

UNIVERSIDAD POLITÉCNICA DE MADRID. FACULTAD DE INFORMÁTICA
DISEÑO DE SISTEMAS OPERATIVOS. 23-6-2004. Examen de la parte teórica.

Ejercicio 2 (4,5 puntos)

Sea una impresora laser (en blanco y negro) de tipo departamental accesible vía IP. Internamente es básicamente un computador de propósito general, sin disco duro pero con una tarjeta de red y con un dispositivo (/dev/lp) capaz de imprimir una página dado el bitmap que debe dibujarse sobre ella. La impresora obtiene vía DHCP la dirección IP en la cual atenderá peticiones de transferencia de archivos (vía diversos protocolos) para ser impresos.

El sistema operativo que ejecuta es tipo UNIX. La imagen del mismo está en una memoria tipo flash desde la cual se arranca. Dispone internamente de 32 MB de RAM, de los cuales 16 se reservan para ser un dispositivo RAMDISK. El tamaño de bloque (en bytes) de este dispositivo puede ser configurado a cualquier potencia de dos. Sobre este RAMDISK se construye un sistema de archivos tipo UNIX minimalista (con nodos-i de 32 bytes (tamaño no modificable)), que se utiliza internamente como *spooler* de impresión.

El diseño de este sistema de ficheros debe maximizar el espacio útil para datos, es decir, el tamaño del mayor archivo imprimible. Dado que el número simultáneo de archivos no se prevé muy grande se limitará éste considerando un único bloque para el mapa de bits de nodos-i. Un estudio preliminar ha mostrado que un tamaño de agrupación de 64 Bytes minimiza el espacio destinado a metadatos (49344 Bytes). No obstante, esta solución presenta un problema inaceptable.

Se desea que identifique claramente el problema mencionado y adopte una solución razonable que cumpla lo mejor posible los requisitos del enunciado. Para ello:

- a) Dibuje un esquema detallado del nodo-i (32 bytes) mínimo necesario, indicando nombre, tamaño y cometido de cada uno de sus campos (razone las decisiones que tome).
- b) Analice la capacidad de direccionamiento del nodo-i necesario para un tamaño de agrupación de 64 Bytes.
- c) Identifique el problema mencionado.
- d) ¿Qué tamaño mínimo de agrupación eliminaría el problema?
- e) Para el tamaño contestado en **d**), calcule el número de bloques que ocuparían los metadatos (bitmaps, etc., etc.) de este sistema de ficheros y su equivalente en bytes.
- f) Calcule paso a paso, el espacio (en bytes) desaprovechado por fragmentación interna en el peor caso posible de utilización de este *spooler* de impresión.

Solución

a)

El bloque es la unidad de acceso al medio de almacenamiento. Se puede acceder al medio para leer o escribir cualquier secuencia de bloques. Como se indica en el enunciado, el tamaño de bloque de este RAMDISK es configurable a cualquier potencia de 2 (en bytes).

Por su parte, la agrupación es la unidad de asignación del espacio del medio y puede ser cualquier potencia de 2 (en bloques). Los punteros del nodo-i y los de los bloques de indirección no apuntan a bloque sino a agrupación (salvo que se considere agrupaciones de tamaño 1 bloque) y el bitmap de gestión de espacio libre indica el estado de libre u ocupada de cada agrupación del disco y no de cada bloque.

Con un tamaño de agrupación de 64 Bytes, un disco de 16 MBytes tiene:

$$2^{24} \text{ (Bytes/Disco)} / 2^6 \text{ (Bytes/Agrupación)} = 2^{18} \text{ (Agrupaciones/Disco)}$$

Luego necesitamos 18 bits para numerar todas las agrupaciones del disco. Lo razonable es que los campos de dirección del nodo-i (y de los bloques de indirección) tengan un ancho potencia de dos. Luego deberían ser de 32 bits (4 bytes).

Como posible optimización podríamos pensar en campos de dirección de 24 bits (3 bytes), pero esto no es nada recomendable, dado que al no ser potencia de 2 nos aparecerían un número no entero de punteros a agrupación en el nodo-i y/o en los bloques de indirección. Esto implicaría complicar la algorítmica de acceso a los ficheros innecesariamente.

Dado que el tamaño del nodo-i es muy reducido (32 bytes) y vamos a intentar reducir los atributos que deben aparecer en él:

Campo	Tamaño (Bytes)
Tipo y Permisos	2
Tamaño	4
Fecha de Modificación	4
Propietario y Grupo	2
Total	12

Caben otras posibles simplificaciones.

