

1) Dada la siguiente FAT, estructura de directorio y estructura de bloques libres (administrada a traves de Bit Vector) se pide que la transforme en una estructura de I-Nodos, con la administración de bloques libres a través de bloques de direcciones libres contiguas. Cada I-Nodo contiene 10 punteros a datos mas un puntero al próximo bloque (si fuese necesario).

Tabla de asignación de archivos

Archivo	Inicio
Pato	8
Peto	5
Pito	99
Poto	2
Ultimo	88

Fat

	0	1	2	3	4	5	6	7	8	9
0	1	15	3	4	73	0	EOF		22	
1			13	14	30	80				
2			12	35						
3	89			EOF		62				
4					33		47	48	23	
5						44				
6			EOF				55			
7				74	75	92				
8	81	6							66	EOF
9			46					EOF	97	98

Bit Vector

1	1	1	1	1	1	1	0	1	0	0	0	1	1	1	1	0	0	0	0	0	0	1	1	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	1	1	0
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	1
50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
1	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1	1	1
75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99

Tabla de asignación de archivos

Archivo	I-nodo
Pito	I-nodo 2
Poto	I-nodo 3
Ultimo	I-nodo 4
Pato	I-nodo 0
Peto	I-nodo 1

I-nodo 0	I-nodo1	I-nodo 2	I-nodo 3	bloque 7	I-nodo 4
8	5	99	2	23	88
22	0	98	3	35	66
12	1	97	4	62	55
13	15	EOF	73	EOF	44
14	80		74		33
30	81		75		EOF
89	6		92		
EOF	EOF		46		
			47		
			48		
			7		

Lista de espacios libres

Bloque inicio	Cantidad
9	3
16	6
24	6
31	2
34	1
36	8
45	1
49	6
56	6
63	3
67	6
76	4
82	6
90	2
93	4

2) Dada la siguiente estructura de I-nodos. Se pide que la transforme a FAT y que represente la administración de espacios libres a través de bloques de direcciones libres contiguas.

Tabla de asignación de archivos

Archivo	I-nodo
Archiuno	I-nodo 2
Archidos	I-nodo 3
Architre	I-nodo 4
Archicua	I-nodo 0
Archicin	I-nodo 1

I-nodo 0	I-nodo1	I-nodo 2	I-nodo 3	bloque 7	I-nodo 4
8	5	99	2	48	88
22	0	98	3	23	66
12	1	97	4	35	55
13	15	EOF	73	62	44
14	80		74	EOF	33
30	81		75		EOF
89	6		92		
EOF	EOF		46		
			47		
			7		

Solución:

Tabla de asignación de archivos

Archivo	Inicio
Pato	8
Peto	5
Pito	99
Poto	2
Ultimo	88

Fat

	0	1	2	3	4	5	6	7	8	9
0	1	15	3	4	73	0	EOF		22	
1			13	14	30	80				
2			12	35						
3	89			EOF		62				
4					33		47	48	23	
5						44				
6			EOF				55			
7				74	75	92				
8	81	6							66	EOF
9			46					EOF	97	98

Lista de espacios libres

Bloque inicio	Cantidad
---------------	----------

7	1
9	3
16	6
24	6
31	2
34	1
36	8
45	1
49	6
56	6
63	3
67	6
76	4
82	6
90	2
93	4

3) Dada la siguiente estructura de I-nodos. Se pide que la transforme a Asignación Enlazada y que represente la administración de espacios libres a través de lista enlazada de espacios libres

Tabla de asignación de archivos

Archivo	I-nodo
Archiuno	I-nodo 2
Archidos	I-nodo 3
Architre	I-nodo 4
Archicua	I-nodo 0
Archicin	I-nodo 1

I-nodo 0	I-nodo1	I-nodo 2	I-nodo 3	bloque 7	I-nodo 4
8	5	99	2	48	88
22	0	98	3	23	66
12	1	97	4	35	55
13	15	EOF	73	62	44
14	80		74	EOF	33
30	81		75		EOF
89	6		92		
EOF	EOF		46		
			47		
			7		

Solución:

Tabla de asignación de archivos

Archivo	Bloque de comienzo	Bloque final
Archiuno	99	97
Archidos	2	62
Architre	88	33
Archicua	8	89
Archicin	5	6

Lista enlazada de espacios libres

FSLH
7

4) Dada la siguiente estructura FAT, se pide que complete las respectivas entradas en la tabla de directorio.
 Nota: los nombres de los archivos son indistintos (cualquier archivo puede relacionarse con cualquier nombre)

Fat

	0	1	2	3	4	5	6	7	8	9
0	1	15	3	4	73	0	EOF		22	
1			13	14	30	80				
2			12	35						
3	89			EOF		62				
4					33		47	48	23	
5						44				
6			EOF				55			
7				74	75	92				
8	81	6							66	EOF
9			46					EOF	97	98

Tabla de asignación de archivos

Archivo	Inicio
Archivo1	
Archivo2	
Archivo3	
Archivo4	
Archivo5	

Solución:

Tabla de asignación de archivos

Archivo	Inicio
Archivo1	8
Archivo2	5
Archivo3	99
Archivo4	2
Archivo5	88

5) Dada la siguiente asignación de bloques libres en formato Bit Vector, construya una tabla de bloques libres contiguos.

