



Universidad de  
Huelva

**ETSI**  
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, Enero de 2021

# 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>

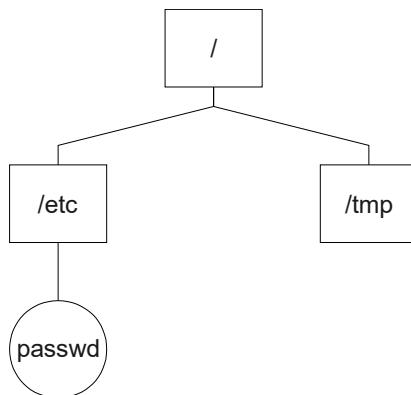
# TABLA DE CONTENIDOS

---

<b>1. Pelegrin returns .....</b>	<b>2</b>
1.1. Solución.....	5
<b>2. Tapitas III.....</b>	<b>8</b>
2.1. Solución.....	9
<b>3. Enemigo Público.....</b>	<b>11</b>
3.1. Solución.....	12
<b>4. Buen olfato.....</b>	<b>14</b>
<b>5. No me vengas con esa.....</b>	<b>16</b>
<b>6. Fat man.....</b>	<b>17</b>
<b>7. Siguiendo el rastro.....</b>	<b>18</b>
<b>8. El de siempre.....</b>	<b>20</b>
<b>9. Hay Inodos que matan.....</b>	<b>21</b>
<b>10. A falta de pan buenas son tortas.....</b>	<b>22</b>
<b>11. Inquisición.....</b>	<b>24</b>
<b>12. El tamaño sí que importa.....</b>	<b>26</b>
<b>13. No te fies.....</b>	<b>27</b>

# 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 Empanadas 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 llenar:

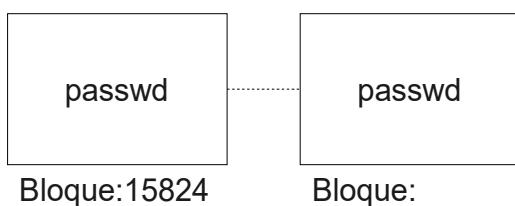
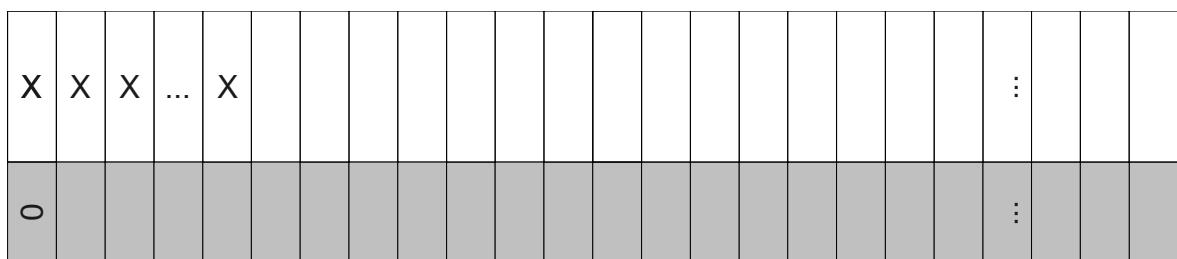
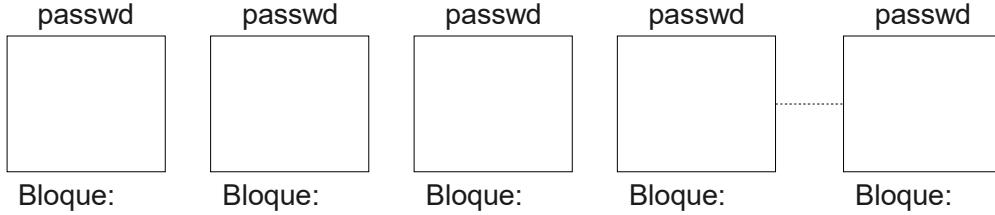
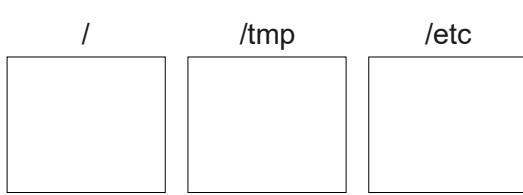


