

Se dispone de un disco de 10 GB. Sobre dicho dispositivo se va a construir una estructura de sistema de ficheros con las siguientes características.

- 4096 bloques para i-nodos.
- Direcciones de i-nodo de 4 bytes.
- i-nodo con los siguientes tamaños:
  - 16 bytes de cabecera (permisos, fechas de modificación, propietario, tipo, ...)
  - 9 punteros directos a bloque.
  - 2 punteros indirectos simple.
  - 1 puntero indirecto doble.

Nota: considérese como tamaño de bloque 512 bytes.

- a) ¿Cuál es el número máximo de ficheros o directorios que puede contener?
- b) ¿Cuál es el fichero más grande direccionable por este formato de i-nodo?
- c) ¿Cuál es el tamaño que deben tener los metadatos apropiados para esta estructura y este disco?

Muchos sistemas UNIX disponen de la posibilidad de utilizar un mismo dispositivo simultáneamente como partición de *swap* (intercambio) y como sistema de ficheros */tmp*. Dicho sistema de ficheros es de carácter temporal y es borrado cada vez que se reinicia la máquina. Un dispositivo compartido de esta forma se va ocupando con páginas de *swap* o ficheros temporales dinámicamente y bajo demanda.

- d) Considerando el carácter temporal de los datos de este sistema de ficheros, ¿Qué elementos de los que componen la estructura del mismo no harían falta tenerlos almacenados sobre el dispositivo? Dibuje la estructura.
  - d.1) ¿Aunque no existan en disco deberían existir en memoria?
- e) Describa el proceso de creación de un nuevo fichero (de 1KB). Indique qué elementos y bloques son accedidos y cuáles son modificados.
- f) Si queremos que el uso del área de *swap* sea gestionado por el sistema de ficheros ¿qué elementos debería modificar una operación de reserva de una página de *swap*?

Al competir por el espacio sobre un mismo dispositivo un problema de estas alternativas es que dicho espacio se agote con las consecuencias que ello implica.

- g) Plantee un rediseño de la estructura o gestión del sistema de ficheros que pueda resolver estas situaciones de la forma más apropiada. Considere las suposiciones que le parezcan más coherentes y razonables.
- h) ¿Sería mejor tener dividido el disco en dos particiones cada una de uso exclusivo de *swap* o fichero temporales?

## Septiembre 2003 - Ejercicio 2

a) ¿Cuál es el número máximo de ficheros o directorios que puede contener?

[1 punto]

El número máximo de ficheros/directorios de un sistema de ficheros tipo UNIX viene determinado por el número de i-nodos disponibles. En el caso de no disponer datos este número se puede estimar en base al tamaño medio del fichero, pero en este caso se nos dice que hay *4096 bloques de i-nodos*. En base a ese dato:

El número de i-nodos en un bloque de i-nodos es:

$$\#_i - nodos = \frac{4096 \text{ bloques}}{Ni - nodos/bloque} \quad (1)$$

Para saber cuántos i-nodos hay en un bloque de i-nodos debemos conocer el tamaño de un i-nodo:

$$Tam_{i-nodo} = 16 \text{ bytes} + (9 + 2 + 1) \text{ direcciones} \times 4 \text{ bytes/direccion} = 64 \text{ bytes} \quad (2)$$

Por lo tanto la fórmula 1 queda como:

$$\#_i - nodos = \frac{4096 \text{ bloques}}{\frac{64 \text{ bytes/i-nodo}}{512 \text{ bytes/bloque}}} = 32768 \quad (3)$$

Como máximo puede haber 32768 (o lo que es lo mismo  $2^{15}$ ) fichero7directorios.

b) ¿Cuál es el fichero más grande direccionable por este formato de i-nodo?