110011000011001011110010010

Solución:

Lista de espacios libres

Bloque inicio	Cantidad
2	2
6	4
12	2
15	1
20	2
23	2
26	1

6) Dada la siguiente estructura de I-nodos. Se pide que la transforme a Asignación Enlazada y que represente la administración de espacios libres a través de bloques de direcciones libres contiguas. Suponemos los bloques 0 a 99 en el área de datos.

Tabla de asignación de archivos

Archivo	I-nodo
Archiuno	I-nodo 2
Archidos	I-nodo 3
Architre	I-nodo 4
Archicua	I-nodo 0
Archicin	I-nodo 1

I-nodo 0	I-nodo 1	I-nodo 2	I-nodo 3	I-nodo 4
1	2	0	90	21
3	4	10	91	22
5	6	20	92	23
7	8	30	93	31
9	EOF	40	94	32
11		50	EOF	33
EOF		EOF		41
				42
				43
				EOF

Solución:

Tabla de asignación de archivos

Archivo	Bloque de comienzo	Bloque final
Archiuno	0	50
Archidos	90	94
Architre	21	43
Archicua	1	11
Archicin	2	8

Lista de espacios libres

Bloque inicio	Cantidad
12	8
24	6
34	6
44	6
51	39
95	5

7) Dados los siguientes I-nodos, estructura de bloques libres (mantenida en forma ordenada, y suponiendo que por bloque físico entran 20 direcciones de bloques libres, contando el puntero al próximo bloque), y Directorio, se pide:

- a. Analice el estado actual del File System (la columna de inicio deberá ser completada por Ud., luego de hacer el punto b).
- b. Si encuentra errores justifique la respuesta. Solo se representan el área de datos, las del sistema no están dentro de los datos presentados.

Tabla de asignación de archivos

Archivo	Inicio	I-nodo
Tato		0
Tito		1
Toto		2
Tuto		3
Tutu		4
Tete		0

I-NODOS

I-Nodo 0	10	I-Nodo 1	20	I-Nodo 2	21	I-Nodo 3	I-Nodo 4
5	25	7	98	40	50	11	13
6	33	8		41	51	3	15
2		90		42	52	9	16
14		91		43	53	4	17
0		92		44			18
80		93		45			71
81		94		46			
22		95		47			
23		99		48			
24		1		49			
10		20		21			

Lista de bloques libres:

Pos.	0	96	97
0	12	58	78
1	19	59	79
2	26	60	82
3	27	61	83
4	28	62	84
5	29	63	85
6	30	64	86
7	31	65	87
8	32	66	88
9	34	67	89
10	35	68	97
11	36	69	
12	37	70	
13	38	72	
14	39	73	
15	54	74	
16	55	75	
17	56	76	

18	57	77	
19	96	97	

Solución:

Tabla de asignación de archivos

Archivo	Inicio	I-nodo
Tato	5	0
Tito	7	1
Toto	40	2
Tuto	11	3
Tutu	13	4
Tete	5	0

El sistema se encuentra inconsistente.

El bloque 0 no puede utilizarse para almacenar direcciones de bloques libres porque está siendo utilizado por el I-Nodo 0.

El bloque 97 no debe figurar en la lista de bloques libres porque está utilizado para almacenar direcciones de bloques libres.

8) Dada la estructura anterior, se pide que la transforme en una estructura FAT, con la administración de bloques libres a través de Bit Vector. En caso de que el punto anterior tenga algún error se debe realizar el traspaso de la versión corregida.

Solución:

Tabla de asignación de archivos

Archivo	Inicio
Tato	5
Tito	7
Toto	40
Tuto	11
Tutu	13
Tete	5

FAT

--	0	1	2	3	4	5	6	7	8	9
0	80	98	14	9	EOF	6	2	8	90	4
1		3		15	0	16	17	18	71	
2			23	24	25	33				
3				EOF						
4	41	42	43	44	45	46	47	48	49	50
5	51	52	53	EOF						
6										
7		EOF								
8	81	22								
9	91	92	93	94	95	99			EOF	1

Bit Vector:

--	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	0
2	0	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
8	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1

9) Dada la siguiente FAT, estructura de Bit Vector, y Directorio, se pide que analice el estado actual del File System. Si encuentra errores justifique la respuesta. Solo se representa el área de datos, las del sistema no están dentro de los datos presentados.