Hasta los 32 Bytes del nodo-i, restan 20 bytes para los punteros a agrupación, luego (a 4 bytes por puntero) serán un total de 5. Para maximizar la capacidad de direccionamiento del nodo-i, sin desvirtuar excesivamente el nodo-i UNIX, podríamos considerar un puntero directo, un simple indirecto, y así hasta un cuádruple indirecto.

b)

La capacidad máxima de direccionamiento de un modo-i está dada básicamente por el máximo nivel de indirección que admite y por el número de punteros a agrupación que caben en cada agrupación.

Si la agrupación es de 64 bytes y los punteros a agrupación son de 4 bytes, caben 16 punteros por agrupación. Si usamos hasta un cuádruple indirecto, podremos direccionar:

$$(2^6_{\text{(Bytes/Agrupación)}} / 2^2_{\text{(Bytes/Puntero)}})^4_{\text{(Agrupaciones)}} = 2^{16}_{\text{(Agrupaciones)}}$$

El máximo fichero direccionable podrá ocupar algo más de la cuarta parte del disco.

c)

No es suficiente. El máximo fichero direccionable debe ser mayor (o igual) que el tamaño del disco.

Lo podemos conseguir aumentando el nivel de indirección hasta 6 (pero no caben los punteros en el nodo-i) o reduciendo el número de agrupaciones del disco a 2^{16} .

d)

Precisamente para resolver este tipo de problemas es por lo que existe el concepto de agrupación. Si en vez de 1 agrupación = 1 bloque consideramos agrupaciones de 4 bloques, el número de agrupaciones del disco se reduce inmediatamente a 2^{16} . Esta solución es directa y muy conveniente porque además conseguimos un segundo objetivo. Al reducir el número de agrupaciones a 2^{16} inmediatamente pasamos de necesitar punteros a agrupación de 32 bits a punteros de 16 bits (2 bytes), con lo que el número de punteros que cabrán en el nodo-i (y en los bloques de indirección) se multiplica por dos.

De los 10 punteros (de 2 bytes) a agrupación que ahora caben en el nodo-i podríamos ahora considerar 7 directos, y tres de indirecciones simple, doble y triple.

El número de punteros (de 2 bytes) a agrupación que ahora caben en una agrupación (de $64*4=256$ bytes) es de 128.

Por lo tanto, la capacidad de direccionamiento del nodo-i crece hasta superar el número de agrupaciones:

$$(2^8_{\text{(Bytes/Agrupación)}} / 2^1_{\text{(Bytes/Puntero)}})^3_{\text{(Agrupaciones)}} = 2^{21}_{\text{(Agrupaciones)}}$$

e)

Los metadatos serán (en bloques):

Bloque de Boot (1), Superbloque (1), Bitmap de Agrupaciones (X), Bitmap de Nodos-i (1 (es dato)) y Vector de Nodos-i (Y).

$$X = 2^{16}_{\text{(Bits)}} / 2^{6+3}_{\text{(Bits/Bloque)}} = 2^7_{\text{(Bloques)}} = 128_{\text{(Bloques)}}$$

$$Y = 2^5_{\text{(Bytes/Nodo-i)}} * 2^{6+3}_{\text{(Nodos-i)}} / 2^6_{\text{(Bytes/Bloque)}} = 2^8_{\text{(Bloques)}} = 256_{\text{(Bloques)}}$$

Sumando 387 bloques de 64 Bytes, o 24768 Bytes.

Curiosamente y gracias al ajuste realizado (1 agrupación = 4 bloques), hemos conseguido además reducir a la mitad el espacio ocupado por los metadatos frente a los 49536 Bytes indicados en el enunciado.

f)

La fragmentación interna por fichero se considera normalmente media agrupación (dado que el enunciado habla de "el peor caso posible", cabría tomar este valor como una agrupación menos un byte). El peor caso posible será para el máximo número de ficheros en esta situación.

El resultado es:

$$\text{Fragmentación} = 2^{6+3}_{\text{(Nodos-i)}} * 2^8_{\text{(Bytes/Agrupación)}} * 2^{-1}_{\text{(Agrupación/Nodo-i)}} = 2^{16}_{\text{(Bytes)}} = 65536_{\text{(Bytes)}}$$