



Universidad de
Huelva

• • • • • • • • • •
Escuela Técnica
Superior de
Ingeniería
ETSI
Grado en Informática

DISEÑO Y ESTRUCTURA DE LOS SISTEMAS OPERATIVOS

Problemas de Sistemas de Ficheros

Miguel Ángel Vélez Vélez

José Ponce González

Huelva, diciembre de 2025

Diseño y Estructura de los Sistemas Operativos

Problemas de Sistemas de Ficheros

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.

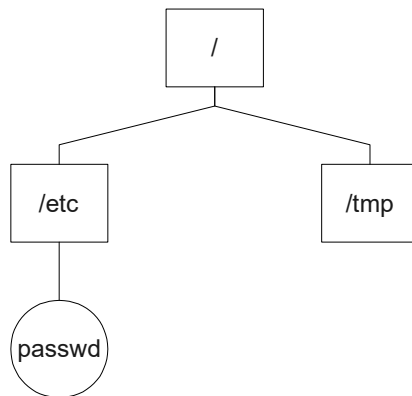
Esto es un resumen del texto legal (la licencia completa) disponible en <http://creativecommons.org/licenses/by-nc/2.5/es/legalcode.es>

TABLA DE CONTENIDOS

1. Pelegrin returns.....	2
1.1. Solución.....	5
2. Tapitas III.....	8
2.1. Solución.....	9
3. Enemigo Público.....	11
3.1. Solución.....	12
4. Nada Tiene Fácil Solución.....	14
5. Podado de árboles.....	16
6. Rebobinado manual.....	17
7. Yo me quedo en casa.....	19
8. Siguiendo el rastro.....	20
9. El de siempre.....	22
10. No Tengas FobiaS.....	23
11. A falta de pan buenas son tortas.....	24
12. Rastreando nueva variante.....	26
13. Enlace fatídico.....	27
14. No te fíes.....	28
15. Enlaces inFATigables.....	29
16. Mes de mudanza.....	30

1. PELEGRIN RETURNS

Para evitar los problemas informáticos acaecidos durante el desarrollo del espectáculo de luz y sonido “*Pelegrin returns*”, se ha contratado los servicios de la empresa Empa-nadas Flying S.L. que propone la instalación en todos los servidores que controlarán este acto de un disco duro con sectores de 1 KBytes y capacidad total de 80 GBytes dividido en cuatro particiones. En una de ellas se ha instalado un sistema de ficheros formado por un total de 16384 bloques siendo el tamaño total del mismo de 16 Mbytes. Si en dicho sistema de ficheros tenemos almacenados la siguiente estructura de ficheros:



Si de cada fichero/directorio conocemos la siguiente información:

Nombre	Primer Bloque	Tamaño en Bloques	Nombre	Primer Bloque	Tamaño en Bloques
/		1	/etc	15800	1
/tmp	14000	1	passwd		560

Si el sistema de ficheros fuera de tipo FAT:

1. ¿Qué tipo de FAT se debería usar?
2. ¿Cuánto ocuparía esta FAT?
3. ¿Cuál sería el tamaño máximo de un fichero regular si el sistema de ficheros estuviera vacío?
4. Rellene lo más posible la estructura FAT que se adjunta.

Si el sistema de ficheros fuera de tipo UNIX:

5. Rellene lo más posible la estructura UNIX adjunta.
6. ¿Cuál sería el tamaño máximo de un fichero? No tenga en cuenta las limitaciones del disco pero sí la metainformación (no incluya el espacio ocupado por el directorio, los mapas de bits y el i-nodo) necesaria para almacenarlo.
7. ¿Cuántos ficheros como máximo puede albergar?
8. Si se pretende acceder al byte 536000 del ficheros passwd, ¿cuántos accesos a disco se requieren? Haga los cálculos o bien para el sistema FAT o para el UNIX.

Notas:

- En todos los sistemas de ficheros se usa el mismo tamaño de apuntador usado por la FAT.

Estructuras a rellenar:

[illegible]

<i>I</i>	
.	
..	

Bloque:

/tmp	

Bloque:

/etc	

Bloque:

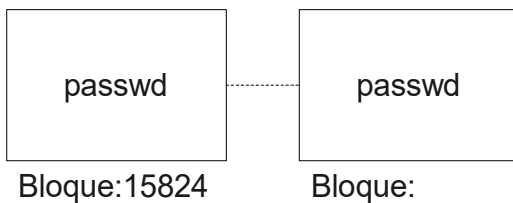
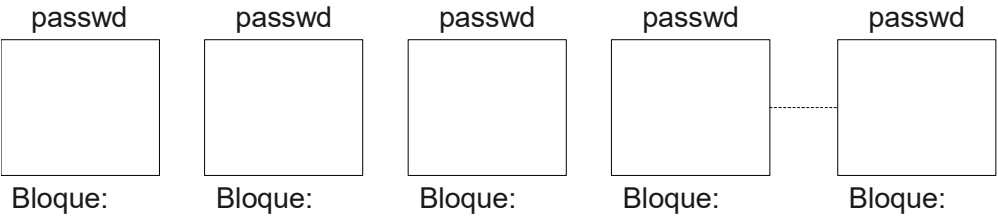
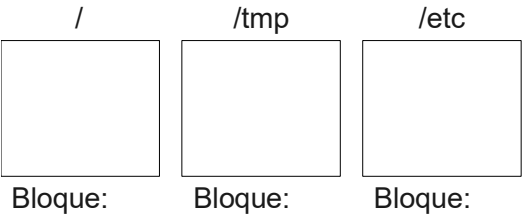


Tabla i-nodos

/		/tmp		/etc		passwd	
1		2		3		4	
						5	
						6	



1.1. Solución

1. Como la partición a redireccionar dispone de 16384 bloques de disco, la FAT que se tendría que usar es la 16 ya que esta permite direccionar hasta 65536 bloques.
2. El tamaño de la FAT es $16384 \text{ bloques} * 16 \text{ bits} / \text{bloque} = 31.99 \text{ KB} = 32 \text{ bloques}$.
3. El tamaño máximo de un fichero sería $16384 \text{ bloques} - 64 \text{ bloques por FAT} - 1 \text{ bloque por BOOT} - 1 \text{ bloque por directorio } /$, al total de bloques en disco tenemos que restar los bloques reservados y, hay que tener en cuenta, que dicho fichero regular tendrá que estar localizado en un directorio. En total el fichero ocuparía 16318 bloques.
4. Para el sistema de ficheros FAT

