solamente tenemos que contarlos: $1(\text{BSI}) + 1(\text{BDI}) + 512(\text{BSI}) + 1(\text{BTI}) + 512(\text{BDI}) + 512 \times 512(\text{BSI}) = 263\,171$ bloques.

■ Ejercicio 4:

1. Tamaño de bloque: 1 KB

Los bytes solicitados para lectura secuencial pertenecen a los bloques: 9, 10, 259, 979, 980, 1023.

Con un sistema de asignación contigua, solamente es necesario leer los bloques indicados.

Total de bloques leídos: 6.

2. Con lista ligada, por cada bloque de un fichero se utilizan 4 bytes como puntero al bloque siguiente de este fichero, por lo que la zona de almacenamiento útil de un bloque queda reducida a 1020 bytes.

Los bytes solicitados para lectura secuencial pertenecen a los bloques: 10, 10, 261, 983, 983, 1027.

En un sistema de lista ligada para leer el bloque i-esimo de un fichero hay que recorrer la lista de punteros de este fichero, cargando en memoria los i-1 bloques anteriores, por lo que el **total de bloques leídos es 1028** (desde el 0 al 1027).

3. Tamaño de bloque: 1 KB

Los bytes solicitados para lectura secuencial pertenecen a los bloques: 9, 10, 259, 979, 980, 1023.

Como el puntero a un bloque de memoria es de 4 bytes, cada bloque de índices es capaz de apuntar a 256 bloques.

Bloque 9: se accede directamente desde el nodo-i (el nodo-i apunta a los 10 primeros bloques de datos, desde el 0 al 9). Total: 1 lectura.

Bloque 10: Se accede desde el BSI (el BSI apunta a 256 bloques de datos, desde el 10 al 265). Total: 2 lecturas.

Bloque 259: Como en el bloque 10, se accede desde el BSI, que por tanto, ya ha sido leído. Total: 1 lectura.

Bloque 979: Se accede desde el tercer BSI apuntado por el BDI (desde el BDI, su primer BSI apunta del 266 al 521, el segundo del 522 al 777 y el tercero del 778 al 1033). Total: 3 lecturas.

Bloque 983: Como en el bloque 979, se accede desde el tercer BSI apuntado por el BDI. Por tanto, el BDI y el BSI ya estaban cargados en memoria. Total: 1 lectura.

Bloque 1033: Como en el bloque 979, se accede desde el tercer BSI apuntado por el BDI. Por tanto, el BDI y el BSI ya estaban cargados en memoria. Total: 1 lectura.

Total: 9 lecturas de bloques.

■ Ejercicio 5:

- a) Para saber el número de bloques que leeremos, tendremos que saber en qué bloques comienzan el primer byte a leer y el último, ya que puede ser que la lectura comience por mitad de un bloque. Para ello dividimos el número de byte entre el tamaño de bloque, y obtenemos: $201\,533\,000/1024 = 196\,809,57$ y $(201\,533\,000 + 5000 \times 1024 - 1) = 206\,652\,999$, que si lo dividimos entre 1024 nos da 201 809,57. Por tanto tendremos que leer desde el bloque 196 809 al 201 809 ambos inclusive, dando un total de 5001 bloques.
- b) Para localizar la entrada dentro del nodo-i correspondiente a un bloque concreto, por ejemplo el 196 809, tendremos que restarle 10 correspondientes a las entradas que se encuentran en el propio nodo-i, dando 196 799. Ahora dividimos entre las 256 entradas que tiene cada BSI y nos dirá en qué BSI se encuentra el número de bloque: $196\,799/256 = 768,75$, es decir, el 768 contando desde 0 y desde el BSI independiente. Para saber en qué entrada de ese BSI estará el dato del bloque que buscamos, haremos el módulo: $196\,799 \bmod 256 = 191$.

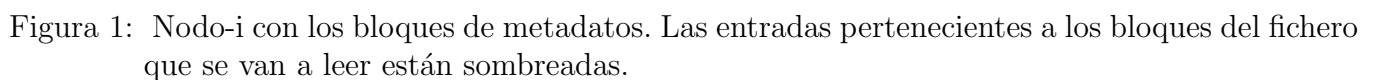
Puesto que se encuentra en el BSI 768, no está en el BSI independiente (sería el BSI 0), así que tendremos que buscar dónde se halla ese BSI. Para ello, restaremos 1 por el BSI independiente, y dividiremos el número entre el número de BSIs que contiene un BDI. Esto nos dará el número de BDI donde se encuentra: $(768 - 1)/256 = 767/256 = 2,996$. Así que está en el BDI 2. Con el módulo sabremos en qué entrada de ese BDI se encuentra: $767 \bmod 256 = 255$. Por tanto ya sabemos que el BSI buscado se encuentra en la entrada 255 del BDI 2. Como el BDI independiente es el BDI 0, el 2 se encontrará en el BTI, así que ya sabemos que estará en el BDI $2 - 1 = 1$ contando desde 0 en el BTI.

Resumiendo, el número del bloque de disco correspondiente al bloque de datos del fichero número 196 809 se encuentra en el BDI 1 del BTI, BSI 255 de ese BDI, y entrada 191 de ese BSI.

Haciendo las mismas operaciones con el bloque final, obtendremos:

- $(201\,809 - 10)/256 = 201\,799/256 = 788,28$, luego el BSI es el 788 contando desde 0, y la entrada en el BSI sería la $201\,799 \% 256 = 71$
- calculamos el BDI dividiendo el número de BSI-1 entre 256 BSIs que tiene un BDI, $(788 - 1)/256 = 787/256 = 3,074$, lo que nos da el BDI 3, y dentro de él sería el BSI $787 \bmod 256 = 19$.
- Al ser un BDI mayor que 0, se encontrará en el BTI, en su entrada $3 - 1 = 2$.

c) Como sabemos dónde se encuentran el primer número de bloque en el nodo- i y el último, podemos examinar la figura 1 y contar el número de bloques que tendremos que leer.



7

- d) Al ser un sistema de ficheros con lista ligada, cada bloque tiene un puntero al siguiente, así que contendrá 1020 bytes realmente. Además para llegar al primer bloque a leer hay que recorrer todos los anteriores desde el principio, así que solamente tendremos que calcular la dirección del último bloque.

Por tanto, para saber el número de accesos que necesitamos hacer, dividiremos 206 652 999 entre 1020, dando $206\,652\,999/1020 = 202\,600,98$, es decir, que tendremos que leer hasta el bloque 202 600 (contando desde el 0), lo que supone leer 202 601 bloques en total.