

1. Se implementa ext3 en un disco duro con bloques de 4 kB. En ext3, la estructura de un i-nodo que siempre ocupa un bloque, es la siguiente:
- El tamaño del fichero en bytes. Ocupa 6 bytes.
 - Un identificador del dispositivo. Ocupa 6 bytes.
 - Un identificador del usuario. Ocupa 2 bytes.
 - Un identificador del grupo. Ocupa 2 bytes.
 - Los permisos y tipo del fichero. Ocupa 2 bytes.
 - Varias marcas de tiempo:
 - Última modificación del i-nodo. Ocupa 4 bytes.
 - Última modificación del fichero. Ocupa 4 bytes.
 - Último acceso al fichero. Ocupa 4 bytes.
 - Instante en el que se ha borrado el fichero. Ocupa 4 bytes.
 - Cuenta de enlaces. Ocupa 2 bytes.
 - El resto son punteros directos y un puntero indirecto, un indirecto doble y uno triple. Cada puntero ocupa 4 bytes.

Responde a las siguientes preguntas. **Explica los cálculos según los vayas realizando (sin explicación los resultados no tendrán valor).**

- A. Calcula el tamaño máximo del disco según esta estructura.

Solución:

El tamaño máximo del disco duro viene determinado por el número de bloques direccionables. Como los punteros a los bloques son de 4 B entonces puedo direccionar $2^{4 \times 8}$, es decir 2^{32} bloques. Como cada bloque es de 4 kB (2^{12} B) entonces el tamaño máximo direccionable de un disco con este sistema de archivos es de $2^{32} \times 2^{12}$ B, es decir **16 TB**.

- B. Calcula el tamaño máximo de un fichero.

Solución:

Sumando los valores correspondientes el tamaño de la parte informativa del inodo ocupa 36 bytes. De esta forma quedan disponibles $4096 - 36$ B para punteros, es decir 4060 B. Como los punteros ocupan 4 B entonces tengo 1015 punteros, de los cuales 1012 son directos.

En cada bloque de punteros caben 4 kB / 4 B punteros, es decir, 1024 punteros. de esta forma nos queda lo siguiente:

- **Punteros directos:** 1012 punteros \times 4 kB . Es decir, el tamaño máximo de un fichero con punteros directos es de 4145152 B.
- **Puntero indirecto simple:** es un puntero que apunta a un bloque de punteros que apuntan a bloques de datos y por tanto se dispone de 1024 punteros a bloques de 4 kB. Es decir, el fichero podría ocupar con el indirecto simple lo que ocupa con directos más 4 MB.
- **Puntero indirecto doble:** es un puntero que apunta a un bloque de punteros indirectos simples. Es decir permite acceder a 1024 punteros indirectos simples o lo que es lo mismo 1024×4 MB. Por tanto un fichero podría ocupar lo mismo que en el caso anterior más 4 GB.
- **Puntero indirecto triple:** es un puntero que apunta a un bloque de punteros indirectos dobles. Es decir permite acceder a 1024 punteros indirectos dobles o lo que es lo mismo 1024×4 GB. Por tanto un fichero podría ocupar lo mismo que en el caso anterior más 4 TB.

Tamaño máximo del fichero = 4 TB + 4 GB + 7,9 MB > 4 TB. Como se puede observar, este esquema permite almacenar ficheros de tamaño grande.

Si la capacidad del disco fuera menor, el límite vendría dado por lo que permite el disco descontando aquella cantidad necesaria para la gestión del mismo.

2. Supón un sistema de gestión de ficheros similar al de Unix, con bloques de tamaño 1 kB, en el que los punteros de direccionamiento a bloques son de tamaño 4 bytes. En cada i-nodo se tienen 10 punteros directos a bloques de datos, un puntero indirecto simple y uno indirecto doble. Se quiere incrementar el tamaño máximo del fichero. ¿Cuál de las siguientes acciones permitiría un mayor aumento del tamaño del fichero?
- A. Añadir un puntero a un bloque indirecto triple.
 - B. Aumentar el tamaño del bloque a 4 kB

Razona la respuesta indicando qué ventajas e inconvenientes tiene cada una de las soluciones propuestas.

3. Sea un sistema de fichero tipo Unix que gestiona bloques de datos de 4K. Cada i-nodo, además de otra información, contiene 10 enlaces directos a bloques de datos, un enlace de indirección simple, un enlace de indirección doble y un enlace de indirección triple. Responde razonadamente a las siguientes cuestiones:

- A. Suponiendo que tenemos una cache de 20 bloques de datos y otra de 20 i-nodos inicialmente vacías, que el nodo-i del directorio raíz se encuentra en memoria principal, y que sólo ejecuta en el sistema el siguiente proceso:

```
fd1=open("/usr/pablo/practica1");
fd2=open("/usr/jose/practica1");
lseek(fd1,12*4096);
leidos=read(fd1,datos,5);
exit(0);
```

¿Cuántos accesos a disco realiza el proceso suponiendo que la información de cada directorio que accede cabe en un solo bloque de datos? Enuméralos indicando la acción que se realiza en cada acceso.