0	X
	X
	X
	:
64	X
65	EOF
14000	EOF
15800	EOF
15824	15825
15825	15826
15826	15827
15827	15828
15828	15829
15829	15830
15830	15831
15831	15832
15832	15833
...	...
16381	16382
16382	16383
16383	EOF

/	
.	65
..	65
etc	15800
tmp	14000

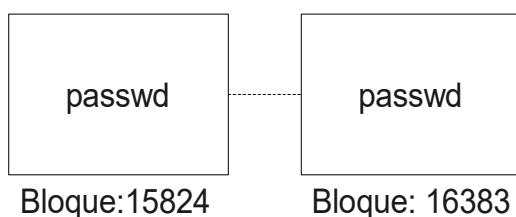
Bloque: 65

/tmp	
.	14000
..	65

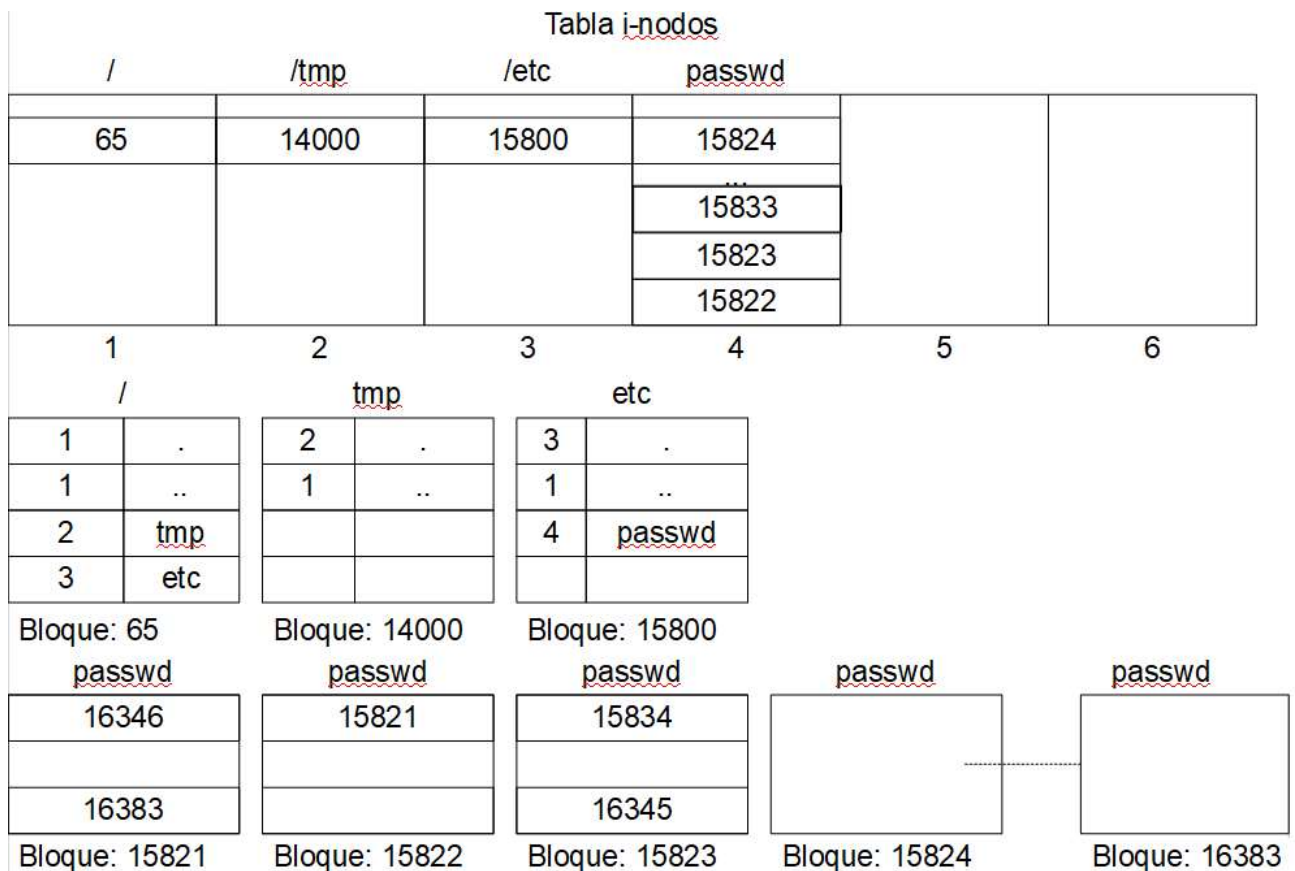
Bloque: 14000

/etc	
.	14000
..	65
passwd	15824

Bloque: 15800



5. Para el sistema de ficheros UNIX



6. Un bloque de disco ocupa 1 KB, como los apuntadores que usamos para redireccionar el disco son de 16 bits tenemos que en cada bloque podemos almacenar hasta 512 direcciones. Por tanto el tamaño máximo que ocupa un fichero en disco es: 10 bloques por apuntadores directos + (1 bloque por apuntador indirecto simple + 512 bloques apuntados por el apuntador directo) + (1 bloque por el apuntador indirecto doble + 512 bloques ocupados por apuntadores indirectos simples a los que apunta el indirecto doble + 512*512 bloques apuntados por los apuntadores directos) + (1 bloque por el apuntador indirecto triple + 512 apuntadores indirectos dobles apuntados por el apuntador indirecto triple + 512 * 512 apuntadores indirectos simples apuntados por los apuntadores indirectos dobles + 512*512*512 bloques apuntadores por los apuntadores directos). En total el espacio ocupado es: 134743565 bloques.
7. Como en la tabla de i-nodos tenemos 6 i-nodos sólo podremos crear hasta 6 ficheros.