Tabla i-nodos

/	/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  $16384 \text{ bloques} * 16 \text{ bits / bloque} = 31.99 \text{ KB} = 32 \text{ bloques.}$
  3. El tamaño máximo de un fichero sería 16384 bloques - 64 bloques por FAT - 1 bloque por BOOT - 1 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

.	65
..	65
etc	15800
tmp	14000

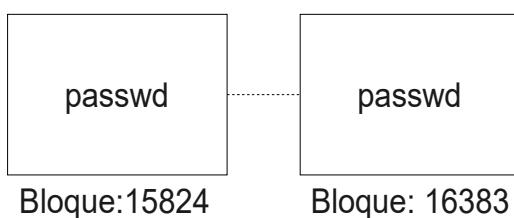
Bloque: 65

/tmp	
.	14000
..	65

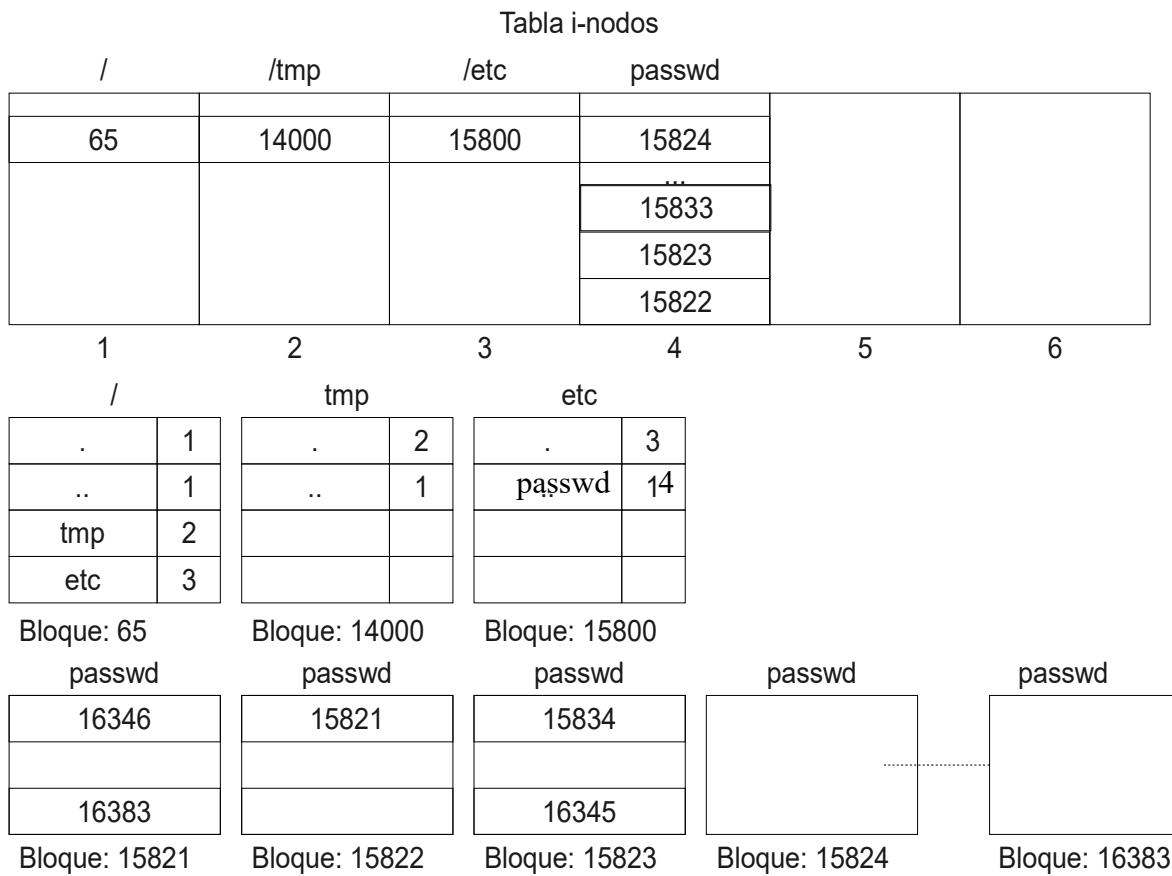
Bloque: 14000

/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 536000 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 (directorío /)
    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