Solución: Empezando por el primer open, tenemos que resolver `/usr/pablo/practica1` con lo que hacemos los siguientes accesos:

1. Leemos el primer bloque de datos del i-nodo del directorio raíz. (Buscamos en la tabla `usr`).
2. Leemos el i-nodo para `usr`.
3. Leemos el primer bloque de datos de dicho i-nodo. (Buscamos en la tabla `pablo`).
4. Leemos el i-nodo para `pablo`.
5. Leemos el primer bloque de datos de dicho i-nodo. (Buscamos en la tabla `practica1`).
6. Leemos el i-nodo para `practica1`.

Continuando con el segundo open, resolvemos `/usr/jose/practica1` teniendo en cuenta que ya tenemos en la cache de i-nodos el i-nodo para `usr` y en la cache de bloques el bloque de datos de dicho i-nodo, por lo que la primera lectura será:

7. Leemos el i-nodo para `jose`.
8. Leemos su primer bloque de datos. (Buscamos en la tabla `practica1`).
9. Leemos el i-nodo para `practica1`.

Ahora continuamos con el `lseek`, que no realiza acceso alguno a disco. y nos deja posicionados justo al principio del bloque 12 del primer fichero abierto (si empezamos a contar los bloques desde 0). Tras ello leemos 5 bytes, con lo que habrá que leer dicho bloque:

10. Leemos el bloque, B, apuntado por el primer puntero indirecto simple del i-nodo que corresponde a `/usr/pablo/practica1`.
11. Leemos el bloque apuntado por el tercer puntero presente en B (el primer puntero apuntará al bloque 10 del fichero, el segundo puntero apuntará al 11, y el tercero es el que nos interesa).
12. Leemos los 5 primeros bytes de ese bloque.

- B. ¿Tiene fragmentación este tipo de sistema ficheros? ¿De que tipo? ¿Cuál es el tamaño medio de la fragmentación de cada fichero?

Solución: En cuanto a la fragmentación, tenemos algo de fragmentación interna dado que asignamos bloques a ficheros. La fragmentación externa es nula dado que todos los bloques son del mismo tamaño. En media, cada fichero desperdicia la mitad de su último bloque, esto es, 2K por fichero.

4. Un fabricante de memorias tipo flash lanza una tarjeta con un sistema de ficheros FAT 16 destinada a cámaras fotográficas. El tamaño de bloque se fija en 32 KB. El fabricante tiene la intención de ir sacando al mercado tarjetas de capacidad cada vez mayor según avance la tecnología.

Contesta razonadamente a las siguientes cuestiones:

- A. Calcula el límite de la capacidad de almacenamiento de estas tarjetas.
- B. Para una tarjeta de 512 Mbytes, calcula el espacio que hay que asignar para almacenar la FAT en la tarjeta.
- C. Se prevé que el tamaño medio de las fotografías que almacenarán estas tarjetas va a ser de aproximadamente 1 MB. Calcula una estimación de la fragmentación interna que puede esperarse en estas tarjetas.

5. El sistema de ficheros **ext3** de Linux está basado en i-nodos. El diseño es el siguiente:

- Cada i-nodo ocupa 128 bytes, estando 68 bytes dedicados a los atributos del archivo (permisos, usuario, etc.) y 60 bytes a los punteros a bloques de disco.
- Estos punteros se organizan de la siguiente forma: 12 punteros directos, 1 indirecto simple, 1 indirecto doble y 1 indirecto triple.
- Para almacenar cada puntero a bloque se utiliza un entero de 4 bytes con signo, es decir, quedan 31 bits para direccionar el número de bloque.
- El número de i-nodos (default) viene dado por el tamaño del volumen (por ejemplo el disco) expresado en bytes dividido por 2^{13} . Por ejemplo: en un disco de 2 GB, el número de i-nodos por defecto es 2^{18} .

Suponga que se configura un sistema **ext3** con un tamaño de bloque de 2 kB (2^{11} Bytes). Determinar razonadamente:

- A. Cantidad máxima de bloques que puede contener un sistema de archivos con esta configuración.
- B. Tamaño máximo del sistema de archivo (volumen) que se puede crear con esta configuración.
- C. Tamaño máximo de archivo que se puede crear con esta configuración.
- D. Suponga que se crea un sistema de archivos para un disco de tamaño 1 TB (2^{40} Bytes). ¿Cuál es el tamaño de la tabla de i-nodos?
- E. Con la misma configuración del apartado anterior, ¿cuál es la máxima cantidad de ficheros que pueden ser creados en ese disco?
- F. Para el mismo disco del apartado anterior, si se utiliza un mapa de bits para llevar el registro de bloques ocupados y desocupados, ¿cuánto ocupará este mapa?

6. Se tiene un sistema de ficheros basado en i-nodos con las siguientes características: 10 punteros directos, 2 indirectos simples, 2 indirectos dobles y 1 indirecto triple. El tamaño de los punteros es de 64 bits y los bloques del disco son de tamaño 1024 B.

- A. ¿Cuál es el tamaño máximo de un archivo?

Solución: En un bloque caben $1024/8 = 128$ punteros a bloques.

