

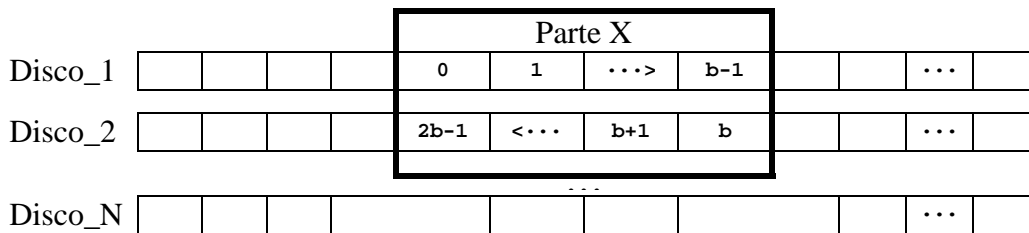
ENUNCIADO

Se desea diseñar una variedad de Sistema de Ficheros tipo UNIX (basado en nodos-i, bitmaps, etc.) adaptándolo a su uso sobre un dispositivo virtual de almacenamiento constituido por un conjunto de **N** discos (o particiones) todos de igual tamaño.

- A) Se pide que **describa** y **dibuje un esquema** que refleje la disposición de cada una de las partes del Sistema de Ficheros sobre el dispositivo, indicando claramente cual sería la proyección de la numeración lógica de los bloques de cada parte sobre el dispositivo.

Ejemplo de presentación de resultado:

En este ejemplo se muestra la posición de la **Parte X** sobre el dispositivo virtual, y se detalla la localización de los **2b** bloques de dicha parte que está ubicada sobre los 2 primeros discos.



Haga esto para cada una de las siguientes tres prioridades estratégicas posibles:

- 1) Que el Sistema de Ficheros pueda adaptarse fácilmente a ampliaciones del dispositivo virtual consistentes en añadir un nuevo disco de igual tamaño.
- 2) Que el Sistema de Ficheros tolere el posible fallo de uno de los **N** discos. Comente la(s) técnica(s) aplicada(s).
- 3) Maximizar las prestaciones del Sistema de Ficheros, aprovechando el paralelismo existente entre los **N** discos independientes.

- B) Considere ahora el caso concreto de 5 discos de 4 Gigabytes cada uno, y la aplicación de la primera estrategia del apartado anterior. Indique cuál sería el tamaño concreto de cada una de las partes del Sistema de Ficheros. Realice para ellos las suposiciones razonables que considere necesarias.

SOLUCIÓN

A) En un Sistema de Ficheros tipo UNIX tradicional se distinguen las siguientes partes:

BB	SB	BMB	BMNi	VNi	BBDD...
----	----	-----	------	-----	---------

- BB:** **Bloque de Boot.** Contiene código ejecutable para arrancar de dicho disco o partición. Ocupa un único bloque. Debe aparecer siempre, y es siempre, el primer bloque del dispositivo (disco o partición).
- SB:** **Superbloque.** Contiene la descripción del tipo de Sistema de Ficheros. Ocupa un único bloque. Es vital para el correcto reconocimiento del Sistema de Ficheros como tal y para la localización de sus partes. Por lo tanto, y siempre que se pueda, mantendremos varias copias de seguridad de él en posiciones preestablecidas del dispositivo.
- BMM:** **Bitmap de bloques.** Permite gestionar los bloques del dispositivo (disco o partición). Con un bit por bloque se indica si dicho bloque está libre u ocupado. Ocupa el número de bloques necesario para tener un bit por bloque.
- BMNi:** **Bitmap de nodos-i.** Permite gestionar los nodos-i del sistema de ficheros. Con un bit por nodo-i se indica si dicho nodo-i está libre u ocupado. Ocupa el número de bloques necesario para tener un bit por nodo-i.
- VNi:** **Vector de Nodos-i.** Contiene, uno detrás de otro, los nodos-i del Sistema de Ficheros. Ocupa el número de bloques necesario para contener todos los nodos-i.
- BBDD:** **Bloques de Datos.** El resto del espacio se dedica a los bloques que dan soporte al almacenamiento de los datos de ficheros, directorios, etc. así como los bloques de direcciones directas, indirectas o doble indirectas a los que apuntan los nodos-i.

1) Para que el Sistema de Ficheros que nos piden pueda adaptarse a ampliaciones consistentes en nuevos discos (o particiones) de igual tamaño, realizaremos una gestión independiente de cada disco o partición del dispositivo virtual, pero manteniendo una numeración única para los bloques de datos.

Dicho de otra manera, en cada uno de los discos tendremos:

- Un **Bloque de boot**, para que eventualmente el sistema pueda arrancar de cualquiera de los discos.
- Un **Superbloque**, de forma que tendremos tantas réplicas del superbloque como discos.
- Un **Bitmap de bloques** y un **Bitmap de nodos-i**. Servirán para gestionar los bloques y nodos-i de ese disco. Un nuevo disco contendría sus propios bitmaps para su gestión.
- Un **Vector de Nodos-i** conteniendo los nodos-i de ese disco.

Pero para el dispositivo virtual como un todo, tendremos:

- Un único espacio de **Bloques de Datos** compuesto por la concatenación del espacio para datos de cada disco. Numeraremos los bloques de datos consecutivamente dentro de un disco, continuando la numeración en el siguiente.