8. Al acceder a la posición 53600 estamos accediendo al bloque 524 (536000 bytes / 1024 bytes). Para cada uno de los sistemas de ficheros los accesos a dar son:

1. FAT

1. Leo el bloque 65 (corresponde al /)
2. Leo el bloque 15800 (corresponde al directorio /etc)
3. Cargo la FAT
4. Leo el bloque 16384 (corresponde con el bloque 524 del fichero)

2. UNIX

1. Leo el i-nodo 1 (corresponde i-nodo /)
2. Leo el bloque 65 (directorio /)
3. Leo el i-nodo 3 (corresponde i-nodo /etc)
4. Leo el bloque 15800 (corresponde al directorio /etc)
5. Leo el i-nodo 4 (correponde al i-nodo de passwd)
6. Leo el bloque 15822 (corresponde al apuntador indirecto doble)
7. Leo el bloque 15821 (corresponde al 2 nivel del apuntador indirecto doble)
8. Leo el bloque 16347 que es el bloque 524 de dicho fichero

2. TAPITAS III

1. En la siguiente figura se representa una tabla FAT.

				FAT			
Nombre	T	Fecha	Blq				
Datos	F	8-2-90	3	1	X	10	
				2	X	11	
				3	EOF	12	
				4		13	
				5		14	
				6		15	
				7		16	
				8		17	
				9		18	

También se ha representado la entrada del directorio raíz. Como simplificación del ejemplo, suponemos que en cada entrada del directorio se almacena: Nombre de archivo/directorio, el tipo (F=archivo, D=directorio), la fecha de creación y el número del bloque inicial. Además, tenemos en cuenta que el tamaño de bloque es de 512 bytes. Represente cómo quedaría la FAT tras la siguiente secuencia de operaciones (los bloques se van ocupando de menor a mayor):

- a. Creación del archivo DATOS1 con fecha 1-3-90 y tamaño de 1200 bytes.
 - b. Aumento del tamaño del fichero Datos en 2 bloques más.
 - c. Creación del directorio D, con fecha 3-3-90.
 - d. Renombrar el fichero D por E.
2. Si usamos un Mapa de Bits para la gestión del espacio libre, especifique la sucesión de bits que contendría respecto a los bloques libre y ocupados del punto anterior.
 3. Suponga un sistema de archivos parecido al de Unix, cuyos bloques son de tamaños de 1 Kbyte, los apuntadores a disco de 32 bits y sin apuntador indirecto triple. Se quiere incrementar el tamaño máximo del fichero. Cuál de las siguientes acciones permitiría un mayor aumento: añadir un bloque indirecto triple o incrementar el tamaño del bloque en 4 Kb.

2.1. Solución

1. Antes de rellenar la FAT, hay que tener en cuenta que el fichero DATOS1 ocupa $1200 / 512 \text{ bytes} = 2,3 \text{ bloques} = 3 \text{ bloques}$.

a) Tras la creación del fichero DATOS1

Nombre				FAT			
Nombre	T	Fecha	Blq				
Datos	F	8-2-90	3	1	X	10	
DATOS1	F	1-3-90	4	2	X	11	
				3	EOF	12	
				4	5	13	
				5	6	14	
				6	EOF	15	
				7		16	
				8		17	
				9		18	

b) Tras el aumento de tamaño de Datos

Nombre				FAT			
Nombre	T	Fecha	Blq				
Datos	F	8-2-90	3	1	X	10	
DATOS1	F	1-3-90	4	2	X	11	
				3	7	12	
				4	5	13	
				5	6	14	
				6	EOF	15	
				7	8	16	
				8	EOF	17	
				9		18	

c) Tras la creación del directorio D

Nombre				FAT			
Nombre	T	Fecha	Blq				
Datos	F	8-2-90	3	1	X	10	
DATOS1	F	1-3-90	4	2	X	11	
D	D	3-3-90	9	3	7	12	
				4	5	13	
				5	6	14	
				6	EOF	15	
				7	8	16	
				8	EOF	17	
				9	EOF	18	

d) Tras renombrar D

				FAT			
Nombre	T	Fecha	Blq				
Datos	F	8-2-90	3	1	X	10	
DATOS1	F	1-3-90	4	2	X	11	
E	D	3-3-90	9	3	7	12	
				4	5	13	
				5	6	14	
				6	EOF	15	
				7	8	16	
				8	EOF	17	
				9	EOF	18	

- El mapa de bits tiene tantas entradas como bloques, 18, las 9 primeras entradas estarían a 1 para marcar que dichos bloques están ocupadas y el resto se quedaría a 0.
- Actualmente, el tamaño máximo que ocupa un fichero es:
 $10 + 256 + 256 * 256 = 65802 \text{ Bloques} * 1 \text{ Kbyte} = 65802 \text{ Kbytes} = 64,25 \text{ Mbytes}$

Si aumentamos el tamaño del bloque a 4 Kbytes el tamaño máximo del fichero sería
 $10 + 1024 + 1024 * 1024 = 1049610 \text{ Bloques} * 4 \text{ Kbytes} = 4198440 \text{ Kbytes} = 4100 \text{ Mbytes}$

Si lo que hacemos es mantener el tamaño del bloque pero añadimos apuntadores indirectos triples:

$10 + 256 + 256 * 256 + 256 * 256 * 256 = 16843018 \text{ Bloques} * 1 \text{ Kbyte} = 16843018 \text{ Kbytes} = 16448 \text{ Mbytes}$

Por tanto añadiendo apuntadores indirectos triples aumenta más el tamaño máximo de un fichero.

3. ENEMIGO PÚBLICO

Un disco de $\frac{1}{2}$ Gbyte de capacidad tiene instalado un sistema de ficheros tipo Unix. Las características de este sistema de ficheros son:

- El boot ocupa 2 Kbytes
- El superbloque ocupa 1 Kbyte (en el se encuentra el mapa de bits de i-nodos pero no el mapa de bits de bloques de datos)
- El tamaño de bloque es de 1Kbytes
- El tamaño del i-nodo es de 128 bytes
- Los apuntadores son 32 bits
- **No hay apuntador indirecto triple**
- El número máximo de i-nodos es 600

Se pide:

1. Indique la estructura del disco distinguiendo los bloques que ocupan cada una de las diferentes zonas del disco (No es necesario dibujar todo el disco).
2. ¿Cuál será el tamaño máximo de un fichero en este sistema?

