



Universidad de
Huelva



Escuela Técnica
Superior de
Ingeniería
ETSI
Grado en Informática

Diseño y Estructura de los Sistemas Operativos

Problemas de Sistemas de Ficheros (Soluciones)

Miguel Ángel Vélez Vélez

José Ponce González

Huelva, diciembre de 2025

Diseño y Estructura de los Sistemas Operativos

Problemas de Sistemas de Ficheros (Soluciones)

Autores:

MIGUEL ÁNGEL VÉLEZ VÉLEZ

JOSÉ PONCE GONZÁLEZ

ISBN: XXXXXXXXX

Usted es libre de:



copiar, distribuir y comunicar públicamente la obra



Hacer obras derivadas

Bajo las condiciones siguientes:



Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).



No comercial. No puede utilizar esta obra para fines comerciales.

- Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.
- Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor
- Nada en esta licencia menoscaba o restringe los derechos morales del autor.

Esto es un resumen del texto legal (la licencia completa) disponible en <http://creativecommons.org/licenses/by-nc/2.5/es/legalcode.es>

TABLA DE CONTENIDOS

1. A falta de pan, buenas son tortas.....	2
1.1. Solución.....	4
2. El de siempre.....	5
2.1. Solución.....	6
3. Nada Tiene Fácil Solución.....	9
3.1. Solución.....	10
4. Enlaces inFATigables.....	11
4.1. Solución.....	12

1. A falta de pan, buenas son tortas

Disponemos de un disco duro con bloques de 1 Kb que está dividido en dos particiones de 256 MB, cada una de las cuales tiene instalada un sistema de ficheros tipo ntfs y ext2 respectivamente.

Del sistema tipo ntfs sabemos:

- Que el boot ocupa 1 bloque
- La MFT ocupa un total de 32 Megas.
- Las cabeceras de las entradas de la MFT ocupan 2 bytes
- Los nombre de los ficheros ocupan como máximo 256 caracteres.
- Los descriptores de seguridad 6128 bits.
- La información estándar ocupa 1 Kb.
- Se puede almacenar en la MFT ficheros de hasta 6 Kb.
- Las 16 primeras entradas de la MFT están ocupadas.
- El tamaño máximo de una extensión es de 16 Mb.
- No hay mapas de bits.

Se pide:

- ¿Cuántos ficheros como máximo se pueden almacenar en este sistema de ficheros?
- En cada entrada de la MFT, ¿información sobre cuántas extensiones se puede almacenar?
- Si la información de un fichero sólo pudiera ocupar una entrada de la MFT, ¿cuál es el tamaño máximo de un fichero?

Del sistema tipo unix se sabe:

- El boot ocupa un bloque
- La tabla de i-nodos ocupa un total de 32 Mbytes.
- No hay mapas de bits
- Los i-nodos dedican 64 bits al tamaño del archivo y 18 bits a los apuntadores a bloques.
- El i-nodo dispone de 8 apuntadores directos, uno indirecto simple y otro indirecto doble.
- En cada i-nodo hay un campo de 16 bits que se usa para almacena su identificador.
- Además de toda esta información, en cada i-nodo hay 32508 bits adicionales que se usan para guardar las fechas de modificación, ultimo acceso, usuarios propietarios etc.

Se pide:

- ¿Cuántos ficheros como máximo se pueden almacenar en este sistema de ficheros?
- ¿Cuál es el tamaño máximo de un fichero?

1.1. Solución

1. Para el sistema tipo NTFS tenemos que calcular el número de entradas que contiene la MFT. Sabemos que cada entrada ocupa 1 Kb (de la información estándar) + 256 bytes (del nombre del fichero, 1 carácter → 1Byte) + 6132 bits (que pasados a Bytes serían 766, correspondientes a los descriptores de los ficheros) + 6 Kbytes (6144 Bytes, del área de datos) + tamaño de cabecera (2 Bytes). Así pues cada entrada ocupa 8192 Bytes (8 KB).