Además de estos ejercicios, estos alumnos resuelven los ejercicios siguiente para así asentar un poco más sus conocimientos:

1. (UGR) En la siguiente figura se representa una tabla FAT.

Nombre				T	Fecha	Blq	FAT	
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. Represeñe 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. (UGR) 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. (UNED) 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 a 4 Kb.

## 2.1. Solución

- Antes de llenar la FAT, hay que tener en cuenta que el fichero DATOS1 ocupa 1200 / 512 bytes = 2,3 bloques = 3 bloques.
- Tras la creación del fichero DATOS1

Nombre	T	Fecha	Blq
Datos	F	8-2-90	3
DATOS1	F	1-3-90	4

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

- Tras el aumento de tamaño de Datos

Nombre	T	Fecha	Blq
Datos	F	8-2-90	3
DATOS1	F	1-3-90	4
D	D	3-3-90	9

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

- Tras la creación del directorio D

Nombre	T	Fecha	Blq
Datos	F	8-2-90	3
DATOS1	F	1-3-90	4
D	D	3-3-90	9

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

- Tras renombrar D

Nombre	T	Fecha	Blq
Datos	F	8-2-90	3
DATOS1	F	1-3-90	4
E	D	3-3-90	9

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

2. 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.
3. Actualmente, el tamaño máximo que ocupa un fichero es:  
 $10 + 256 + 256 * 256 = 65802$  Bloques \* 1 Kbyte = 65802 Kbytes = 64,25 Mbytes

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

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

$10 + 256 + 256 * 256 + 256 * 256 * 256 = 16843018$  Bloques \* 1Kbyte = 16843018 Kbytes =  
16448 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 él 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	.
1	..
2	A
3	B
14	C
2	.
1	..
4	D
5	E
6	F
3	.
1	..
8	G
9	H
8	.
3	..
4	I
10	J
5	K
15	L
9	.
3	..

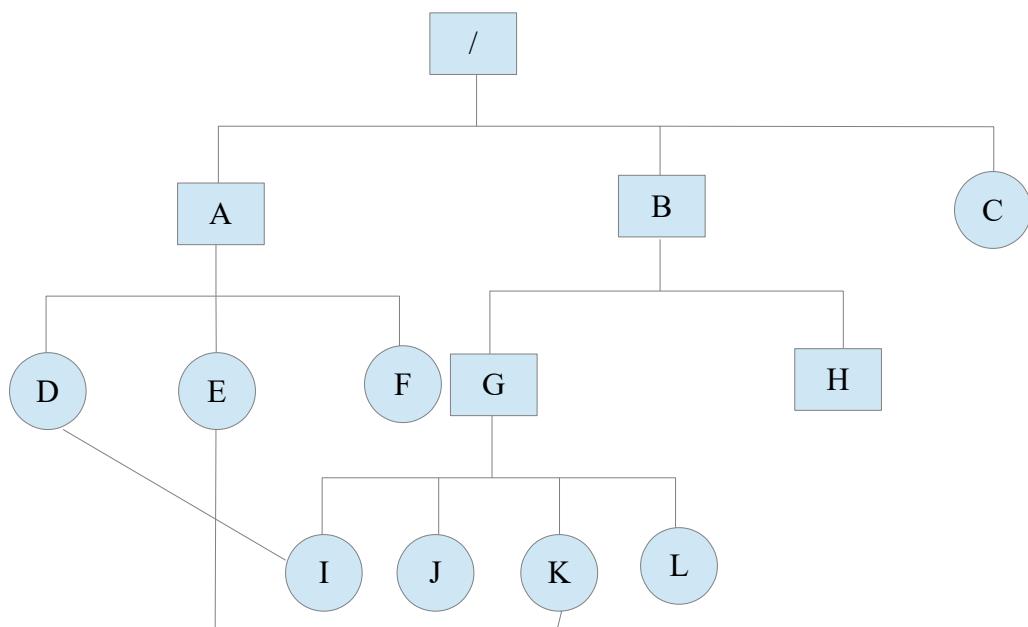
Todos los ficheros ocupan 1 Kbyte, excepto F que tiene 30 Kbytes.