Supongamos que la información de todos los directorios que existen en el disco es la siguiente:

1	.	2	.	3	.	8	.	9	.
1	..	1	..	1	..	3	..	3	..
2	A	4	D	8	G	4	I		
3	B	5	E	9	H	10	J		
14	C	6	F			5	K		
						15	L		

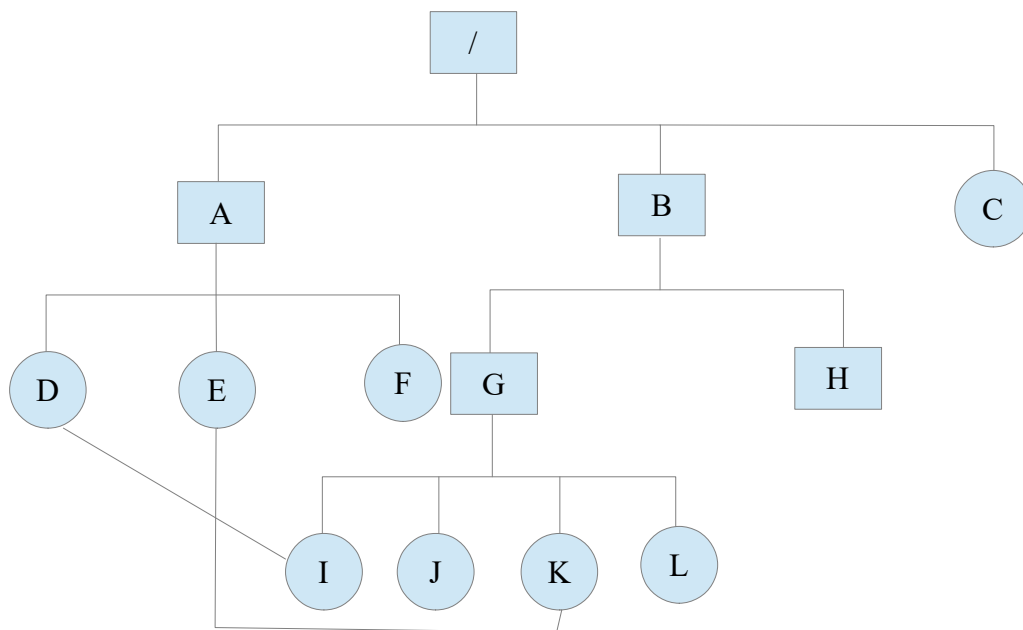
Todos los ficheros ocupan 1 Kbyte, excepto F que tiene 30 Kbytes.

Partiendo de dicha información, se pide:

3. Dibuje el árbol de directorios y ficheros indicando los enlaces que existan.
4. Rellene todo lo posible los mapas de bits.
5. ¿Cuántos accesos a disco se necesitarán para leer completamente el fichero F?
6. Convierta el sistema de ficheros a una FAT rellenando todos los bloques posibles y la FAT. Use la FAT real más adecuada. Indique el tamaño ocupado por las dos copias de la FAT. (Se supone que están el . y el .. en cada directorio).

3.1. Solución

1. El boot ocupa 2 bloques, el mapa de bits de los i-nodos ocupa 600 bits = 75 bytes (1 bloque), la tabla de i-nodos ocupa $600 \times 128 = 75$ Kbytes (75 bloques), el mapa de bits es de 512×1024 bits = 64 KB (64 bloques). El espacio reservado para los datos es de $524288 - 2 - 1 - 75 - 64 = 524146$ bloques.
2. Según la estructura del i-nodo, y teniendo en cuenta que en cada bloque entran $1024 \text{ bytes} / 4 = 256$ apuntadores, un fichero puede ocupar como máximo: $10 \text{ bloques} + 256 + 256^2 = 65802 \text{ bloques} = 65802 \text{ Kbytes}$.
3. La estructura de directorios y ficheros a la que da lugar es:



K. Existen dos enlaces duros: Fichero D y fichero I es uno de ellos. El otro entre E y

4. Con los datos obtenidos podemos rellenar la diversas estructuras de la manera siguiente:
 1. El mapa de bits pondría a uno los bloques ocupados por el boot, el superbloque, el mapa de bits y la tabla de i-nodos. Esto es, de la entrada 0 a la 141 todo iría a uno. Como en el enunciado no dice los números de bloques de datos ocupados no se pueden poner más unos.
 2. El mapa de bits de la tabla de i-nodos tendría a 1 las entradas: 1, 2, 3, 4, 5, 6, 8, 9 10, 14 y 15. El resto a cero.

5. Para leer completamente el fichero F necesito:

1. Leer i-nodo /
2. Bloque /
3. Leer i-nodo A
4. Bloque A
5. i-nodo F
6. 10 acceso para los primeros 10 bloques
7. 1 acceso para bloque indirecto simple
8. 20 accesos para los bloques que quedan
9. Un total de 36 accesos.

6. Como el disco duro es de 512 MB necesitamos punteros de 19 bits. La única FAT que cubre esta necesidad es la FAT32. Cada una de las tablas de la FAT ocupará 2048 bloques. Así pues los bloques 0 y 1 estarán ocupados por el boot, del 2 al 4097 estarán reservados para la FAT y el bloque 4098 será el del directorio raíz.

/			A			B			G			H		
.	D	4098	.	D	4099	.	D	4100	.	D	4105	.	D	4106
..	D	4098	..	D	4098	..	D	4098	..	D	4100	..	D	4100
A	D	4099	D	-	4102	G	D	4105	I	-	4102			
B	D	4100	E	-	4103	H	D	4106	J	-	4107			
C	-	4101	F	-	4104				K	-	4103			
									L	-	4108			

X		X	EO F	EOF	EOF	EOF	EOF	EOF	410 9	EOF	EOF	EOF	EOF	411 0		EOF	FRE E
0	...	409 7	409 8	409 9	410 0	410 1	410 2	410 3	410 4	410 5	410 6	410 7	410 8	410 9	...	4137	4138