En el primer nivel tenemos 10 punteros a bloques y por tanto se pueden direccionar 10×1 kB. Con dos punteros indirectos simples se pueden direccionar $2 \times 128 \times 1$ kB. Con dos dobles: $2 \times 128 \times 128 \times 1$ kB. Con uno triple: $1 \times 128 \times 128 \times 128 \times 1$ kB. En total el tamaño máximo de un fichero es: $(10 + 2 \times 128 + 2 \times 128 \times 128 + 128 \times 128 \times 128) \times 1$ kB ≈ 2 GB

- B. Si ya hemos leído el i-nodo correspondiente ¿cuántos bloques de disco hace falta leer para acceder al byte número 1228864 de un archivo, considerando que el i-nodo correspondiente al archivo ya está en la memoria principal?

Solución: Como la división entera es $1228864/1024=1200$ y el resto es 64 hay que usar el bloque 1200.

El direccionamiento se realiza por el primer indirecto doble. Se lee el bloque al que apunta el indirecto doble. En este bloque se usa el puntero $(1200-256-10)/128$, es decir el puntero 7 que direcciona a un bloque donde se encuentra el byte deseado.

Todo ello implica que se leen tres bloques: dos bloques del direccionamiento + bloque donde se encuentra la posición buscada.

- C. Si se utilizara un mapa de bits para mantener la información sobre los bloques ocupados/libres del disco, ¿cuál sería el tamaño de dicho mapa?

Solución: El tamaño máximo del disco viene determinado por el tamaño del puntero en el i-nodo (64 bits). Por tanto, como máximo tendremos 2^{64} bloques. Si para cada bloque necesitamos 1 bit, el tamaño del mapa de bits máximo deberá ser de 2^{64} bits = 2^{61} bytes. 2 HB !!!!

7. Se tiene un sistema de ficheros basado en i-nodos con las siguientes características: 8 punteros directos, 2 indirectos simples, 1 indirecto doble y 1 indirecto triple. El tamaño de cada bloque del disco es 256KB y el tamaño de los punteros a bloques es 4 B.

- A. ¿Cuál es el tamaño máximo que puede tener un archivo en este sistema?

Solución: En cada bloque caben $256/4$ kPunteros = 64 kPunteros = 65536. El tamaño máximo de un fichero es $(8 + 2 * 65536 + 65536^2 + 65536^3 \approx 2^{48} \text{ B} = 256 \text{ TB}) \times 256 \text{ kB}$

- B. ¿Cuál es el tamaño máximo del disco?

Solución: Con punteros de 4 B se pueden direccionar $2^{8 \times 4}$ bloques. Como cada bloque tiene 2^{18} B, el tamaño máximo del disco es $2^{50} = 1024 \text{ TB}$

- C. Comenta la relación entre el tamaño máximo de un archivo y el tamaño máximo de disco e indica cuál puede ser el objetivo de tener dichos tamaños.

Solución: El tamaño máximo de un fichero levemente inferior al tamaño máximo del disco, de hecho la cuarta parte. Esta estructura permite tener ficheros extremadamente grandes y con sólo 4 ficheros podría llenarse completamente el disco.

8. Se tiene un sistema de ficheros basado en i-nodos con las siguientes características: 8 punteros directos, 2 indirectos simples, 1 indirecto doble y 1 indirecto triple. El tamaño de cada bloque del disco es 256KB y el tamaño de los punteros a bloques es 4 B.

- A. ¿Cuál es el tamaño máximo que puede tener un archivo en este sistema?

- B. ¿Cuál es el tamaño máximo del disco?

- C. Comenta la relación entre el tamaño máximo de un archivo y el tamaño máximo de disco e indica cuál puede ser el objetivo de tener dichos tamaños.

9. Tenemos un obsoleto disco duro de 32 MB que queremos recuperar del desván para un ordenador donde hemos instalado una versión muy antigua de Unix. Este S.O. formatea el disco del siguiente modo:

Sector de arranque	Superbloque	Mapa de bits	Tabla de i-nodos	Bloques de datos	...
--------------------	-------------	--------------	------------------	------------------	-----

Las especificaciones del sistema son:

- El tamaño de un bloque es 1KB.

- El sector de arranque y el superbloque ocupa 1 bloque cada uno.
- Tamaño de la dirección de un bloque es de 2 bytes.
- El número de i-nodos disponibles es de 1024.
- El i-nodo contiene la siguiente información:
 - Tipo de archivo y protección. (4 bytes).
 - Número de enlaces. (4 bytes).
 - Propietario (4 bytes).
 - Grupo (4 bytes).
 - Tamaño del fichero.(4 bytes).
 - Fecha de creación (4 bytes).
 - Fecha de actualización (4 bytes).
 - Fecha de último acceso (4 bytes).
 - 10 entradas de referencias directas a bloque. (2 bytes*entrada)
 - 4 entradas de referencia indirecta simple a bloque. (2 bytes*entrada)
 - 2 entradas de referencias indirectas de 2º nivel. (2 bytes*entrada).

A. Calcular el tamaño de:

1. El mapa de bits.
2. La tabla de i-nodos.
3. El máximo efectivo del disco después de formatear.

B. En el caso de no tener limitación de capacidad (disco tan grande como fuera necesario), ¿cuál sería el tamaño máximo teórico de un archivo?