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  $600 * 128 = 75$  Kbytes (75 bloques), el mapa de bits es de  $512 \times 1024$  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 bloques + 256 +  $256^2 = 65802$  bloques = 65802 Kbytes.
3. La estructura de directorios y ficheros a la que da lugar es:



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

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.

I		
.	D	4098
..	D	4098
A	D	4099
B	D	4100
C	-	4101

A		
.	D	4099
..	D	4098
D	-	4102
E	-	4103
F	-	4104

B		
.	D	4100
..	D	4098
G	D	4105
H	D	4106

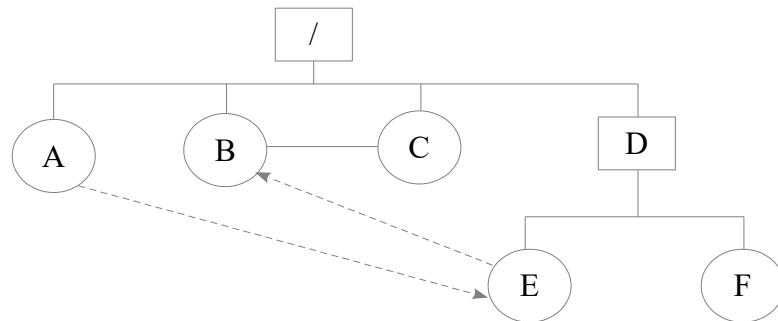
G		
.	D	4105
..	D	4100
I	-	4102
J	-	4107
K	-	4103
L	-	4108

H		
.	D	4106
..	D	4100

X		X	EOF F	EOF	EOF	EOF	EOF	EOF F	410 9	EOF F	EOF	EOF F	EOF	EOF F	EOF	EOF F	EOF	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. BUEN OLFATO

Un disco de 1 Mbyte, formateado con FAT16 contiene 2048 bloques. En ese disco se encuentra el siguiente árbol de ficheros y directorios:



Estos ficheros y directorios se encuentran situados en los siguientes bloques:

Raíz: B8

A: B9, soft link a E

B: B10, B11, B14, B18 y B21

C: hard link a B

D: B12

E: B13, soft link a B

F: B15, B16, B17, B19 y B20

1. Dibuja como se encuentra el disco y la FAT.

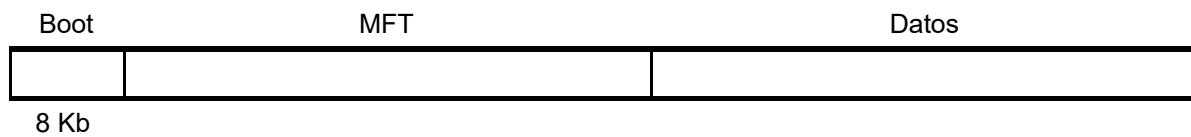
2. Indicar a qué bloques del disco habrá que acceder para leer el registro lógico 9 del fichero B si el tamaño de un registro lógico es de 256 bytes y la FAT se encuentra cargada en memoria.

3. Dibuja cómo quedaría el disco si lo pasamos a un sistema de ficheros tipo Unix donde no hay boot, el superbloque ocupa 4 bloques y contiene los mapas de bits, los apuntadores se mantienen con un tamaño de 2 bytes y tenemos 64 i-nodos con 4 apuntadores directos, uno indirecto simple y otro indirecto doble. El tamaño de la información (una vez excluidos los apuntadores) del i-nodo es de 52 bytes. Los bloques de los ficheros deben mantenerse igual que en la FAT.

