

Se quiere comparar varias configuraciones para los metadatos de un sistema de ficheros para un dispositivo de 32GiB. Las diferentes configuraciones son:

	Configuración A	Configuración B	Configuración C
Tamaño de bloque lógico	4KiB	4KiB	8KiB
Número de punteros (directos, simples, dobles y triples)	5,1,1,1	13,1,1,1	6,1,1,0
Tamaño de puntero a i-nodo / bloque	4 bytes	8 bytes	8 bytes

Aspectos comunes:

- Los campos del i-nodo comunes a las 3 configuraciones ocupan 96bytes.
- Para el vector de i-nodos se reservan 3GiB de espacio en el dispositivo. El resto serán otros metadatos y los propios datos.
- La estructura de directorio tiene entradas que son de tamaño fijo, de 128 caracteres.

Configuraciones especiales:

- La configuración B usa el espacio reservado para punteros directos/indirectos para almacenar el contenido del fichero cuando éste es suficientemente pequeño como para entrar íntegramente en ese espacio (al estilo de XFS).
- La configuración C hace lo mismo, pero además tiene un campo de 256 bytes adicionales en el i-nodo reservado para guardar el comienzo del fichero (que se sumaría al ocupado por los punteros directos/indirectos). Si el fichero excede el tamaño para guardarlo en el i-nodo, este campo no se usa.

Se pide, **para cada una de las configuraciones (A, B y C):**

1. Indicar cuántos i-nodos entran en el vector de i-nodos (tal y como se ha reservado espacio para él).
2. Indicar el tamaño (en bloques lógicos) de los bitmaps de i-nodos y bloques.
3. ¿Cuál es el tamaño de fichero más grande direccionable por la jerarquía de punteros directos e indirectos de cada nodo?

Dada la siguiente estructura de contenidos:

DIRECTORIO RAÍZ	
.	<dir>
..	<dir>
DATA	<dir>
USER.CFG	64bytes
VI.MPG (enlace físico a DATA/VIDEO.MPG)	

DIRECTORIO DATA	
.	<dir>
..	<dir>
FICH.TXT	270bytes
VIDEO.MPG	64MiB
VIDEO_AUX.MPG (enlace simbólico a VIDEO.MPG)	

Teniendo en cuenta la **configuración C únicamente:**

4. Indique cuántos bloques lógicos ocupan los datos y metadatos de cada una de las entradas de ambos directorios, así como los propios directorios.
5. ¿Cuáles serían los bloques que se leerían para resolver la siguiente secuencia de operaciones?:

```
int num, fd;
fd = open ("/DATA/VIDEO_AUX.MPG", O_RDONLY);
seek(fd, -sizeof(int), SEEK_END);
read(fd, &num, sizeof(int);
```

6. Si la configuración C permitiese el uso de *extents*, con un campo de 12 bits para indicar el número de bloques, incluyendo un campo de 4 bits para indicar tipo, uso y otras máscaras. ¿Cuál sería el fichero de mayor tamaño direccionable, independientemente del tamaño del propio dispositivo?

SOLUCIÓN

1. Indicar cuántos i-nodos entran en el vector de i-nodos (tal y como se ha reservado espacio para él).

- Configuración A:
 - Tamaño del i-nodo = 96 bytes + (5+1+1+1)x4 bytes =128bytes
 - I-nodos por vector= 3 GiB / 128 bytes/i-nodo= $3 \cdot 2^{30}$ bytes / 2^7 bytes/i-nodo= $3 \cdot 2^{23}$ i-nodos
- Configuración B:
 - Tamaño del i-nodo = 96 bytes + (13+1+1+1)x8 bytes =224bytes
 - I-nodos por vector= 3 GiB / 224 bytes/i-nodo= $3 \cdot 2^{30}$ bytes / 224 bytes/i-nodo= $13,71 \times 2^{20}$ i-nodos
 - (En el orden de $3 \cdot 2^{22}$ i-nodos)
- Configuración C:
 - Tamaño del i-nodo = 96 bytes + (6+1+1+0)x8 bytes + 256 bytes =416bytes
 - I-nodos por vector= 3 GiB / 416 bytes/i-nodo= $3 \cdot 2^{30}$ bytes / 416 bytes/i-nodo= $7,38 \times 2^{20}$ i-nodos
 - (algo menos de 2^{23} i-nodos)