C. Un proceso necesita acceder a una posición relativa del fichero “datos.txt” situada en el bloque 530. En el momento de estudio el fichero sólo ha sido abierto y, en consecuencia, actualizada la entrada en la tabla de i-nodos. ¿Cuántos accesos a disco habría que realizar para acceder a la posición indicada?

10. Un dispositivo tiene una memoria secundaria gestionada por un sistema de ficheros configurado con una tabla de asignación de ficheros FAT8 (tamaño de los punteros de dirección es de 8 bits). Los bloques del disco son de tamaño 1KB. Contesta **razonadamente** a las siguientes cuestiones:

A. Tamaño máximo del sistema de ficheros.

Solución:

Como se dispone de punteros de 8 bits el máximo posible es 2^8 bloques, es decir $2^8 \times 1$ kB. Por tanto el tamaño máximo es 256 kB.

B. ¿Cuántos bloques ocupa la tabla FAT? ¿Qué fragmentación se produce por mantener la FAT?

Solución:

Como cada puntero a bloque ocupa 1 B y por tanto sólo hace falta un cuarto de bloque para estos punteros. Aunque hay bits de protección y atributos, todos ellos juntos nunca llegarán a ocupar los 24 bits que quedarían a disposición de cada entrada sin que por ello se desborde el tamaño de un bloque. Por tanto la FAT ocupa sólo un bloque y la fragmentación externa debida a la FAT es de tan sólo un bloque y la interna será la parte restante del bloque que no sea necesario ocupar dependiendo de los bits de protección y atributos que se implementen. Si no se tienen en cuenta estos bits, la fragmentación interna sería de 768 B.

En un determinado momento la FAT tiene la siguiente configuración en decimal. La tabla contiene las posiciones como mecanismo de ayuda.

0	EOF	15	7	30	38	45	12	60	110
1	98	16	123	31	67	46	0	61	35
2	32	17	6	32	0	47	3	62	231
3	9	18	14	33	17	48	23	63	0
4	13	19	21	34	8	49	115	64	40
5	87	20	15	35	22	50	88	65	19
6	28	21	53	36	1	51	192	66	89
7	65	22	91	37	143	52	78	67	126
8	16	23	84	38	156	53	46	68	0
9	102	24	5	39	187	54	0	69	2
10	106	25	51	40	11	55	109	70	77
11	59	26	44	41	73	56	4	71	173
12	33	27	203	42	101	57	207	...	
13	69	28	56	43	96	58	201	...	
14	60	29	EOF	44	27	59	71	...	

Se tienen la siguiente información en el directorio:

Nombre del fichero	Tamaño (KB)	Primer Bloque
Examen_Ordinaria.docx	8	20
Notas_Ordinaria.xlsx	12	45

- A. Para cada uno de los ficheros, ¿cuáles son los bloques que ocupa el fichero?

Solución:

Examen_Ordinaria.docx ocupa (en orden) los bloques: 20, 15, 7, 65, 19, 21, 53 y 46.

Notas_Ordinaria.xlsx ocupa (en orden) los bloques: 45, 12, 33, 17, 6, 28, 56, 4, 13, 69, 2 y 32.

- B. Se quiere acceder a la posición 3106 (contabilizando en Bytes) del fichero `Notas_Ordinaria.xlsx`, ¿Cuántos acceso al disco hay que realizar? ¿En qué bloque físico del disco está dicha posición?

Solución:

Como la FAT se carga en memoria en el arranque sólo es necesario leer el 4º bloque del fichero que es donde se encuentra el byte ($3106/1024 = 3$ y algo), es decir, sólo es necesario leer el bloque **17**.

11. Un dispositivo de memoria USB tiene una capacidad de 16 GB, el dispositivo se puede formatear con un sistema de ficheros FAT. El tamaño de los bloques de datos es de 4 KB. Se está dudando si formatear con FAT16 o FAT32 (16 o 32 bits por dirección).

Con FAT16:

- A. ¿Cuántos bloques ocuparía la FAT y por qué?
 B. ¿Qué porcentaje del dispositivo USB sería visible por el sistema de ficheros y por qué?

Con FAT32:

- A. ¿Cuántos bloques ocuparía la FAT y por qué?
 B. ¿Qué porcentaje del dispositivo USB sería visible por el sistema de ficheros y por qué?

12. Un sistema de archivos tipo system V tiene un tamaño de bloque de 8 Kbytes e i-nodos con una estructuras de direccionamiento de: 10 direcciones directas de bloques, una indirecta simple, una indirecta doble y una indirecta triple. Además, utiliza direcciones de bloques de 8 bytes.

- A. Determina el número de accesos físicos a disco necesarios, como mínimo, en este sistema, para ejecutar la siguiente operación:

```
fd = open (‘‘/home/Horus/SOPER.2014-15/Notas/Extraordinaria.csv’’, RD_ONLY);
```

Supón que la caché del sistema de archivos está inicialmente vacía y que el inodo del directorio `/home/` está ya en memoria. Representa los distintos accesos en un diagrama.