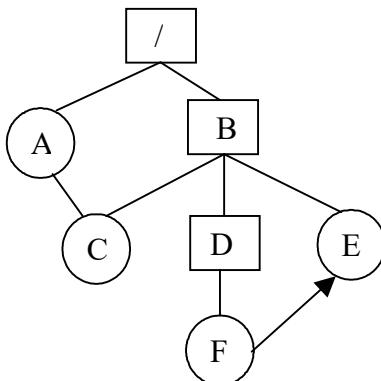
Dado que tenemos 32MB para la MFT (32768 KBytes), el número de entradas que tiene la MFT es de 4096. Como las 16 primeras entradas están ocupadas, tenemos un total de 4080 entradas que podrán ser usadas para almacenar información de archivos.

2. Cada extensión viene definida por dos números que identificar el número de bloque de comienzo y el número de bloques que la forman. Como en el disco hay un total de 262144 bloques, necesitamos 18 bits para hacer referencia a uno de ellos y 14 bits más para codificar el tamaño de cada extensión. Así pues, la información de cada extensión se guarda en 32 bits (18+14). En cada entrada de la MFT disponemos de 6 Kb para almacenar esta información, en dicho espacio somos capaces de almacenar 6 Kb / 32 bits =1536 extensiones.
3. El tamaño máximo del fichero sería de 1536 extensiones * 16 Mb =24576 Mb que es mucho mayor al tamaño libre que queda en disco que es de aproximadamente 224 Megabytes (ya que habría que descontar el tamaño del bloque destinado al Boot que sería un KB menos).
4. Primero tenemos que calcular el tamaño de un i-nodo. Este es igual a 64 bits + 18*10 bits + 32508 bits + 16 bits= 32768 bits (4 Kbytes por inodo). Sabemos que hay 32MB para la tabla de inodos por lo que resulta 2^{13} inodos. Por tanto puede haber hasta 8192 i-nodos = número máximo de ficheros.
5. El tamaño máximo del fichero puede estar limitado por distintos aspectos:
 1. Por el tamaño del campo tamaño del fichero = 2^{64} bloques
 2. Por el tamaño del espacio disponible en disco = aproximadamente 224 Megabytes= 229376
 3. Por la cantidad de bloques a los que pueden apuntar un i-nodo = $8 + 455 + 455^2 = 207488$ bloques.

En este caso el menor número es el de bloques apuntados por el i-nodo

2. El de siempre

Tenemos un disco con 256 pistas, 16 sectores y 2 caras. El tamaño del sector es de 256 bytes. El último bloque de dicho disco es el 1023. En dicho disco hay que instalar los siguientes ficheros:



Los Registros Lógicos son de 1 Kbyte. Entre A y C hay un *hard-link*, y F es un *soft-link* a E.

1. Dibujar la estructura del sistema de ficheros, suponiendo que tenemos FAT16, que el boot ocupa tres bloques y que el fichero A tiene 15 registros lógicos y el fichero E tiene 8 registros lógicos.
 - a) ¿Que tamaño ocupa la FAT?
 - b) ¿Que tamaño ocuparía la FAT si se implementase con FAT32?
 - c) Especifica la secuencia de bloques de disco a los que se accede para leer el registro 1 del fichero E. (Se deben indicar los números de bloque).
2. Dibujar la estructura del sistema de ficheros anterior, suponiendo que tenemos un sistema tipo LINUX, donde mantenemos el tamaño de bloque, tenemos un Boot de un bloque, el Superbloque ocupa 2 Kbytes (en él se encuentran los mapas de bits de datos y de inodos), hay 24 inodos y éstos tienen 246 bytes de INFO, 3 apuntadores directos, 1 indirecto simple y 1 indirecto doble (no hay indirecto triple). Los apuntadores son de 16 bits.
 - a) Especifique la secuencia de bloques de disco a los que se accede para leer el registro 8 del fichero F. (Se deben indicar los números de bloque).
 - b) Especifique la secuencia de bloques de disco a los que se accede para leer completo el fichero A. (Se deben indicar los números de bloque).