2. Indicar el tamaño (en bloques lógicos) de los bitmaps de i-nodos y bloques.

- Configuración A:
 - BitMap i-nodos: 3×2^{23} i-nodos $\rightarrow 3 \times 2^{23}$ bits= 3×2^{23} bits/ 2^3 bits/byte= 3×2^{20} bytes= 3×2^{20} bytes / 4kiB/Bloque= 3×2^{20} bytes / 2^{12} bytes/bloque= 3×2^8 bloques=768 bloques
 - BitMap bloques= $32-3$ GiB= 29×2^{30} bytes/4kiB/Bloque= 29×2^{18} bloques de datos
 29×2^{18} bloques de datos $\rightarrow 29 \times 2^{18}$ bits = 29×2^{18} bits/ 2^3 bits/byte= 29×2^{15} bytes= 29×2^{15} bytes / 4kiB/Bloque= 29×2^{15} bytes / 2^{12} bytes/bloque= 29×2^3 bloques=232 bloques
- Configuración B:
 - BitMap i-nodos: $13,71 \times 2^{20}$ i-nodos $\rightarrow 13,71 \times 2^{20}$ bits= $13,71 \times 2^{20}$ bits/ 2^3 bits/byte= $13,71 \times 2^{17}$ bytes= $13,71 \times 2^{17}$ bytes / 4kiB/Bloque= $13,71 \times 2^{17}$ bytes / 2^{12} bytes/bloque= $13,71 \times 2^5$ bloques ≈ 438 bloques
 - BitMap bloques [Ídem que en el caso anterior, los bloques son de 4KiB] =232 bloques
- Configuración C:
 - BitMap i-nodos: $7,38 \times 2^{20}$ i-nodos $\rightarrow 7,38 \times 2^{20}$ bits= $7,38 \times 2^{20}$ bits/ 2^3 bits/byte= $7,38 \times 2^{17}$ bytes= $7,38 \times 2^{17}$ bytes / 8kiB/Bloque= $7,38 \times 2^{17}$ bytes / 2^{13} bytes/bloque= $7,38 \times 2^4$ bloques ≈ 118 bloques
 - BitMap bloques= $32-3$ GiB= 29×2^{30} bytes/8kiB/Bloque= 29×2^{17} bloques de datos
 29×2^{17} bloques de datos $\rightarrow 29 \times 2^{17}$ bits = 29×2^{17} bits/ 2^3 bits/byte= 29×2^{14} bytes= 29×2^{14} bytes / 8kiB/Bloque= 29×2^{14} bytes / 2^{14} bytes/bloque= 29×2 bloques=58 bloques

3. ¿Cuál es el tamaño de fichero más grande direccionable por la jerarquía de punteros directos e indirectos de cada nodo?

- Configuración A:
 - Número de punteros por bloque indirecto=4KiB bloque / 4 bytes/dirección= 2^{10} direcciones
 - El límite del tamaño direccionable: 5 bloques (directos) + 1×2^{10} bloques (indirecto simple) + $1 \times 2^{10} \times 2^{10}$ bloques (indirecto doble) + $1 \times 2^{10} \times 2^{10} \times 2^{10}$ bloques (indirecto triple) $\approx 2^{30}$ bloques = 2^{30} bloques x 4KiB/bloque = 2^{42} bytes = 4 TiB
- Configuración B:
 - Número de punteros por bloque indirecto=4KiB bloque / 8 bytes/dirección= 2^9 direcciones
 - El límite del tamaño direccionable: 13 bloques (directos) + 1×2^9 bloques (indirecto simple) + $1 \times 2^9 \times 2^9$ bloques (indirecto doble) + $1 \times 2^9 \times 2^9 \times 2^9$ bloques (indirecto triple) $\approx 2^{27}$ bloques = 2^{27} bloques x 4KiB/bloque = 2^{39} bytes = $\frac{1}{2}$ TiB

