

## TGR – SISTEMA DE FICHEROS 1

En un sistema de archivos (tipo System V) tenemos un archivo (**bujuan**) correspondiente al buzón de correo del usuario Juan, quien es su propietario, y la ruta absoluta del archivo es:

**`/var/spool/mail/students/bujuan`**

Responde las siguientes preguntas:

1. ¿Qué número mínimo de accesos a disco serán necesarios en la operación **`open("/var/spool/mail/students/bujuan",O_RDONLY)`** para obtener el i-nodo de **bujuan** si las entradas de los diferentes subdirectorios se encuentran siempre en el primer bloque del directorio padre, excepto **students** que se encuentra en el cuarto bloque y el archivo **bujuan** que se encuentra en el segundo? Se supone que el Caché de Buffer y el Cache de I-Nodos están inicialmente vacíos.

**Número de accesos mínimos: 10**

**Área de datos: 9**

**Área de i-nodos: 1**

2. En un sistema de archivos de este tipo, el tamaño de bloque es de 2 kilobytes y los i-nodos tienen 12 direcciones de bloque directo, 1 dirección de bloque indirecto simple, 1 dirección de bloque indirecto doble y 1 dirección de bloque indirecto triple. Además, las direcciones de bloque se representan con 4 bytes. ¿Cuántos bloques de disco son necesarios para almacenar el archivo **bujuan** si su tamaño es de 6 megabytes? La respuesta debe detallar cuántos bloques son para datos y cuántos son para índices.

**Número de bloques de datos: 3072**

**Número de bloques de índices: 7**

3. Siguiendo el ejercicio 1, una vez que el archivo está abierto, el proceso ejecuta la llamada al sistema:

**`lseek(fd, 4194304, SEEK_SET)`**

¿Cuántos bloques tendría que leer el sistema operativo para cumplir con la sentencia:

**`c=fgetc(fd)`**

si se supone que la Caché de Búfer está vacía? (Nota:  $4194304 = 4 * 2^{20}$ )

**Número de bloques que el sistema operativo debe leer: 3**

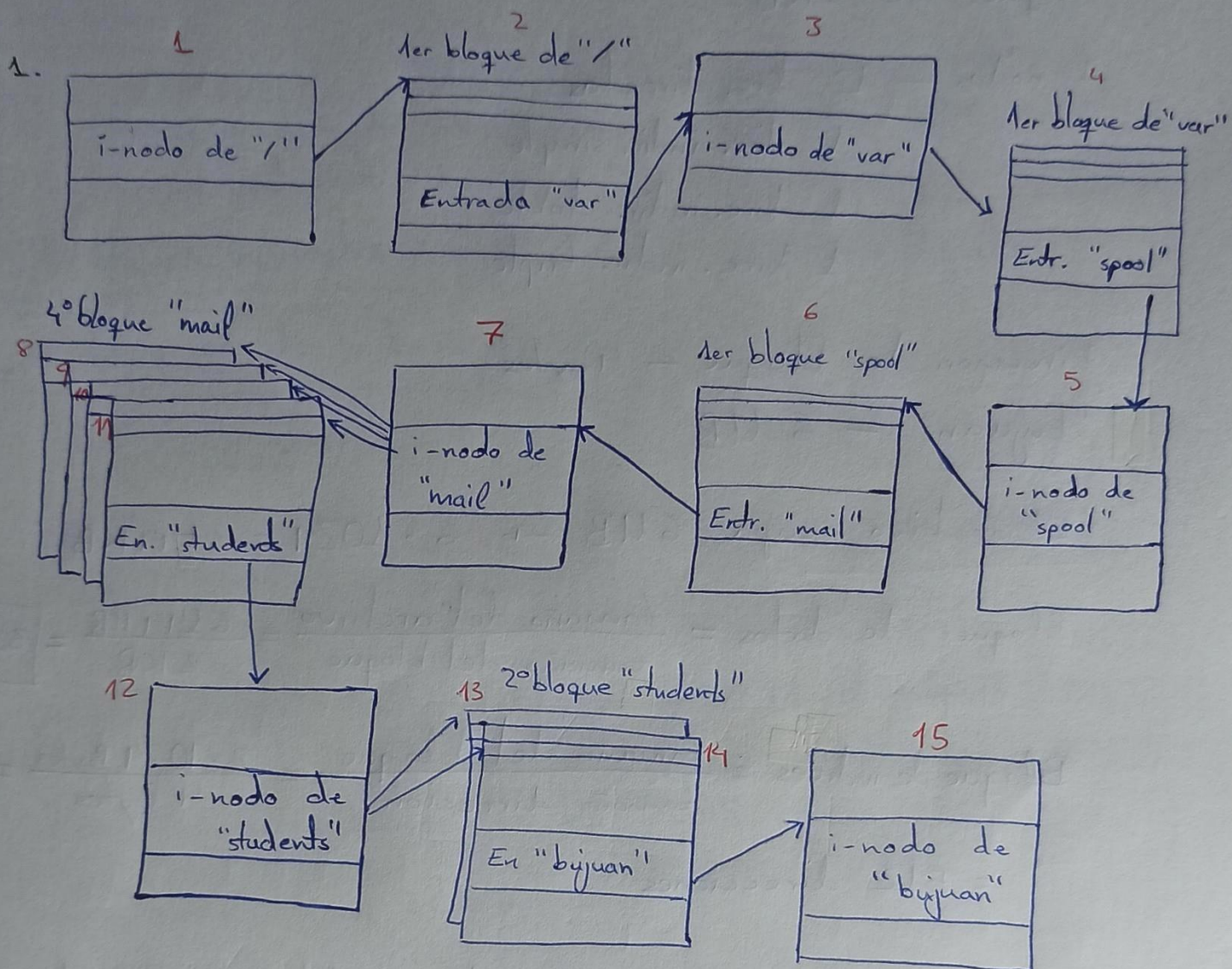
4. ¿Cuál es el número de bloque lógico en el sistema de archivos que corresponde al i-nodo del directorio raíz? (los bloques lógicos se numeran comenzando con el bloque lógico 0) y ¿cuál es el número de bloque lógico que corresponde al i-nodo del archivo **bujuan**? Suponiendo lo siguiente:

- i. El número de i-nodo de "/" es 2, y al archivo ***bujuan*** se le asigna el número de i-nodo 35 (los i-nodos se numeran a partir del i-nodo 1).
- ii. El tamaño del i-nodo es de 128 bytes.
- iii. El arranque (boot) ocupa 1 bloque y el superbloque (superblock) requiere 8 bloques.

**Número de bloque lógico del inode de "/": 9**

**Número de bloque lógico del inode de "bujuan": 11**

# TGR - Sistema de Ficheros 1



Total = 15 accesos = 6 i-nodos + 9 datos

En el mejor de los casos todos los i-nodos vienen y en el bloque de "/" (el primer bloque).

Solución:

1 bloque con los i-nodos.

9 bloques de datos.

10 accesos mínimos.



2.

Bloques de 2KB

I-nodos — 12 dire. directas  
                  1 dire. indir. simple  
                  1 dire. indir. doble  
                  1 dire. indir. triple

Dirección de bloque — 4bytes

"bjuan" — 6 MB

El archivo ocupa 6 MB  $\rightarrow 6 \cdot 1024 = 6144 \text{ KB}$

$$\text{Bloques de datos} = \frac{\text{tamaño del archivo}}{\text{tamaño del bloque}} = \frac{6144 \text{ KB}}{2 \text{ KB}} = \boxed{3072 \text{ bloques}}$$

$$\begin{aligned} \text{Bloque de índices} &= \frac{\text{tamaño del bloque}}{\text{tamaño dirección}} = \frac{2 \cdot 1024 \text{ Bytes}}{4 \text{ Bytes}} = \\ &= 512 \text{ direcciones} \end{aligned}$$

Los 12 primeros bloques de datos se obtienen directamente desde el i-nodo.

Los 512 siguientes requieren 1 bloque indirecto simple.

En el siguiente nivel tenemos 1 bloque indirecto doble que permite indexar hasta 512 bloques indirectos simples.

$$\begin{aligned} 3072 \text{ bloques} &\rightarrow (12 \text{ bloque directo simple} + 512 \text{ indirecto simple} = \\ &= 2548 \text{ bloques por indexar en grupos de 512.} \end{aligned}$$

$$\frac{2548 \text{ bloques}}{512 \text{ grupos}} = 4,97 \rightarrow 5 \text{ bloques indirectos simples}$$



2.

Solución:

1 bloque indirecto simple  
 + 1 bloque indirecto doble  
 5 bloques indirectos simples

7 bloques de índices

3072 bloques de datos
+ 7 bloques de índices
3079 bloques

3.

Primeros  $4 \cdot 2^{20}$  bytes se enumeran  $0 \dots (4 \cdot 2^{20} - 1)$

$$\frac{4 \cdot 2^{20} \text{ bytes}}{2 \cdot 2^{10} \text{ bytes/bloque}} = 2 \cdot 2^{10} \text{ bloques} = 2048 \text{ bloques}$$

Están almacenados en los 2048 primeros bloques y se enumeran  $0 \dots 2047$

El byte  $4 \cdot 2^{20}$  está en el byte 2049 - ésimo (en número, el 2048).

Los 12 primeros bloques ( $0 \dots 11$ ) van por índice directo.

Los 512 siguientes ( $12 \dots 523$ ) van por indirecto simple.

Los  $512 \times 512$  siguientes ( $524 \dots 524 + 512^2 - 1$ ) van por indirecto doble. 2048 está en ese intervalo.

Solución:

1 bloque indirecto simple  
 + 1 bloque indirecto doble  
 1 bloque de datos

3 bloques



4. Datos:  
i-nodo  $\rightarrow$  128 bytes

"/" inodo 2  
"bajuan" inodo 35 } Se empieza a contar en el i-nodo 1

boot ocupa 1 bloque

superblock ocupa 8 bloques

1 i-nodo ocupa 128 B =  $2^7$  Bytes

1 bloque son 2KBytes  $\rightarrow 2 \times 2^{10}$  B =  $2^{11}$  Bytes

$$\frac{2^{11} \text{ Bytes/Bloque}}{2^7 \text{ Bytes/i-nodo}} = 2^4 \text{ i-nodos/bloque} = 16 \text{ i-nodos/bloque}$$

El bloque lógico 0 es la "boot", los bloques lógicos 1... 8 son el "superblock".

El bloque 9 contiene los i-nodos del 1 al 16. Por tanto:  
bloque lógico de "/" = 9.

El bloque 10 contiene los i-nodos del 17 al 32. El bloque 11 contiene los i-nodos del 33 al 48. Por tanto:

bloque lógico de "bajuan" = 11.

Solución:

Número de bloque lógico del i-nodo "/": 9

Número de bloque lógico del i-nodo "bajuan": 11