[1 punto]

El fichero más grande direccionable es aquel que ocupe todos los bloques que es posible direccionar desde un í-nodo. Para ello es necesario calcular cuántas direcciones almacena un bloque intermedio de direccionamiento (simple o doble).

$$\#_{dir}/bloque = \frac{512 \text{ bytes/bloque}}{4 \text{ bytes/direccion}} = \frac{2^9}{2r} \text{ direcciones} = 2^7 \text{ direcciones} = 128 \text{ direcciones} \quad (4)$$

Por lo tanto el máximo tamaño direccionable es:

$$Tam = 9 \text{ bloques} + (1 \times 2^7) \text{ bloques} + (2 \times (2^7)^2) \text{ bloques} \geq 2^{15} \text{ bloques} \quad (5)$$

En bytes es aproximadamente.

$$Tam \geq 2^{15} \text{ bloques} \times 512 \text{ bytes/bloque} = 2^{15} \times 2^9 \text{ bytes} = 2^{24} \text{ bytes} = 8 \text{ Mbytes} \quad (6)$$

c) ¿Cuál es el tamaño que deben tener los metadatos apropiados para esta estructura y este disco?

[1 punto]

Los metadatos están compuestos por

- **Sector de Boot:** 1 *bloque*
- **Superbloque:** 1 *bloque*
- **Bitmap de i-nodos:** Si tenemos  $2^{15}$  *i - nodos* se necesitarán.

$$Bloques_{bitmap \ i-nodos} = \frac{2^{15} i - nodos}{8 \text{ bits/byte} \times 512 \text{ bytes/bloque}} = \frac{2^{15}}{2^3 \times 2^9} \text{ bloques} = 2^3 \text{ bloques} \quad (7)$$

- **Bitmap de bloques:** Si despreciamos el resto de estructuras tenemos 10 *Gbytes* de datos que son:

$$Bloques_{datos} = \frac{10 \times 2^{30} \text{ bytes}}{2^9 \text{ bytes/bloque}} = 10 \times 2^{21} \text{ bloques} \quad (8)$$

Para mapear esos  $10 \times 2^{21}$  *bloques* se necesitan

$$Bloques_{bitmap \text{ datos}} = \frac{10 \times 2^{21} \text{ bloques}}{8 \text{ bits/byte} \times 512 \text{ bytes/bloque}} = \frac{10 \times 2^{21}}{2^3 \times 2^9} \text{ bloques} = 10 \times 2^9 \text{ bloques} \quad (9)$$

- **I-nodos:** 4096 *bloques* (según lo descrito por el enunciado).

Para hacernos una idea estos valores en bytes son:

- Boot: 512 *bytes*
- Superbloque: 512 *bytes*
- Bitmap i-nodos: 4096 *bytes*
- Bitmap datos: 5 *Mbytes*
- I-nodos: 4 *Mbytes*

#### d) ¿Qué elementos de los que componen la estructura del mismo no harían falta tenerlos almacenados sobre el dispositivo?

[1 punto]

Si consideramos que el sistema de ficheros se va a borrar cada vez que se monte no tiene sentido guardar ninguna de las estructuras que se corresponden a metadatos, salvo el superbloque. El superbloque, entre otras cosas define el tipo de formato del sistema de ficheros y el tamaño de cada uno de sus componentes.

Los bitmaps pueden eliminarse puesto que al comenzar siempre estarán a cero (debido a que todo está libre) y sólo se necesitará el tamaño. Esto además permitirá borrar implícitamente el dispositivo (si no hay bitmaps todos los bloques e i-nodos están libres).

##### d.1) ¿Aunque no existan en disco deberían existir en memoria?

Evidentemente lo que no exista en disco tiene que existir en memoria, puesto que es necesario para mantener la estructura y gestionar el sistema de ficheros.

Aunque estas son las consideraciones adicionales se pueden considerar cómo afectaría el tamaño de cada estructura sólo existente en memoria:

- El tamaño de los metadatos existentes sólo en memoria es de más de 9 *Mbytes*, esto puede ser excesivo para cada sistema de ficheros que se monte.
- Para sistemas de ficheros tradicionales los bitmaps son imprescindibles en memoria (muchas de las operaciones requieren recorrer dichas estructuras).
- Los i-nodos sin embargo se cargan en memoria bajo demanda. De un fichero que no está en uso se puede prescindir de su i-nodo hasta que se vuelva a trabajar sobre él.