4. NADA TIENE FÁCIL SOLUCIÓN

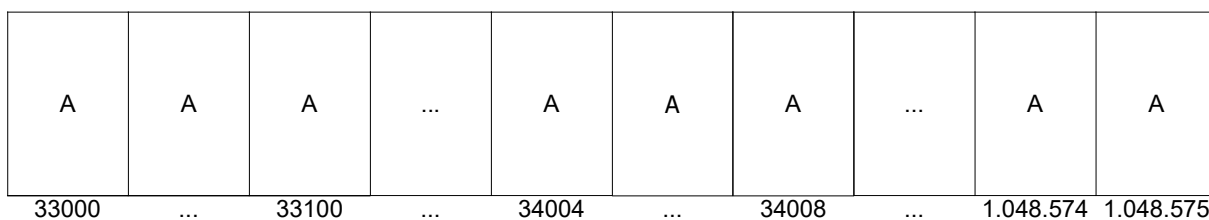
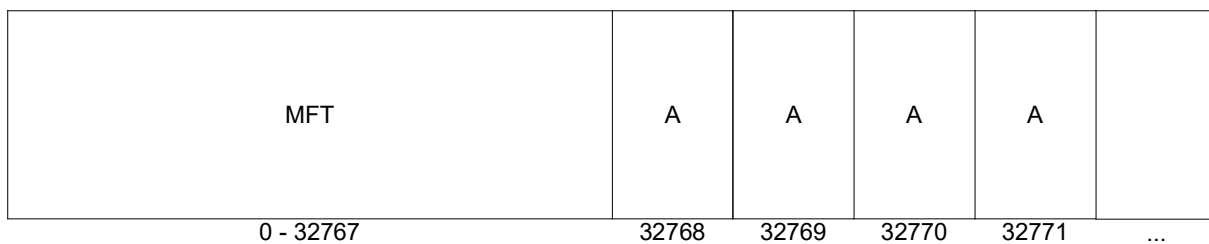
Disponemos de un disco duro con bloques de 1 Kb que está dividido en dos particiones de 1 Gbyte cada una. En la primera partición tenemos instalado un sistema de ficheros tipo NTFS.

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:

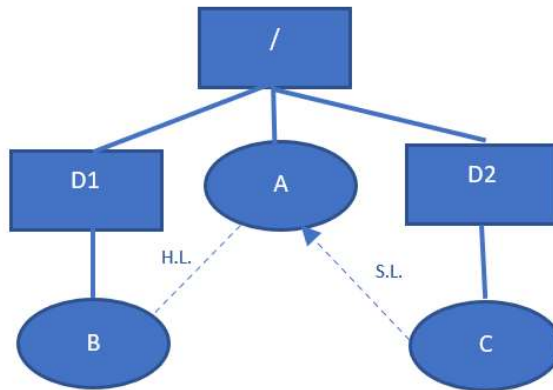
1. ¿Cuántos ficheros como máximo se pueden almacenar en este sistema de ficheros?
2. En cada entrada de la MFT, ¿cuántas extensiones se puede almacenar?
3. Si la información de un fichero sólo pudiera ocupar una entrada de la MFT, ¿cuál es el tamaño máximo de un fichero?
4. Si en el sistema sólo está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



5. PODADO DE ÁRBOLES

Un disco de 128 Mbytes, con 8 caras, 512 cilindros y 16 sectores por pista, tiene montado un sistema de ficheros Unix. El tamaño del bloque es el mismo que el del sector, los registros lógicos son de 1 Kbyte y los apuntadores son de 32 bits.

Si tenemos el siguiente árbol de ficheros y directorios:



Información Adicional		
Fich. / Dir.	Tam. (KB)	1 ^{er} bloque
/	2	13
D1	2	14
B	1300	60
D2	2	15
A	HARD LINK CON B	
C	SOFT LINK HACIA A	

Teniendo en cuenta el árbol de directorios, la información adicional de los mismos (todos los bloques de datos de los ficheros se encuentran en bloques contiguos) y la estructura de disco que determina la información de arriba, realizar los siguientes apartados:

1) Recorriendo el árbol en anchura, dibuje la estructura lógica del disco rellenando toda la información posible con los datos de que disponemos (incluido el mapa de bits de bloques y el mapa de bits de inodes), sabiendo que no hay boot, el superbloque ocupa 5 bloques (en dicho bloque se encuentra el mapa de bits de bloques y el mapa de bits de inodes), disponemos de 16 inodes y cada uno ocupa medio bloque. Los inodes tienen la estructura normal del mismo, esto es, 10 apuntadores directos, 1 indirecto simple, 1 indirecto doble y 1 indirecto triple.

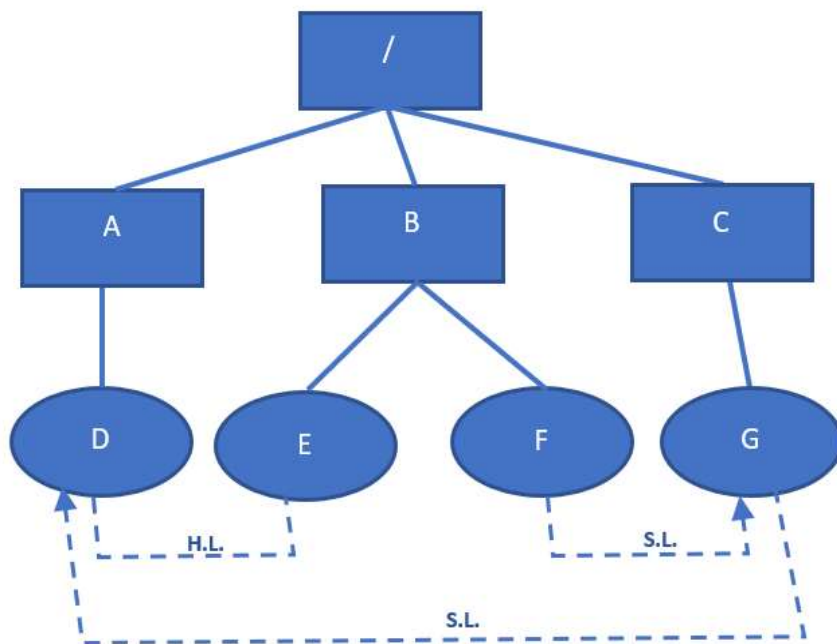
2) Teniendo en cuenta la estructura anterior, detalle los accesos necesarios a los ficheros indicados en los siguientes puntos:

- a) Acceso al registro lógico 19 del fichero B
- b) Acceso al registro lógico 26 del fichero A
- c) Acceso al último registro lógico del fichero C

6. REBOBINADO MANUAL

En un disco de 90 Mbytes con sistema de ficheros tipo Unix, tenemos Boot con un tamaño de un bloque, un Superbloque de 8 bloques (en el que se encuentra el mapa de bits de i-nodes pero no el mapa de bits de bloques). Todos los apuntadores son de 32 bits y el último bloque es el bloque 46079. Los registros lógicos son de 512 bytes. Los i-nodes tienen la siguiente estructura: una parte de Info de 2000 bytes, 10 apuntadores directos, uno indirecto simple y otro indirecto doble (no hay indirecto triple). El primer i-node es el i-node nº 1 y pertenece al directorio raíz. Se conoce que el primer bloque en el que se almacenan datos es el 72.