- Configuración C:
 - Número de punteros por bloque indirecto=8KiB bloque / 8 bytes/dirección= 2^{10} direcciones
 - El límite del tamaño direccionable: 256 bytes + 6 bloques (directos) + 1×2^{10} bloques (indirecto simple) + $1 \times 2^{10} \times 2^{10}$ bloques (indirecto doble) $\approx 2^{20}$ bloques = 2^{20} bloques x 8KiB/bloque = 2^{33} bytes = 8 GiB

4. *Indique cuántos bloques lógicos ocupan los datos y metadatos de cada una de las entradas de ambos directorios, así como los propios directorios.*

Debemos calcular cuál es el fichero más grande que entra dentro del i-nodo (a-la XFSS): 256 bytes + (8 punteros x 8bytes/puntero)=320 bytes.

- Directorio raíz (5 entradas=5 entradas x (128+4) bytes/ entrada=660 bytes):
 - 1 i-nodo
 - 1 bloque de datos.
- Las entradas "." y ".." hacen referencia (ambas al mismo i-nodo del directorio raíz).
- Directorio DATA (5 entradas=5 entradas x (128+4) bytes/ entrada=660 bytes):
 - 1 i-nodo
 - 1 bloque de datos.
- Fichero USER.CFG (64 bytes)
 - 1 i-nodo
 - 0 bloque de datos (entra todo dentro del i-nodo).
- Enlace físico V1.MPG
 - 0 i-nodos (es el mismo i-nodo que el fichero destino, aquí no lo contamos).
 - 0 bloques de datos (aquí tampoco los contamos)
- Fichero FICH.TXT (270 bytes)
 - 1 i-nodo
 - 0 bloque de datos (entra todo dentro del i-nodo).
- Fichero VIDEO-MPG (64 MiB= 2^{26} bytes / 8 KiB/bloque= 2^{13} bloques)
 - 1 i-nodo (compartido con el enlace físico V1.MPG).
 - 2^{13} bloque de datos (8×2^{10} bloques)
 - Como excede el número de bloques directos necesitará un indirecto (que le permite 2^{10} bloques más) además del indirecto doble (y 7 bloques del siguiente nivel de indirección)= 9 bloques de indirección.
- Enlace simbólico VIDEO_AUX.MPG (enlace simbólico)
 - 1 i-nodo
 - 0 bloque de datos (entra todo dentro del i-nodo, el contenido es la ruta a VIDEO.MPG).

5. *¿Cuáles serían los bloques que se leerían para resolver la siguiente secuencia de operaciones?:*

```
int num, fd;
fd = open ("/DATA/VIDEO_AUX.MPG", O_RDONLY);
seek(fd, -sizeof(int), SEEK_END);
read(fd, &num, sizeof(int);
```

Apertura del fichero (resolución de la ruta):

- Se lee el i-nodo del directorio raíz.
- Se lee su bloque de datos (contenido).
- Se lee el i-nodo del directorio DATA
- Se lee su bloque de datos (contenido).
- Se lee el i-nodo del fichero VIDEO_AUX.MPG (es un enlace simbólico, se tiene que resolver).
- Se lee el i-nodo del directorio VIDEO.MPG.

Nos vamos al final del fichero y se lee:

- Se debe acceder al puntero doble, luego al último puntero indirecto ocupado en el siguiente nivel.
- Finalmente se lee el último bloque de datos.

6. *Si la configuración C permitiese el uso de extents, con un campo de 12 bits para indicar el número de bloques, incluyendo un campo de 4 bits para indicar tipo, uso y otras máscaras. ¿Cuál sería el fichero de mayor tamaño direccionable, independientemente del tamaño del propio dispositivo?*

- Cada extent permite direccionar 2^{12} bloques = 2^{12} bloques x 8 kiB/bloque = 2^{25} bytes = 32 MiB/extent
- El fichero dispone de 256 bytes + (8 punteros x 8 bytes/puntero) = 320 bytes dentro del i-nodo.
- Cada extent son 12 bytes (tamaño) + 4 bytes (tipo y máscaras) + 8 bytes (puntero al primer bloque) = 24 bytes/extent
- Cada i-nodo puede referenciar 320 bytes / 24 bytes/extent = 13 extents = 13 extents x 32 MiB/extent = 416 MiB