B. Una vez abierto el archivo, las dos siguientes ordenes son:

```
lseek(fd, 2097152, SEEK_SET);
```

```
c=fgetc(fd);
```

Calcula de forma razonada cuántos bloques habría que leer ahora de disco para obtener el valor de la variable c. (Nota: $2097152 = 2^{21}$).

13. El sistema de ficheros de una tarjeta flash está basado en FAT16 con tamaño de bloques de hasta 32 KB.

¿Cuál es tamaño máximo permitido para las tarjetas flash?

En una tarjeta de 1 GB con bloques de tamaño 32 KB, ¿Cuál es el tamaño de la tabla FAT?

En una tarjeta flash con tamaño de bloque de 32 KB y FAT16, la información de un directorio presente en la tarjeta flash y una fracción de la tabla FAT se representan a continuación:

0	MBR
1	Arranque
2	Superbloque
3	14
4	EOF
5	EOF
6	5
7	3
8	EOF
9	6
10	LIBRE
11	LIBRE
12	LIBRE
13	DAÑADO
14	8
15	LIBRE
...	...

2.5 Nombre	Primer bloque	Longitud (B)
image052.jpg	7	1024×1024
image053.jpg	4	828
image054.jpg	9	20×1024

¿Qué bloques ocupa cada uno de los ficheros especificados en el directorio dado?

Nombre	Bloques
image052.jpg	
image053.jpg	
image054.jpg	

¿Qué tipo de fragmentación se produce? Explica por qué.

¿Qué cantidad de fragmentación se produce en cada uno de los ficheros especificados?

Nombre	Fragmentación
image052.jpg	
image053.jpg	
image054.jpg	

Se quiere acceder a la posición 102400 (en bytes) del fichero image052.jpg. Sabiendo que el sistema permite el acceso aleatorio y la tabla FAT está completamente cargada en memoria, ¿qué bloques se deberían cargar en memoria para acceder a dicha posición? Explica por qué.

14. Se dispone de un disco duro de 32 TB con bloques de 4 kB. El disco duro se formatea con un sistema de ficheros basado en i-nodos. En cada bloque caben exactamente 32 i-nodos que tienen la siguiente estructura:

- El identificador de dispositivo del dispositivo que alberga al sistema de archivos (1 byte)
- El número de i-nodo que identifica al archivo dentro del sistema de archivos (4 bytes)
- La longitud del archivo en bytes (4 bytes)
- El identificador de usuario del creador o un propietario del archivo con derechos diferenciados (2 bytes)
- El identificador de grupo de un grupo de usuarios con derechos diferenciados (2 bytes)
- El modo de acceso: capacidad de leer, escribir, y ejecutar el archivo por parte del propietario, del grupo y de otros usuarios (1 byte)
- Las marcas de tiempo con las fechas de última modificación (mtime), acceso (atime) y de alteración del propio i-nodo (4 bytes cada uno)
- El número de enlaces, esto es, el número de entradas de directorio asociadas con este i-nodo.(1 byte)
- Punteros directos (4 bytes cada uno)
- Un puntero de indirección simple (4 bytes)
- Un puntero de indirección doble (4 bytes)
- Dos punteros de indirección triple (4 bytes)

Bytes que ocupa el i-nodo: 128 B

Bytes que ocupan los atributos del i-nodo: 27 B

Número de punteros directos: 21

¿Cual es el tamaño máximo del sistema de ficheros? Razona el resultado

Solución:

El tamaño máximo del sistema de ficheros es el número máximo de bloques direccionables por el puntero (2^{32}) multiplicado por el tamaño del bloque (2^{12}). Es decir, el tamaño máximo es de 2^{44} bytes, es decir, 16 TB. Como el disco duro es de 32 TB, será necesario utilizar al menos dos particiones para aprovechar completamente el disco.

¿Cual es el tamaño máximo de un fichero? Razona el resultado

Solución:

Si la estructura del i-nodo menos los punteros directos es de 43 bytes y el tamaño del i-nodo es de 128 bytes (4 kB/32) se dispone de 128-43 bytes para punteros directos, disponemos de (128-43)/4 punteros directos, es decir 21 punteros directos. Además, en cada bloque caben $4 \text{ kB} / 4 = 1024$ punteros a bloques.

El tamaño máximo direccionable según la estructura del i-nodo es $21 \times 4 \text{ kB} + 1024 \times 4 \text{ kB} + 1024 \times 1024 \times 4 \text{ kB} + 2 \times 1024 \times 1024 \times 1024 \times 4 \text{ kB} \approx 2^{43}$ bytes o lo que es lo mismo 8 TB. Según el campo de la longitud del i-nodo el tamaño máximo del fichero debe almacenarse en este campo de 4 bytes y por tanto no puede ser mayor de 2^{32} bytes. En este caso el factor limitante del tamaño de fichero es el campo del i-nodo que determina el tamaño del fichero y no el determinado por los punteros del i-nodo.