La estructura del árbol de directorios del disco es la siguiente:



Donde D es un hard-link a E y F es un soft-link a G. G es un soft-link a D. El fichero E tiene 2088 registros lógicos en zonas consecutivas del disco.

1. Dibuje la estructura lógica del disco rellenando toda la información posible con los datos de que disponemos, sabiendo que los siguientes accesos se realizan de la forma:

Reg nº 1 de E

Bloque Nº 12 (I-node nº 1), Bloque Nº 72, Bloque Nº 14,

Bloque Nº 74, Bloque Nº 16 y Bloque Nº 79

Reg nº 2088 de F

Bloque Nº 12, Bloque Nº 72, Bloque Nº 14, Bloque Nº 74,

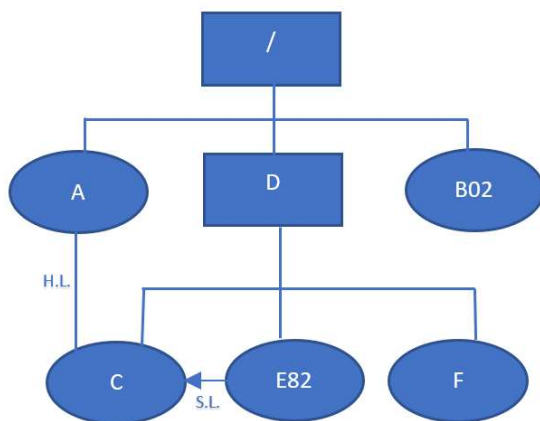
Bloque Nº 17, Bloque Nº 76, Bloque Nº 12, Bloque Nº 72,
Bloque Nº 15, Bloque Nº 75, Bloque Nº 18, Bloque Nº 77,
Bloque Nº 12, Bloque Nº 72, Bloque Nº 13, Bloque Nº 73,
Bloque Nº 16, Bloque Nº 78, Bloque Nº 600

2. Si convertimos este sistema de ficheros a FAT 16, ¿Cuántos bloques ocuparía cada una de las copias de la FAT?
3. En un sistema de ficheros FAT 16 ¿Cuántos accesos supondría leer por completo todos los registros del fichero D si tenemos espacio suficiente para que la FAT está completa en memoria?

7. YO ME QUEDO EN CASA

Tenemos un disco de 4 MBytes, con 16 sectores, 64 cilindros y 4 superficies. El tamaño del bloque es el doble que el del sector.

- El boot ocupa 1 bloque
- El Superbloque tiene un tamaño de 2 KBytes (contiene el mapa de bits de i-nodos y bloques del disco)
- Hay 64 i-nodos donde cada uno tiene 22 bytes para la Info, 3 apuntadores directos, un indirecto simple y un indirecto doble que funcionan mediante apuntadores de 16 bits



Información Adicional		
Fich. / Dir.	Tam. (KB)	1 ^{er} bloque
A	2056	100
B02	3	70
D	2	4
C	HARD LINK CON A	
E82	SOFT LINK HACIA C	
F	9.5	50

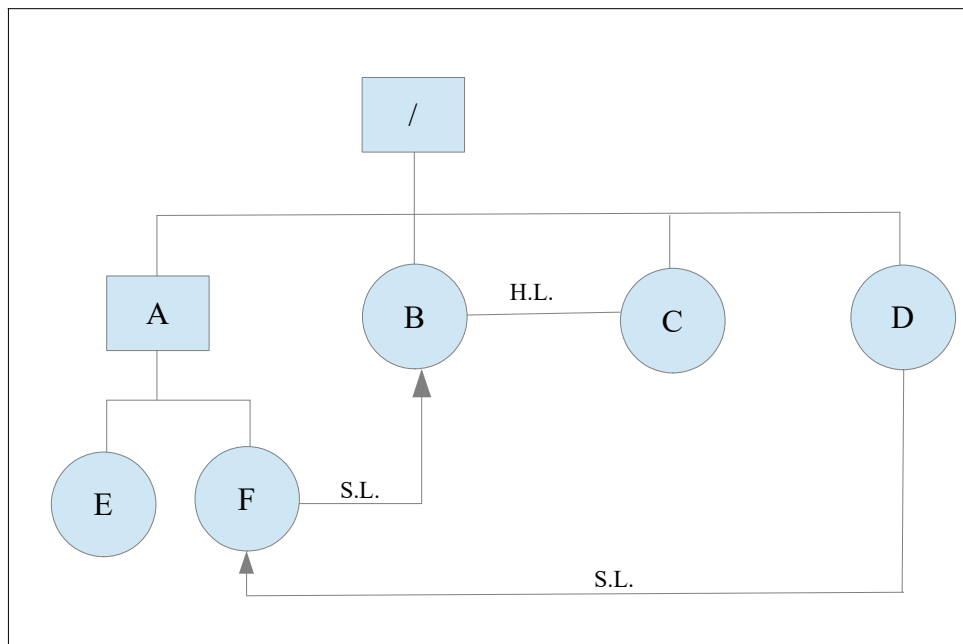
Teniendo en cuenta el árbol de directorios de arriba, la información adicional de los mismos y la estructura de disco que determina la información de arriba, realizar los siguientes apartados:

1. Dibuje cómo quedaría el disco y la tabla de inodos, si sabemos que:
 - El directorio raíz (/) se encuentra en el bloque que está a continuación del que contiene la tabla de inodos.
 - Se sabe que todos los bloques de datos de cada fichero se encuentran de manera contigua al primer bloque indicado en la tabla de Información.
 - En caso necesario, para el uso de bloques de apuntadores o rutas se comenzará a utilizar desde el último bloque del disco (a partir de ahí se iría hacia bloques inferiores).
 - El recorrido del árbol se realizará en profundidad
2. Detalle los accesos a los ficheros indicados en los siguientes puntos:
 - a) Acceso al registro lógico 8220 del fichero E82 (el tamaño de un registro lógico es de 256 Bytes)
 - b) Recorrer completamente el fichero F.

8. SIGUIENDO EL RASTRO

En un disco de 128 Mbytes con sistema de ficheros tipo Unix, el Boot ocupa 2 bloques, el Superbloque tiene un tamaño de 16 Kbytes (dentro se encuentra el mapa de bits de i-nodes y el mapa de bits de bloques). Los apuntadores son de 16 bits y hay 65536 bloques en el disco. Los registros lógicos son de 512 bytes. Los i-nodes tienen una parte de Info de 1000 bytes, 10 apuntadores directos, uno indirecto simple, otro indirecto doble y NO TIENEN indirecto triple. El primer i-node es el i-node nº 1 y pertenece al raíz. El primer bloque de datos es el 18.