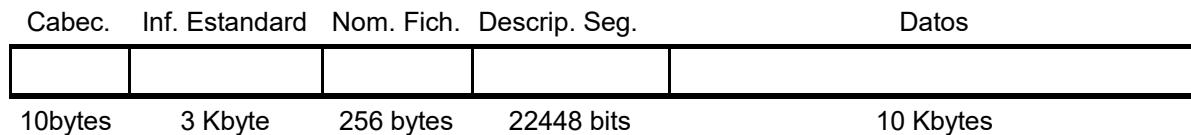
4. Con esta estructura Unix, indica a qué bloques del disco se accedería para leer el registro lógico 10 de A.

## 5. NO ME VENGAS CON ESA

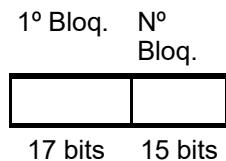
Un disco duro de 512 Mbytes con sistema de ficheros tipo NTFS está estructurado de la siguiente forma:



Una entrada de la MFT tiene la siguiente estructura:



Y las extensiones están definidas de la siguiente forma:



Si conocemos que la MFT tiene un total de 8192 entradas, responda a las siguientes preguntas:

1. ¿Cuál es el tamaño del bloque en este sistema de ficheros?
2. ¿Qué tamaño tiene la MFT? ¿Cuánto espacio nos queda en el disco para datos?
3. ¿Qué tamaño máximo puede tener una extensión?
4. ¿Cuántas extensiones se pueden almacenar en una entrada de la MFT?
5. Si un fichero sólo pudiera ocupar una entrada de la MFT, ¿cuál es el tamaño máximo de un fichero si no estamos limitados por el tamaño del disco?

## 6. FAT MAN

Disponemos de un disco con 512 Kbytes, formateado con FAT16. El disco contiene la siguiente información:

Raíz			B10			B16			B17		
A	DIR	10	F	LNK	17		/A/F				
B	DAT	7	E	DAT	14						
C	DAT	7									
D	LINK	16									

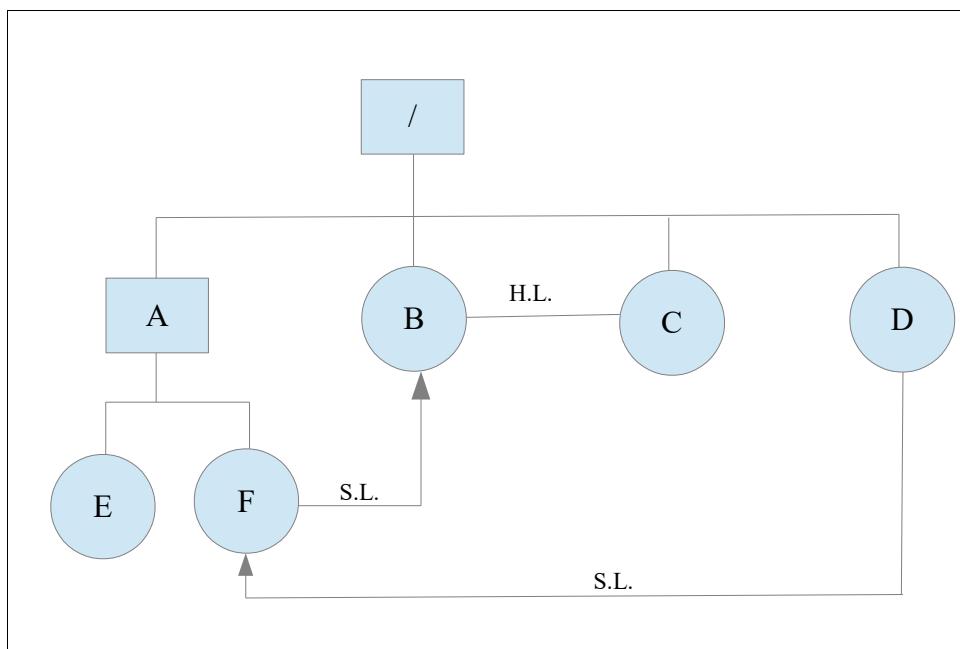
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	511	
FAT	X	X	eof	6	5	12	eof	11	9	15	eof	4	13	eof	8	3	eof	eof		