2.1. Solución

El tamaño del disco es: $256 * 16 * 2 * 256 = 2097152$ bytes = 2048 Kbytes = 2 Mbytes

Puesto que el último bloque es el 1023, hay 1024 bloques (desde 0 a 1023). Como el disco es de 2048 Kbytes, cada bloque ocupa 2 Kbyte.

Puesto que los registros lógicos ocupan 1 Kbyte, hay 2 registros lógicos por bloque.

- Dado que A tiene 15 registros lógicos necesita 8 bloques.

Dado que E tiene 8 registros lógicos necesita 4 bloques.

La FAT y los bloques serán:

FAT		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	1023
X	X	X	X	X	X	X	7	8	9	10	11	12	13	eof	eof	eof	17	18	19	eof	eof	free	

BOOT	BOOT	BOOT	FAT	Copia	FAT	A	Dat	6	B	Dir	14										C	Dat	6
0	1	2	3	4	5	A/C		"	A/C		13	14	D	Dir	15	E	Dat	16					

F	Link	20	E	E	/B/E																	
15	16	19	20																			1023

1.a) La FAT tiene 1024 pos. de 2 bytes, por tanto ocupa: $1024 * 2 = 2048$ bytes = 1 bloque

1.b) La FAT tendría 1024 pos. de 4 bytes, por tanto: $1024 * 4 = 4096$ bytes = 2 bloques

1.c) Los accesos serían:

1. Bloque del directorio RAIZ (Bloque 5)
2. Bloque del directorio B (Bloque 14)

3. Bloque del Reg. 1 de E (Bloque 16)

2. Puesto que el bloque es de 2 Kbytes, y los apuntadores de 16 bits, caben 1024 apuntadores por bloque.

A necesita 8 bloques de datos.

E necesita 4 bloques de datos.

La estructura del sistema de ficheros queda:

Boot	Superbloque	Tabla de I-nodes	2 3 4 5 6 7 8 9	A B 	A/C	A/C	A/C	10 11 12 13 14
------	-------------	------------------	--------------------------------------	------------	-----	-----	-----	----------------------------

A/C	A/C	A/C	A/C	A/C	2	C	6	F	E	E	E
					4	D					
					5	E					
10	11	12	13	14	15	16	17	18	19		

21		/B/E							
20	21	22							1023

Tabla de I-nodes

/	A/C	B	D	E	F		
5	6	15	16	17	22		
	7			18			
	8			19			
	9			20			
1	2	3	4	5	6		24

2.a) Para leer el registro lógico 8 de F:

1. i-nodo del directorio RAIZ (Bloque 2)
 2. Bloque del directorio RAIZ (Bloque 5)
 3. i-nodo de B (Bloque 2)
 4. Bloque de B (Bloque 15)
 5. i-nodo de D (Bloque 2)

6. Bloque de D (Bloque 16)
7. i-nodo de F (Bloque 2)
8. Bloque de F (Bloque 22)
9. i-nodo del directorio RAIZ (Bloque 2)
10. Bloque del directorio RAIZ (Bloque 5)
11. i-nodo de B (Bloque 2)
12. Bloque de B (Bloque 15)
13. i-nodo de E (Bloque 2)
14. Bloque de apuntadores indirectos simple (Bloque 20)
15. Bloque de datos 4º de E (Bloque 21)

2.b) Para leer el registro lógico 15 de A:

1. i-nodo del directorio RAIZ (Bloque 2)
2. Bloque del directorio RAIZ (Bloque 5)
3. i-nodo de A (Bloque 2)
4. 1º y 2º Reg. Lógico de A (Bloque 6)
5. 3º y 4º Reg. Lógico de A (Bloque 7)
6. 5º y 6º Reg. Lógico de A (Bloque 8)
7. Bloque de apuntadores indirectos simple (Bloque 9)
8. 7º y 8º Reg. Lógico de A (Bloque 10)
9. 9º y 10º Reg. Lógico de A (Bloque 11)
10. 11º y 12º Reg. Lógico de A (Bloque 12)
11. 13º y 14º Reg. Lógico de A (Bloque 13)
12. 15º Reg. Lógico de A (Bloque 14)

3. Nada Tiene Fácil Solución

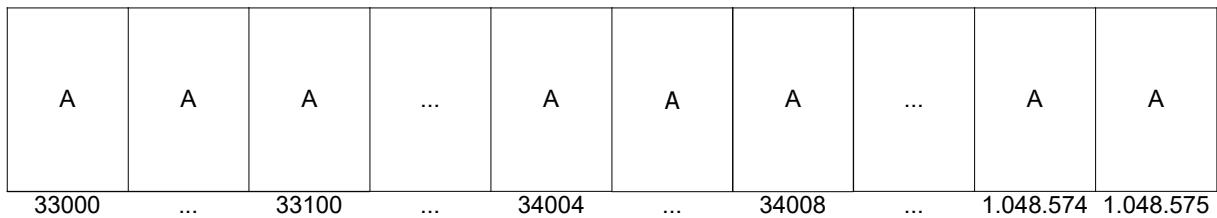
Disponemos de un disco duro con bloques de 1 Kb que está dividido en dos particiones de 1 Gbyte cada una. En la primera partición tenemos instalado un sistema de ficheros tipo NTFS.

Del sistema tipo NTFS sabemos:

- No tiene Boot
- La MFT ocupa un total de 32 Mbytes.
- Las cabeceras de las entradas de la MFT ocupan 32 bytes
- Los nombres de los ficheros ocupan como máximo 512 bytes.
- Los descriptores de seguridad tienen 9696 bytes.
- La información estándar ocupa 6 Kbytes.
- Se puede almacenar en la MFT ficheros de hasta 16 Kbytes.
- Las 16 primeras entradas de la MFT están ocupadas.
- El tamaño máximo al que puede llegar una extensión es de 64 Mbytes.

Se pide, explicando razonadamente cada uno de los apartados:

1. ¿Cuántos ficheros como máximo se pueden almacenar en este sistema de ficheros?
2. En cada entrada de la MFT, ¿cuántas extensiones se puede almacenar?
3. Si la información de un fichero sólo pudiera ocupar una entrada de la MFT, ¿cuál es el tamaño máximo de un fichero?
4. Si en el sistema sólo está el directorio / y un fichero A que se encuentra en el disco en los siguientes bloques, rellena todo lo que puedas de la MFT:



3.1.Solución

- Para el sistema tipo NTFS tenemos que calcular el número de entradas que contiene la MFT. Calculamos el tamaño de cada entrada:

cabecera: 32 bytes

información estandard: 6 Kybtes = 6144 bytes

nombre del fichero: 512 bytes

descriptor de seguridad: 9696 bytes

datos: 16 Kbytes = 16384 bytes

Total = 32768 bytes = 32 Kbytes

La MFT tiene un total de 32 Mbytes = 32768 Kbytes

Si dividimos el tamaño de la MFT entre el tamaño de cada entrada: 32768 Kbytes / 32 Kbytes = 1024 entradas

Como las 16 primeras entradas están ocupadas, tenemos un total de 1008 entradas para poder almacenar un número máximo de 1008 ficheros.