La estructura del árbol de directorios del disco es la siguiente:



Donde D es un soft-link a F y F es un soft-link a B. Entre B y C hay un hard-link. El fichero C tiene 4140 registros lógicos en bloques consecutivos del disco. El fichero E tiene 12 registros lógicos en bloques consecutivos del disco.

1. Dibuje la estructura lógica del disco rellenando toda la información posible con los datos de que disponemos, sabiendo que los siguientes accesos se realizan de la forma:

Reg nº 12 de E

Bloque Nº 10 (I-node nº 1), Bloque Nº 18, Bloque Nº 10,

Bloque Nº 19, Bloque Nº 12 y Bloque Nº 22

Reg nº 4140 de D

Bloque Nº 10, Bloque Nº 18, Bloque Nº 11, Bloque Nº 23,
Bloque Nº 10, Bloque Nº 18, Bloque Nº 10, Bloque Nº 19,
Bloque Nº 12, Bloque Nº 24, Bloque Nº 10, Bloque Nº 18,
Bloque Nº 11, Bloque Nº 1061, Bloque Nº 1062 y Bloque Nº 1059

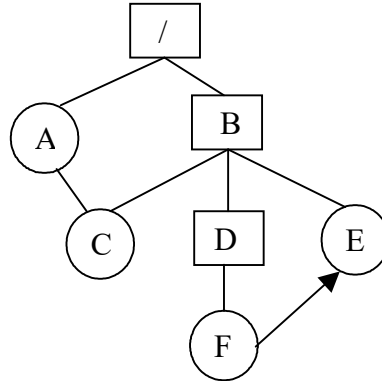
Reg nº 1 de B

Bloque Nº 10, Bloque Nº 18, Bloque Nº 11 y Bloque Nº 25

2. Si convertimos este sistema de ficheros a FAT 16, ¿Cuántos bloques ocuparía cada una de las copias de la FAT?
3. En un sistema de ficheros FAT 16 ¿Cuántos accesos supondría leer completamente todos los registros del fichero C si sabemos que la FAT está en memoria?

9. 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:

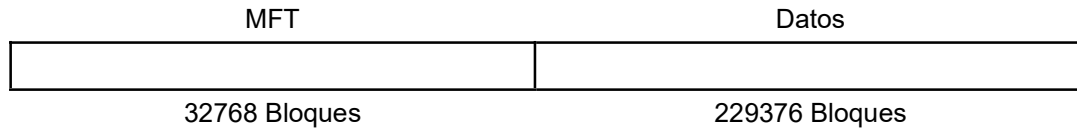


Los Registros Lógicos son de 1 Kbyte. Entre A y C hay un *hard-link*, y F es un *soft-link* a E.

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 el 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).

10. NO TENGAS FOBIAS

En un disco de 1 Gbyte cuya estructura es de 4 caras, 1024 pistas y 128 sectores tiene almacenado un sistema de ficheros NTFS como el que muestra la figura:



Una entrada de la MFT ocupa 8 bloques y tiene la siguiente estructura:

Cabec.	Inf. Estandar	Nom. Fich.	Descrip. Seg.	Datos
32bytes	6 Kbyte	512 bytes	9696 bytes	

Las extensiones tienen un tamaño de 4 bytes y están definidas de la siguiente forma:

1º Bloq. Nº
 Bloq.

--	--

Responda, justificando la respuesta:

1. ¿Cuál es el tamaño del bloque en este sistema de ficheros?
2. ¿Qué espacio nos queda en una entrada de la MFT para Datos?
3. ¿Qué tamaño máximo puede tener una extensión?
4. ¿Cuántas extensiones se pueden almacenar en una entrada de la MFT?
5. Si una entrada de directorio de la MFT tiene la misma estructura que una entrada de ficheros y el espacio dedicado a los Índices es el mismo que el dedicado a Datos. ¿Cuántos ficheros podría tener como máximo un directorio si sólo ocupa una entrada de la MFT? Debemos suponer que no estamos limitados por el número de entradas de la MFT y que el tamaño de un índice es el mínimo para poder apuntar a esta MFT.

11. 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.

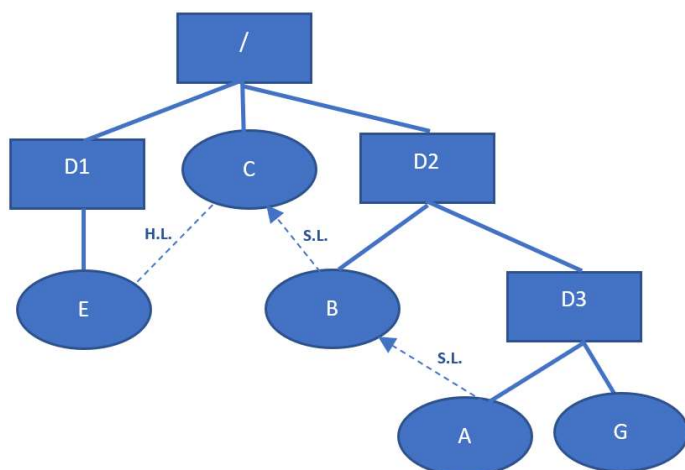
Se pide:

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

12. RASTREANDO NUEVA VARIANTE

Un disco de 64 Mbytes, con 8 caras, 256 cilindros y 32 sectores por pista, tiene montado un sistema de ficheros Unix. El tamaño del bloque es el doble que el del sector, los registros lógicos son de 1 Kbyte y los apuntadores son de 32 bits.