En este sistema de ficheros se tienen los siguientes directorios:

Directorio /home/ana/asignaturas/soper		Directorio /home/eloy/ssoo	
Nombre	i-nodo	Nombre	i-nodo
imagen.jpg	37	migato.jpg	2451
documento.pdf	1238	documento.pdf	451
examenDeHoy.txt	847	examenParaHoy.txt	847
abejaruco.mp3	72	huecco.mp3	72
vayapeli.mp4	291	quepeli.jpg	347

En este sistema sólo tenemos dos usuarios. A las 10:03 del 10/05/2016 el usuario **ana** se haya “dentro” del directorio `/home/ana/asignaturas/soper` y ejecuta el siguiente comando: `rm examenDeHoy.txt`. Explica qué sucede con el fichero y qué sucede con su i-nodo.

Solución: Como se puede ver en el directorio `/home/elay/ssoo` el i-nodo correspondiente al fichero `examenDeHoy.txt` es el mismo que el de el fichero `examenDeHoy.txt` y por tanto en el campo `cuenta` del i-nodo habrá un valor de 2. Cuando el usuario **ana** borra el fichero se elimina la entrada de su directorio y se decrementa el valor de `cuenta` del i-nodo y al no ser 0 éste permanece ocupado.

El usuario **elay** entra en el sistema y desde su directorio raiz ejecuta a las 11:13 del 10/05/2016 el comando `cat ssoo/examenParaHoy.txt`. Qué lecturas de ficheros se realizan (si es que se pueden realizar) y los i-nodos de que ficheros o directorios tienen que leerse.

Solución: Como el usuario ya ha entrado al sistema el i-nodo correspondiente a su directorio de usuario `/home/elay` ya ha sido leído y por tanto está en memoria. Al ejecutar el comando se realizan por orden las siguientes acciones:

1. Se lee el fichero correspondiente al directorio `/home/elay`.
2. De este fichero se obtiene el número del i-nodo del directorio `/home/elay/ssoo`.
3. Se lee el i-nodo correspondiente.
4. Con este i-nodo se lee el directorio `/home/elay/ssoo`.
5. De este fichero se obtiene el i-nodo del fichero `/home/elay/ssoo/examenParaHoy.txt`
6. Se lee el i-nodo correspondiente.
7. Con el i-nodo del fichero ya puede leerse el fichero y presentarse por pantalla.

Una vez realizado el comando anterior el usuario **elay** ejecuta el comando `rm ssoo/examenParaHoy.txt`. Explica qué sucede con el fichero y con su i-nodo.

Solución: Como su cuenta ya está a 1, al eliminarlo este usuario se elimina de su directorio, se pone la cuenta a 0 y por tanto es necesario dejar libre el i-nodo y los bloques asignados a ese fichero se marcan como libres en el mapa de bits correspondiente.

El usuario **ana** ejecuta el comando `rm documento.pdf` mientras el usuario **elay** está haciendo el comando anterior y, posteriormente, el usuario **elay** ejecuta el comando `rm documento.pdf`. Explica qué sucede en este proceso.

Solución: Como los i-nodos de los dos ficheros son distintos cada uno se borra independientemente lo que implica que cada uno de los i-nodos queda libre cuando se ejecuta el comando. Así mismo los bloques asignados a cada fichero deben marcarse como libres en el mapa de bits de bloques libres y se eliminan las entradas correspondientes a estos ficheros en ambos directorios.

15. Tenemos el conjunto de i-nodos y bloques del anexo. Suponiendo que tras arrancar el sistema sólo está en caché el i-nodo 1 y ejecutamos el siguiente código:

```
FILE *a, *b;  
char datos[8];
```

```

a = fopen ("/hom/ana/fca","r");
fseek(a,44,SEEK_SET);
read(a,datos,8);
fclose(a);

b = fopen ("/hom/elo/fcb","w");
fseek(b,34,SEEK_SET);
write(b,datos,8)
fclose(b);

```

Todo esto teniendo en cuenta que las condiciones de nuestro sistema son las siguientes:

- Bloques de 16 bytes. Los punteros en los bloques de punteros ocupan 1 Byte.
- i-nodos con dos punteros directos y un indirecto simple. (Sólo se presenta esta información en el anexo).
- Las entradas del directorio tienen 1 byte para el número de i-nodo y 3 bytes para el nombre del fichero (en este orden).
- Los índices tanto de los bloques como de los i-nodos empiezan en 1 y cualquier puntero a 0 indica que no hay más punteros en el i-nodo.

Nota: La función fseek tal y como se utiliza, indica la posición dentro del fichero de la que se va a realizar la próxima lectura o escritura.

En la siguientes tablas tienes que indicar la secuencia de accesos que realizará cada parte de este código, indicando según se corresponda el número de i-nodo o de bloque que se está utilizando. La tabla está numerada con el fin de indicar el orden de la acción, es decir, la primera columna es la primera acción y continúa de forma consecutiva.

Lectura

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
i-nodo																
bloque																

Escritura

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
i-nodo																
bloque																

¿Cuál es el valor de **datos** tras la lectura?

--	--	--	--	--	--	--	--

¿Que datos hay en la zona de escritura antes de escribir?

--	--	--	--	--	--	--	--

Bloques

En esta tabla las letras representan directamente las letras correspondientes y los números no son caracteres ASCII sino valores. Esto hace que no sea necesario hacer la traducción de exadecimal a valores o a caracteres según corresponda.