1. Indica la estructura del árbol de ficheros y directorios que hay en el disco. Para cada fichero y directorio se deberá indicar qué bloques ocupa y cuántos bloques ocupa.
2. Indica los accesos al disco se producirán para acceder al registro lógico número 12 del fichero B. El tamaño del registro lógico es de 512 bytes. La FAT se encuentra cargada en memoria. Se deben indicar los bloques a los que se accede.
3. Cómo quedaría la estructura si la pasamos a un sistema de ficheros tipo Unix si sabemos que:
  - No hay boot.
  - El superbloque ocupa 1 bloque y contiene los mapas de bits.
  - Los apuntadores son de 16 bits
  - Tenemos 32 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.
4. Con la estructura Unix, ¿cuántos accesos supondría acceder al registro lógico número 8 del fichero D? Se deben indicar los bloques a los que se accede.

## 7. 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.

- 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

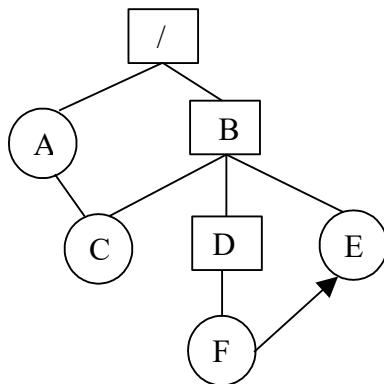
**Reg nº 1 de B**

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?

## 8. 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).

## 9. HAY INODOS QUE MATAN

Tenemos un disco de 32 Mbytes, con 32 sectores, 256 cilindros y 16 superficies. El tamaño del bloque es el doble que el del sector.

- El boot ocupa 1 bloque
  - El Superbloque tiene un tamaño de 10 Kbytes y contiene el mapa de bits de i-nodos y el mapa de bits del disco
  - Hay 8192 i-nodos y cada uno tiene 38 bytes para la Info, 10 apuntadores directos, un indirecto simple, un indirecto doble y un indirecto triple
  - Los apuntadores tienen el tamaño adecuado para poder direccionar este disco
1. Dibuje cómo quedaría el disco y la tabla de inodos si sabemos que del directorio raíz (el cual se encuentra justo después de la tabla de inodos) cuelga un directorio llamado A, que se encuentra en el bloque 2000. Además, en dicho directorio, hay un fichero llamado B, que ocupa los bloques 25000, 25001, 25002, 36000, 36001, 36002, 19000, 19001, 19002, 36003, y 25003 por ese orden.
  2. Añada a dicho disco un fichero llamado C, que cuelgue de A, y que ocupe 150 Kbytes consecutivos, a partir del bloque 60000. Dibuje sólo los bloques que sufren modificación.

Si quisieramos un sistema de ficheros MS-DOS para este mismo disco:

3. ¿Qué tipo de FAT necesito para manejar dicho disco?
4. ¿Cuánto ocupa dicha FAT?
5. ¿Cuántos accesos a disco se necesitan para leer totalmente el fichero B, si la FAT se encuentra en memoria?