- Cada extensión viene definida por dos números que identifican el número de bloque de comienzo y el número de bloques que la forman. Como en el disco hay un total de 1.048.576 bloques, necesitamos 20 bits para hacer referencia a uno de ellos y 16 bits más para codificar el tamaño de cada extensión. Así pues, la información de cada extensión se guarda en 36 bits (20+16). En cada entrada de la MFT disponemos de 16 Kb para almacenar esta información, en dicho espacio somos capaces de almacenar 16 Kb / 36 bits = 3640 extensiones.
- El tamaño máximo del fichero sería de 3640 extensiones * 64 Mbytes = 232.960 Mbytes = 227'5 Gbytes

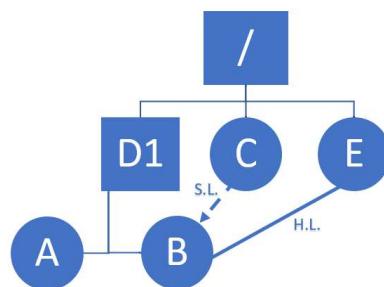
4. MFT

Cabec.	Inf. Est.	Nom. F.	D.S.	Datos									
5		/		16									
16		A		32768	4	33000	101	34004	5	104857 4	2		

4. Enlaces inFATigables

Tenemos un disco de 32 MBytes, con 32 sectores por pista, 256 cilindros y 4 superficies. El tamaño del bloque es el doble que la del sector. Además de esto, tenemos:

- El Boot tiene un tamaño de 4096 Bytes
- El sistema trabaja con FAT32
- Los registros lógicos tienen un tamaño de 256 Bytes
- El fichero E tiene 800 registros lógicos, comenzando su primer bloque en el 69 y el resto contiguos a éste
- El fichero A tiene 127 registros lógicos, comenzando su primer bloque en el 200 y el resto contiguos a éste



- 1) Teniendo en cuenta los datos descritos arriba, y recorriendo el árbol de arriba en **anchura**, dibuje cómo quedaría el disco y la estructura FAT
- 2) Dado el apartado anterior, se crea un directorio D2 que cuelga del directorio raíz (cuya información está en el bloque 169) y a continuación, dentro de este se crea un fichero F que es un soft link del fichero C. El bloque o bloques de F se establecerán a continuación del último bloque del fichero A. Dibuje los bloques del disco y posiciones de la FAT que se necesiten para aplicar los cambios anteriores. **Únicamente** dibuje aquellos bloques de disco y posiciones de la FAT que se vean modificados para aplicar los cambios.

Teniendo en cuenta **la solución del primer apartado**, resuelva los siguientes:

- 3) Dadas las dos estructuras vistas en el tema para gestionar el espacio libre (mapa de bits y listas enlazadas), indicar el tamaño en Kbytes que se necesitaría para cada una de ellas.
- 4) Teniendo en cuenta que la FAT está cargada en memoria, detalle los accesos necesarios a los ficheros indicados en los siguientes puntos:
 1. Acceso al registro lógico 30 del fichero C
 2. Recorrer completamente el fichero A

4.1.Solución

Con los datos que tenemos podríamos calcular inicialmente el número de sectores = $32 \times 25 \times 4 = 32768$ sectores. Teniendo el tamaño del disco 32 MB (32768 KB), si lo dividimos entre el número de sectores (32768), tendríamos el tamaño de un sector, 1KB. Del mismo modo nos indican que el tamaño del bloque es el del doble que el del sector, por tanto, el tamaño de bloque, 2KB. Sabiendo el tamaño de bloque, y conociendo de antes el tamaño del disco, conocemos el número de bloques del disco, que serían 16.384 bloques.

El BOOT, al tener 4096 Bytes y cada bloque saber que son 2048 Bytes, ocuparía 2 bloques.

Con FAT32, tenemos apuntadores de 32 bits por lo que si tenemos 32 bits (4 Bytes) para apuntar a cada uno de los bloques y tenemos 16384 bloques, $16384 * 4B = 65536$ Bytes -> 64 KBytes -> (32 bloques cada copia FAT)

Para conocer los bloques de cada fichero, sabiendo que el tamaño de un registro lógico es de 256 Bytes, tenemos 8 RRLL / bloque. El fichero E tiene 800 RRLL por lo tanto, necesitaremos 100 bloques (contiguos a partir del bloque 69). El fichero A tiene 127 RRLL, por lo que se necesitarían 15,8 bloques, por tanto 16 bloques (contiguos a partir del 200).