Si tenemos el siguiente árbol de ficheros y directorios:



Información Adicional		
Fich. / Dir.	Tam. (KB)	1 ^{er} bloque
/	2	13
D1	2	14
E	1300	60
D2	2	15
C	HARD LINK CON E	
B	SOFT LINK HACIA C	
D3	2	16
A	SOFT LINK HACIA B	
G	22	23

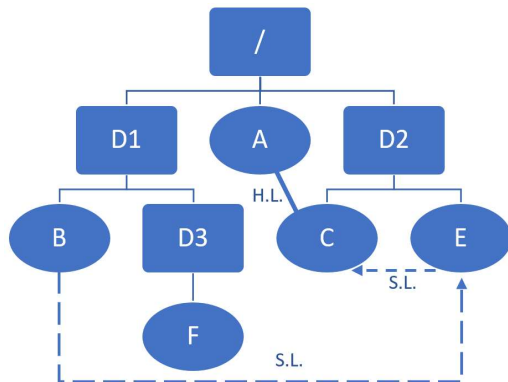
Teniendo en cuenta el árbol de directorios, la información adicional de los mismos y la estructura de disco que determina la información de arriba, realizar los siguientes apartados:

- 1) Dibuje la estructura lógica del disco, rellenando toda la información posible con los datos de que disponemos (incluido el mapa de bits de bloques y el mapa de bits de inodes), sabiendo que no hay boot, el superbloque ocupa 5 bloques (en el se encuentra el mapa de bits de bloques y el mapa de bits de inodes), disponemos de 16 inodes y cada uno ocupa medio bloque. Los inodes tienen la estructura normal del mismo.
- 2) Teniendo en cuenta la estructura anterior, detalle los accesos necesarios a los ficheros indicados en los siguientes puntos:
 - a) Acceso al último registro lógico del fichero B
 - b) Recorrer completamente el fichero G
- 3) Por último, si esto se convirtiera a FAT32, y todos los bloques del disco que fuesen necesario coincidieran tal y como están en el apartado anterior, represente cómo quedaría la estructura FAT así como el bloque que correspondería al directorio D3.

13. ENLACE FATÍDICO

Disponemos de un pequeño disco de 4 MBytes, con 4 superficies, 32 pistas y 16 sectores por pista, tiene montado un sistema de ficheros FAT32. El tamaño del bloque es el doble que el del sector, y el de los registros lógicos es de 1 KByte.

Si tenemos el siguiente árbol de ficheros y directorios:



Información Adicional		
Fich. / Dir.	Tam. (KB)	1 ^{er} bloque
D1	4	5
D2	4	17
D3	4	19
A	43	6
F	15	21

Teniendo en cuenta el árbol de directorios, la información adicional de los mismos (todos los bloques de datos de los ficheros se encuentran en bloques contiguos) y la estructura de disco que determina la información de arriba, realizar los siguientes apartados:

1) Recorriendo el árbol **en anchura**, dibuje la estructura lógica del disco rellenando toda la información posible con los datos de que disponemos, sabiendo que el Boot ocupa los dos primeros bloques. Represente igualmente el contenido de la FAT que refleje el estado del disco.

2) Teniendo en cuenta la estructura anterior y que la FAT no se encuentra cargada en memoria, detalle los accesos necesarios en los siguientes recorridos, indicando el bloque accedido en cada uno de los accesos:

- Recorrer completamente el fichero F
- Acceso al registro lógico 6 del fichero B

3) Si manteniendo la estructura y datos anteriores, nos encontráramos en un sistema UNIX, rellene adecuadamente un inodo X que controle adecuadamente al fichero A. Represente el inodo con la estructura por defecto, así como los bloques auxiliares que necesite.

14. NO TE FÍES

Un disco duro de 1 Gbyte con sistema de ficheros tipo NTFS tiene 524.288 sectores. El tamaño del bloque es el doble que el tamaño del sector.

1. ¿Qué tamaño de bloque tiene este sistema de ficheros?
2. ¿Cuántos bloques tiene el disco?

Si sabemos, además, que el tamaño máximo que puede tener una extensión es de 64 Mbytes y la estructura de una extensión es:

Nº Bloq.	1º Bloq.
<div></div>	<div></div>

3. ¿Qué tamaño en bytes tiene una extensión?

Si la MFT ocupa un 12.5% del espacio del disco y no hay Boot:

4. ¿Qué espacio del disco nos queda para los datos?

Si una entrada de directorio de la MFT ocupa 32 Kbytes y tiene 32 bytes para la cabecera, 6 Kbytes de Información Estandar, 512 bytes para los nombres de los directorios y 9696 bytes de Descriptor de Seguridad:

5. ¿Cuánto espacio está dedicado para los Índices?
6. ¿Cuántas entradas tiene la MFT?

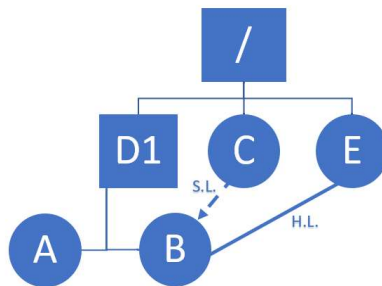
Si no estamos limitados por el número de entradas de la MFT y si el tamaño de un índice fuese el mínimo para poder apuntar a esta MFT:

7. ¿Cuántos ficheros podría tener como máximo un directorio si sólo ocupa una entrada de la MFT?

15. 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. **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

16. MES DE MUDANZA

Disponemos de un disco de 1 Mbyte, formateado con FAT16. El disco contiene la siguiente información:

B2 (Raíz)			B10			B11			B14			B16		
A	DIR	10	E	DAT	3	G	LINK	16	/B			/A/F		
B	DA T	7	F	LIN K	14									
C	DA T	7												
D	DIR	11												

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	511
FAT	X	X	eof	4	5	12	13	15	17	8	eof	eof	6	eof	eof	9	eof	eof			

1. Dibuja el árbol de directorios de este sistema de ficheros. Indica para cada fichero y directorio su tamaño y en qué bloques se encuentran.
2. ¿Cuántos bloques ocuparía la FAT en el disco?
3. Indica cuántos accesos y a qué bloques del disco se accederán para leer al registro lógico número 9 del fichero B si el tamaño del registro lógico es de 1 Kbyte y la FAT se encuentra cargada en memoria.
4. Cómo quedaría la estructura si la pasamos a un sistema de ficheros tipo Unix **haciendo coincidir siempre que sea posible** los bloques con la estructura de la FAT si sabemos que:
 - No hay boot.
 - El superbloque ocupa 1 bloque y contiene los mapas de bits.
 - Los apuntadores son de 16 bits
 - Tenemos 64 i-nodos. En cada i-nodo hay 3 apuntadores directos. Uno indirecto simple y otro indirecto doble. El tamaño de la información (excluidos los apuntadores) del i-nodo es de 22 bytes.
5. Con la estructura Unix, ¿cuántos accesos supondría acceder al 8 registro del fichero G?. Se deben indicar los bloques a los que se acceden.