## 10.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 almacenar su identificador.

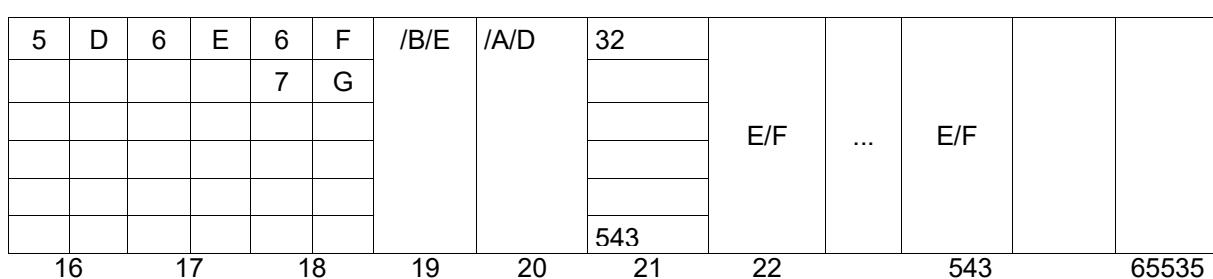
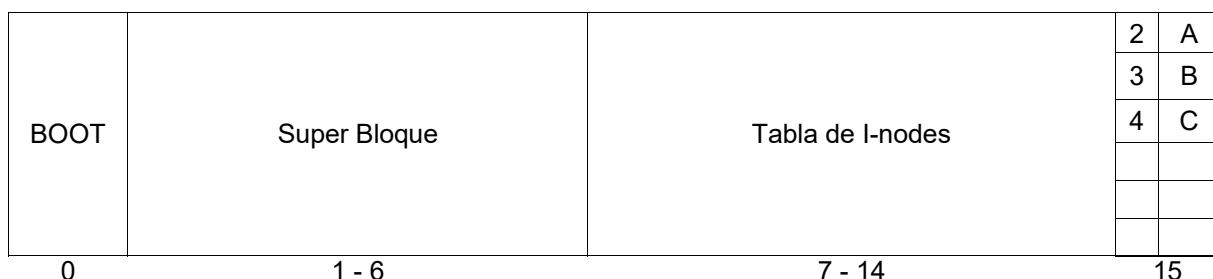
- Además de toda esta información, en cada i-nodo hay 32508 bits adicionales que se usan para guardar las fechas de modificación, ultimo acceso, usuarios propietarios etc.

Se pide:

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

## 11. INQUISICIÓN

Un sistema de ficheros tipo Unix tiene bloques de 2 Kbytes y la siguiente estructura:



Y se conoce la siguiente información sobre los i-nodos

## Tabla de I-nodes

/	A	B	C	D	E/F	G		
15	16	17	18	19	22	20		
					23			
					24			
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
					31			
					21			
1	2	3	4	5	6	7		16

1. Dibuje el mapa de bits de i-nodes y el mapa de bits de datos (se considera que están incluidos en el Super Bloque).
  2. ¿Qué tamaño tiene el disco?

3. ¿Qué tamaño tienen los i-nodes?
4. Dibuje la estructura de ficheros y directorios contenida en el disco, indicando en que bloques se sitúan cada fichero y cada directorio.
5. Si los registros lógicos son de 1 Kbyte, indique los accesos necesarios para acceder al último registro del fichero G .
6. ¿Qué tamaño en bytes tendría la FAT de dicho disco, si se desease formatear con la FAT más adecuada, manteniendo el tamaño del bloque?.

## 12. EL TAMAÑO SÍ QUE IMPORTA

Un sistema de ficheros basado en Unix utiliza **11 apuntadores directos**, uno indirecto simple, uno indirecto doble y otro indirecto triple en sus i-nodos. Por defecto, cada bloque de datos es de 1 Kbyte y los apuntadores a disco son de 16 bits. Cada i-nodo ocupa un bloque.

1. ¿Qué tamaño tendría el campo INFO del i-nodo?
2. Si atendemos sólo a la estructura del i-nodo, ¿cuál sería el tamaño máximo que podría tener un fichero en este sistema si no estamos limitados por el tamaño del disco?
3. Si quisiéramos almacenar un fichero en el directorio raíz con un tamaño de 10 MBytes. ¿Cuántos bloques en total se utilizarían en el disco para este fichero teniendo en cuenta la metainformación (i-nodo y bloques de apuntadores indirectos)?
4. Si quisiéramos leer el byte 1000000 del fichero anterior. ¿Cuántos accesos a bloques tenemos que realizar para completar la operación?
5. Si tuviésemos un disco de un 64 Mbytes y quisiéramos aplicar un sistema de ficheros FAT16. ¿Cuánto nos ocuparía la FAT si el tamaño de bloque utilizado fuese de 1 Kbyte? ¿Y si fuese de 512 bytes?

## 13. NO TE FIES

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.

--	--

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?