Bloque																
1	2	h	o	m	3	e	t	c	4	v	a	r	0	0	0	0
2	18	e	l	o	19	a	n	a	0	0	0	0	0	0	0	0
3	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
4	16	f	c	a	17	f	c	d	0	0	0	0	0	0	0	0
5	45	d	k	e	r	t	s	t	s	e	i	q	t	u	s	g
6	a	n	a	r	o	d	t	q	y	z	x	g	a	7	4	6
7	5	e	l	o	6	f	v	a	7	f	u	s	8	f	c	c
8	3	9	12	0	0	0	0	0	0	0	0	0	0	0	0	0
9	q	r	s	t	u	v	w	x	y	z	1	2	3	4	5	6
10	6	12	3	21	18	0	0	0	0	0	0	0	0	0	0	0
11	12	c	e	s	a	u	l	f	23	f	u	r	0	0	0	0
12	g	f	a	r	y	u	s	r	t	s	t	y	q	u	0	0
13	9	a	n	a	0	0	0	0	0	0	0	0	0	0	0	0
14	21	e	t	s	f	23	1	r	u	o	3	d	p	l	d	r
15	a	a	a	a	d	d	d	f	f	z	t	d	l	f	s	r
16	18	f	c	a	0	0	0	0	0	0	0	0	0	0	0	0
17	10	f	n	a	0	0	0	0	0	0	0	0	0	0	0	0
18	1	r	s	k	r	s	5	9	f	21	g	h	32	s	54	s
19	3	5	6	7	8	a	n	d	34	d	5	t	t	a	f	i
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	19	15	11	0	0	0	0	0	0	0	0	0	0	0	0	0
22	5	3	d	t	u	0	0	0	0	0	0	0	0	0	0	0
23	14	f	a	a	15	f	c	b	0	0	0	0	0	0	0	0
24	18	14	21	0	0	0	0	0	0	0	0	0	0	0	0	0
25	e	3	2	t	o	f	h	0	0	0	0	0	0	0	0	0
26	5	f	e	z	s	4	9	a	n	a	4	t	d	6	7	e
27	12	f	l	o	13	f	n	a	0	0	0	0	0	0	0	0
28	s	r	u	i	o	s	y	u	o	e	u	s	t	d	7	z
29	23	f	a	g	t	e	4	e	g	g	g	e	s	d	u	i
30	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	a	c	k	q	t	6	w	6	d	t	y	l	z	3	5	s
32	a	v	x	3	f	5	6	s	4	d	p	j	l	k	f	e
33	12	d	s	t	2	t	4	s	5	6	8	d	g	e	p	l
34	12	s	x	s	0	0	0	0	0	0	0	0	0	0	0	0

35	s	r	s	z	a	t	d	r	s	4	2	23	s	4	g	g
36	33	1	5	6	s	t	c	f	u	d	x	d	t	d	x	s
37	11	a	n	a	0	0	0	0	0	0	0	0	0	0	0	0
38	a	15	6	t	h	d	56	s	z	o	r	21	s	t	u	8
39	q	r	6	34	d	t	y	s	f	0	0	0	0	0	0	0
40	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	19	f	w	w	0	0	0	0	0	0	0	0	0	0	0	0
43	1	5	6	s	g	r	u	i	o	z	0	0	0	0	0	0
44	a	s	i	b	m	o	p	l	0	0	0	0	0	0	0	0
45	q	r	y	u	i	o	p	d	d	g	d	k	x	z	t	6
46	4	t	g	c	o	a	t	0	0	0	0	0	0	0	0	0

i-nodos

En los i-nodos sólo se indican los punteros y no se representa la información adicional del principio como el tamaño o los timestamps. De esta forma en valor superior de cada tabla es el número de i-nodo, los dos siguientes son los dos punteros directos y el último es el puntero indirecto simple. Si es un directorio o un fichero vendría indicado en la información adicional aunque en nuestro caso todos serán directorios salvo que empiecen por f que serán ficheros de datos.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	7	27	17	23	29	22	20	16	37	42	38	33	31	28	43
0	13	0	0	0	0	0	0	0	0	0	36	35	32	26	0
0	0	0	0	0	0	0	0	0	0	0	41	40	30	24	0
17	18	19													
44	5	45													
0	6	46													
0	8	8													

16. El departamento de Ingeniería Informática de la universidad MAU tiene una máquina multiusuario con un sistema operativo tipo UNIX, donde los bloques de disco son de tamaño 1 kB y las direcciones a bloque son de 4 bytes. La estructura del i-nodo tiene espacio para 10 punteros directos y 3 indirectos repartidos como uno indirecto simple, otro indirecto doble y el tercero un indirecto triple. En dicha máquina cada profesor del departamento tiene asignado un directorio que cuelga directamente del /home.

En el directorio /home/Corchado/SOPER_2016/Notas/ está ubicado el fichero de notas NOTAS_SOPER.xlsx.