Ante estas características y si no queremos que se dispare el uso de memoria podemos mantener los i-nodos en disco y sólo prescindir de los bitmaps.

**e) Describa el proceso de creación de un nuevo fichero (de 1KB). Indique qué elementos y bloques son accedidos y cuáles son modificados.**

**[1 punto]**

Una operación de este tipo afecta a:

1. El i-nodo del directorio que lo contiene (ajusta la fecha de actualización).
2. El contenido del directorio (añadir una entrada).
3. El bitmap de i-nodos (se reserva uno).
4. El vector de i-nodos (se escribe un i-nodo libre).
5. El bitmap de i-nodos (se reservan dos).
6. Los bloques de datos (para 1KB se necesitan dos bloques).

**f) Si queremos que el uso del área de swap sea gestionado por el sistema de ficheros ¿qué elementos debería modificar una operación de reserva de una página de swap?**

**[1 punto]**

Lo más sencillo es hacer que la reserva de swap para una página se simplifique marcando en el bitmap de bloques tantos bloques como sean necesario por página.

No sería necesario tocar ni i-nodos ni su bitmap asociado. En resumen una reserva de swap es cómo si el disco perdiese bloques de almacenamiento (marcándolos en el bitmap como ocupados).

**g) Plantee un rediseño de la estructura o gestión del sistema de ficheros que pueda resolver estas situaciones de la forma más apropiada.**

**[4 puntos]**

En este apartado es en el que se pueden plantear los diseños más apropiados para resolver el problema.

El principal problema es que se llene el espacio de almacenamiento por el que compiten swap y ficheros temporales. La idea de separarlos y dar un tamaño fijo a cada uno de los usos se ve en el apartado siguiente y aquí no se va a considerar.

La solución al problema es liberar espacio de alguna forma, para ello se tienen dos opciones:

- Eliminar páginas swap de las cuales se tenga copia en memoria. Esta solución no afecta al sistema de ficheros sino al gestor de memoria (por lo tanto no afecta a nuestro diseño, como mucho el sistema de ficheros deberá notificar del problema a dicho gestor para que tome las decisiones oportunas).
- Borrar los ficheros temporales que lleven más tiempo sin ser accedidos (con la condición de que no se encuentren abiertos en ese instante). Esta solución si puede implicar un rediseño del sistema si queremos que se haga en un tiempo razonable. La opción de recorrer todos los i-nodos y eliminar los más viejos puede ser lento.

Una alternativa más revolucionaria para el segundo punto puede ser sustituir el vector de i-nodos (y su bitmap) por una lista enlazada ordenada por tiempo de acceso. De esta forma es fácil encontrar el fichero que hace más tiempo fue accedido. Esta opción requiere que dicha estructura esté almacenada en memoria. Esto puede resolver el problema visto en el apartado d), puesto que salvo en casos

límite esta estructura podría ocupar menos que el vector de i-nodos (4 *Mbytes*).

Debido a que cualquiera de las alternativas (unas más y otras menos) consume tiempo para su ajuste es mejor invocar dichos mecanismos no justo cuando se ha agotado el espacio sino cuando se supere un límite de ocupación (que se considere alarmante).

#### **h) ¿Sería mejor tener dividido el disco en dos particiones cada una de uso exclusivo de swap o fichero temporales?**

[1 punto]

Esta opción no es muy aconsejable puesto que si se aplica un tamaño fijo a dichas particiones, puede darse el caso que una de ellas se llene quedando espacio libre en la otra. Por lo tanto es más proclive que la anterior al proble de no poder o escribir ficheros temporales o hacer swap.

**NOTA:** La gracia de tener swap y /tmp en el mismo dispositivo la tiene sistemas operativos como Solaris desde hace tiempo e incluso algunas versiones de Linux. Esto se motiva por que el uso de dicha partición es temporal (bien por ficheros, bien como área de intercambio en memoria) y su reserva es perder espacio de almacenamiento neto del sistema. Al compartir soporte se puede ajustar más su espacio y no tener una parte llena y otra medio vacía. Al final la gestión de este área de cara al arranque del sistema es comenzar con ella limpia.