Gráficamente...

					Bloques de Datos					
Disco 1	BB	SB	BMB _[0..n]	BMNi _[0..n]	VNi _[0..n]	0	1	2	...	b-1
Disco 2	BB	SB	BMB _[0..n]	BMNi _[0..n]	VNi _[0..n]	b	b+1	b+2	...	2b-1
Disco 3	BB	SB	BMB _[0..n]	BMNi _[0..n]	VNi _[0..n]	2b	2b+1	2b+2	...	3b-1
Disco 4	BB	SB	BMB _[0..n]	BMNi _[0..n]	VNi _[0..n]	3b	3b+1	3b+2	...	4b-1
...				
Disco N	BB	SB	BMB _[0..n]	BMNi _[0..n]	VNi _[0..n]	(N-1)b	(N-1)b+1	(N-1)b+2	...	Nb-1

- 2) La disposición de las partes del Sistema de Ficheros puede ser idéntica a la del primer apartado, pero para tolerar el posible fallo de un disco es necesario que al menos la información de un disco sea información redundante.

Aplicaremos una técnica tipo RAID orientada a bloque, de manera que los bloques de uno de los discos (del último) serán bloques de paridad, esto es, contendrán la paridad (XOR) de los otros bloques de los otros discos.

Gráficamente...

Contenido de un bloque b de cada disco			
Disco 1	...	A	...
Disco 2	...	B	...
Disco 2	...	C	...
...
Disco N-1	...	Z	...
Disco N	...	$A \wedge B \wedge C \wedge \dots \wedge Z$...

- 3) La disposición de las partes del Sistema de Ficheros puede ser también la del primer apartado, pero como nuestro objetivo prioritario es maximizar las prestaciones en el acceso a los datos, estableceremos una numeración de los **Bloques de Datos** tal que el primer bloque esté en el primer disco, el segundo, en el segundo disco y así crecerá la numeración repartiéndose cíclicamente los bloques sobre los discos.

De esta forma, al ubicar un fichero en bloques lógicamente contiguos, estaremos realmente repartiéndolo cíclicamente entre los discos y un acceso secuencial al fichero (el más común) será realmente un acceso paralelo a varios discos.

Gráficamente...

					Bloques de Datos					
Disco 1	BB	SB	BMB _[0..n]	BMNi _[0..n]	VNi _[0..n]	0	b	2b	...	(N-1)b
Disco 2	BB	SB	BMB _[0..n]	BMNi _[0..n]	VNi _[0..n]	1	b+1	2b+1	...	(N-1)b+1
Disco 3	BB	SB	BMB _[0..n]	BMNi _[0..n]	VNi _[0..n]	2	b+2	2b+2	...	(N-1)b+2
Disco 4	BB	SB	BMB _[0..n]	BMNi _[0..n]	VNi _[0..n]	3	b+3	2b+3	...	(N-1)b+3
...				
Disco N	BB	SB	BMB _[0..n]	BMNi _[0..n]	VNi _[0..n]	b-1	2b-1	3b-1	...	Nb-1

B) Considerando 5 discos de 4 Gigabytes cada uno, y la aplicación de la primera estrategia descrita en el apartado anterior, tendremos que el tamaño de cada una de las partes (y otras unidades) del Sistema de Ficheros será:

(Bloque): Un tamaño de bloque razonable para este caso son 4 Kilobytes

$$= 2^{12} \text{ bytes/bloque.}$$

(Disco): $= 2^{32} \text{ bytes/disco} / 2^{12} \text{ bytes/bloque} = 2^{20} \text{ bloques/disco.}$

BB: Un bloque **Bloque de Boot** por cada uno de los 5 discos, que será el primer bloque de cada disco.

SP: Un bloque **Superbloque** por cada uno de los 5 discos, que será el segundo bloque de cada disco. Todos ellos serán replica de la misma información.

BMB: Un **Bitmap de bloques** por cada uno de los 5 discos, con un bit por bloque de dicho disco.

$$= 2^{20} \text{ bits} / 2^{12+3} \text{ bits/bloque} = 2^5 \text{ bloques.}$$

(Número de Nodos-i): Para determinarlo consideramos el número máximo posible de ficheros no vacíos, de tamaño mínimo (1 bloque). Así tendremos un fichero por bloque en el disco.

$$= 2^{20} \text{ nodos-i.}$$

BMNi: Un **Bitmap de nodos-i** por cada uno de los 5 discos, con un bit por cada nodo-i del Sistema de Ficheros correspondiente a dicho disco, luego este bitmap tendrá igual tamaño que el **Bitmap de bloques** $= 2^5 \text{ bloques.}$

(Nodo-i): Un tamaño razonable para el nodo-i de este sistema son 256 bytes

$$= 2^8 \text{ bytes/nodo-i.}$$

VNi: Un **Vector de Nodos-i** por cada uno de los 5 discos, con tantos nodos-i como los considerados anteriormente.

$$= 2^{20} \text{ nodos-i} * 2^8 \text{ bytes/nodo-i} / 2^{12} \text{ bytes/bloque} = 2^{16} \text{ bloques.}$$

BBDD: El resto de cada uno de los 5 discos, destinado al almacenamiento de bloques.

$$= 2^{20} - (1 + 1 + 2^5 + 2^5 + 2^{16}) \text{ bloques.}$$