En rojo y negrita se indica la solución al segundo apartado.

BOOT	1 ^a	2 ^a	.	DIR	66	.	DIR	67				.	DIR	169			
	CO	CO	..	DIR	66	..	DIR	66				..	DIR	66			
	PI	PIA	D1	DIR	67	A	DAT	200				F	LINK	216			
	A	FAT	C	LINK	68	B	DAT	69	/D1/B	E/B	E/B						
	FA		E	DAT	69												
	T		D2	DIR	169												
0-1	2..33	34..65	66			67			68	69	168	169		169	17	...	
																	0

	A		A	IC				
199	200	...	215	216	217	...	16383	

FAT

X	X	X	EOF	EOF	EOF	70	71		168	EOF	EOF			201	202		215	EOF	EOF				
0-1	2..33	34..65	66	67	68	69	70	...	167	168	169	170	...	199	200	201	...	214	215	216	217	...	16383

Apartado 2) Marcado en el apartado anterior, en rojo y negrita.

Los dos apartados de a continuación, se resuelven en función al primero (sin incluir los cambios del segundo apartado)

Apartado 3)

Con respecto al mapa de bits, se necesitarían tanto bits como bloques tenemos, por tanto 16384 bloques = 16384 bits. Esto sería, 2KBytes. Por tanto, un bloque.

	1	0	0	1	1	0	0
.	1	1	.	1	2	.	2
.	6	6	.	9	0	.	1
.	8	9	.	9	0	.	5

Con respecto a las listas enlazadas, como tenemos apuntadores de 32 bits (4 Bytes), en cada bloque cabrían 512 apuntadores (de los cuales, el último se perdería para apuntar al siguiente bloque de apuntadores a bloques libres). Por tanto útiles en cada bloque para apuntar a bloques libres habría 511 apuntadores por bloque.

Para saber el número de bloques libres del sistema, restaremos al total de los bloques (16.384), los ocupados por el BOOT, copias de FAT, directorios, enlaces y bloques de datos. Esto sería, viendo el mapa de bits anterior, $16384 - 31$ (bloques libres del 169 al 199) – 16168 (bloques libres del 216 al 16383), por tanto 16199 bloques libres.

Si tenemos 16199 bloques libres y por cada bloque podemos direccionar 511 bloques, tendríamos que necesitaríamos 31,7 bloques, por tanto, 32 bloques. Estos 32 bloques, supondrían 64 KBytes de espacio en disco (ya que cada bloque tiene 2 KBytes de tamaño).

Si tuviéramos en cuenta que, estos 32 bloques de apuntadores estarían también ocupados, pues serían $16199 - 32 = 16167$ bloques libres.

Que realizando la misma cuenta anterior, igualmente necesitaríamos 32 bloques, es decir 64 Kbytes.

Apartado 4) Accesos a ficheros, con FAT cargada en memoria

Tal y como calculamos en el primer apartado, hay 8 RR.LL. por cada bloque.

Acceso al registro lógico 30 del fichero C (en el 4º bloque)	Recorrido completo del fichero A
1.- Bloque del directorio / (B66)	1.- Bloque del directorio / (B66)
2.- Leer bloque del fichero C, Soft Link (B68)	2.- Bloque del directorio D1 (B67)
3.- Bloque del directorio / (B66)	FAT cargada en memoria
4.- Bloque del directorio D1 (B67)	3.- 1º Bloque del fichero A (B200)
FAT cargada en memoria	4.- 2º Bloque del fichero A (B201)
5.- 4º Bloque del fichero A que contiene el RL30 (B72)	...
Por tanto, 5 accesos a disco	18.- Último bloque del fichero A (B215)
	Por tanto, 18 accesos a disco