El profesor Corchado ejecuta el comando `ls -lis NOTAS_SOPER.xlsx` y obtiene la siguiente información

```

i-node  blocks      flags      owner      group      bytes      file
182     260      -rw-rw-rw- Corchado   Ing_Inf    265785     NOTAS_SOPER.xlsx

```

Responde a las siguiente cuestiones:

- Tamaño máximo del sistema de ficheros de la máquina del departamento.
- Tamaño máximo de un fichero en este sistema de ficheros.
- Representa gráficamente cuál es la secuencia de i-nodos y bloques que se han tenido que consultar partiendo del directorio /home para poder abrir el fichero NOTAS_SOPER.xlsx. Supón que la caché está vacía y que el i-nodo del directorio /home está en memoria. Considera que cada directorio ocupa un bloque.
- Una vez abierto el fichero NOTAS_SOPER.xlsx, se quiere acceder a la posición 160002 de ese fichero. ¿Cuántos acceso a disco se tiene que hacer para llegar a dicha posición?. Razona la respuesta.

17. Se tiene la tabla FAT indicada a continuación y el directorio /home/ana/ficheros contiene la información que también se presenta a continuación.

FAT	
1	15
2	48
3	18
4	49
5	2
6	7
7	16
8	17
9	22
10	31
11	9
12	6
13	EOF
14	EOF
15	EOF
16	47
17	28
18	45
19	EOF
20	27
21	46
22	40
23	36
24	EOF
25	14
26	25
27	24
28	4
29	EOF
30	EOF
31	23
32	29
33	34
34	19
35	44
36	3
37	43
38	30
39	42
40	41
41	43
42	1
43	38
44	37
45	26
46	11
47	33
48	10
49	39

Directorio	
8	datosbuenos.dat
5	datosmalos.dat
21	datosbrutos.dat
35	figura.png
12	articulo.pdf

Si cada bloque es de 512 B determina qué bloques y en qué orden ocupa cada fichero y su tamaño mínimo y máximo de acuerdo con estos datos.

Fichero	Mínimo	Máximo	Bloques en orden
datosbuenos.dat			
datosmalos.dat			
datosbrutos.dat			
figura.png			
articulo.pdf			

18. Se implementa ext3 en un disco duro con bloques de 4 kB. En ext3, la estructura de un i-nodo que siempre ocupa un bloque, es la siguiente:

- El tamaño del fichero en bytes. Ocupa 6 bytes.
- Un identificador del dispositivo. Ocupa 6 bytes.
- Un identificador del usuario. Ocupa 2 bytes.
- Un identificador del grupo. Ocupa 2 bytes.
- Los permisos y tipo del fichero. Ocupa 2 bytes.
- Varias marcas de tiempo:
 - Última modificación del i-nodo. Ocupa 4 bytes.
 - Última modificación del fichero. Ocupa 4 bytes.
 - Último acceso al fichero. Ocupa 4 bytes.
 - Instante en el que se ha borrado el fichero. Ocupa 4 bytes.
- Cuenta de enlaces. Ocupa 2 bytes.
- El resto son punteros directos y un puntero indirecto, un indirecto doble y uno triple. Cada puntero ocupa 4 bytes.

Responde a las siguientes preguntas. **Explica los cálculos según los vayas realizando (sin explicación los resultados no tendrán valor).**

A. Calcula el tamaño máximo del disco según esta estructura.

Solución:

El tamaño máximo del disco duro viene determinado por el número de bloques direccionables. Como los punteros a los bloques son de 4 B entonces puedo direccionar $2^{4 \times 8}$, es decir 2^{32} bloques. Como cada bloque es de 4 kB (2^{12} B) entonces el tamaño máximo direccionable de un disco con este sistema de archivos es de $2^{32} \times 2^{12}$ B, es decir **16 TB**.

B. Calcula el tamaño máximo de un fichero.

Solución:

Sumando los valores correspondientes el tamaño de la parte informativa del inodo ocupa 36 bytes. De esta forma quedan disponibles $4096 - 36$ B para punteros, es decir 4060 B. Como los punteros ocupan 4 B entonces tengo 1015 punteros, de los cuales 1012 son directos.

En cada bloque de punteros caben 4 kB / 4 B punteros, es decir, 1024 punteros. de esta forma nos queda lo siguiente:

- **Punteros directos:** $1012 \text{ punteros} \times 4 \text{ kB}$. Es decir, el tamaño máximo de un fichero con punteros directos es de 4145152 B.
- **Puntero indirecto simple:** es un puntero que apunta a un bloque de punteros que apuntan a bloques de datos y por tanto se dispone de 1024 punteros a bloques de 4 kB. Es decir, el fichero podría ocupar con el indirecto simple lo que ocupa con directos más 4 MB.
- **Puntero indirecto doble:** es un puntero que apunta a un bloque de punteros indirectos simples. Es decir permite acceder a 1024 punteros indirectos simples o lo que es lo mismo 1024×4 MB. Por tanto un fichero podría ocupar lo mismo que en el caso anterior más 4 GB.
- **Puntero indirecto triple:** es un puntero que apunta a un bloque de punteros indirectos dobles. Es decir permite acceder a 1024 punteros indirectos dobles o lo que es lo mismo 1024×4 GB. Por tanto un fichero podría ocupar lo mismo que en el caso anterior más 4 TB.