Tabla de asignación de archivos

Archivo	Inicio
Tato	5
Tito	7
Toto	40
Tuto	11
Tutu	13
Tete	5

FAT

--	0	1	2	3	4	5	6	7	8	9
0	80	98	14	9	EOF	6	2	8	90	4
1		3		15	0	16	17	18	70	
2			23	24	25	33				
3				EOF						
4	41	42	43	44	45	46	47	48	49	50
5	51	52	53	EOF						
6										
7	EOF									
8	81	22								
9	91	92	93	94	95	99			EOF	1

Bit Vector:

--	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	0	1	1	1	1	1	0	1	0	1	1	1	1	1	1	0
2	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	1

10) Considere que se está utilizando un S.O. de la familia Unix, con una implementación de I-Nodos ISAM con direcciones de 4 bytes y bloques de 512 bytes. Cada I-Nodo se almacena en un bloque de disco. Se pide:

- a. Cual es el espacio total en disco requerido para almacenar un archivo de 2 MB
- b. Cuantos accesos a disco serán necesarios para leer el último bloque del archivo

Solución:

- a. Se necesitan 4130 bloques de disco de 512 bytes, o sea 2065 kb.

4096 bloques de datos

1 bloque para el I-Nodo

1 bloque para los punteros de la indirección simple

32 bloques para los punteros de la indirección doble

- b. Se necesitan 4 accesos

1 para leer el I-Nodo

1 para leer el primer bloque de la indirección doble

1 para leer el segundo bloque de la indirección doble

1 para leer el bloque de datos (el último del archivo)

11) Considere que se está utilizando un S.O. con un File System implementado utilizando asignación indexada con direcciones de 4 bytes y bloques de 512 bytes. En el área de catálogo del disco está la Tabla de asignación de archivos mostrada mas abajo. El archivo UNO mide 2048 kb, el archivo DOS mide 3120 kb y el archivo TRES mide 10 kb.

Tabla de asignación de archivos

Archivo	Bloque Índice
UNO	0
DOS	15
TRES	23

Se pide:

- Cual es el espacio total en disco requerido para almacenar estos archivos (no considere el área de catálogo)
- Cuantos accesos a disco serán necesarios para leer al último bloque del archivo UNO (suponiendo que la tabla de asignación de archivos ya está en memoria)

Solución:

a.

Archivo UNO:	4096 bloques de datos	
	33 bloques para índice	
Archivo DOS:	6240 bloques de datos	
	50 bloques para índice	
Archivo TRES:	20 bloques de datos	
	1 bloque para índice	

Total		10440 bloques = 5220 kb

b.

33 accesos para leer bolques índices mas 1 acceso para leer el bloque de datos = 34 accesos

12) Considere que se está utilizando un S.O. con un File System implementado utilizando asignación enlazada con direcciones de 4 bytes y bloques de 512 bytes. En el área de catálogo del disco está la Tabla de asignación de archivos mostrada mas abajo. El archivo UNO mide 2040 kb, el archivo DOS mide 10 kb y el archivo TRES mide 384 kb.

Tabla de asignación de archivos

Archivo	Bloque Inicio
UNO	22
DOS	13
TRES	146

Se pide:

- Cual es el espacio total en disco requerido para almacenar estos archivos (no considere el área de catálogo)
- Cuantos accesos a disco serán necesarios para leer al último bloque del archivo TRES (suponiendo que la tabla de asignación de archivos ya está en memoria)

Solución:

a.

$$\begin{array}{rcl}
 \text{Archivo UNO } 2040 \text{ kb} / 508 \text{ by} & = & 4113 \text{ bloques} \\
 \text{Archivo DOS } 10 \text{ kb} / 508 \text{ by} & = & 21 \text{ bloques} \\
 \text{Archivo TRES } 384 \text{ kb} / 508 \text{ by} & = & 775 \text{ bloques}
 \end{array}$$

$$\begin{array}{rcl}
 \text{Total} & \text{-----} & \\
 & 4909 \text{ bloques} & = 2454,5 \text{ kb}
 \end{array}$$

b.

Se necesitan 775 accesos

13) Considere que se está utilizando un S.O. con un File System que implementa asignación contigua con bloques de 512 bytes. En el área de catálogo del disco está la tabla de asignación de archivos mostrada más abajo. El archivo PATO mide 23.456 bytes, el archivo PETO mide 100.000 bytes y el archivo PITO mide 320 bytes.

Tabla de asignación de archivos

Archivo	Bloque Inicio	Longitud
PATO	22	
PETO	130	
PITO	146	

Se pide:

- Completar la columna "Longitud" de la tabla de asignación de archivos
- Indicar si hay alguna inconsistencia en el file system
- Cuantos accesos a disco serán necesarios para leer el anteúltimo bloque del archivo PETO (suponiendo que la tabla de asignación de archivos ya está en memoria)

Solución:

a.

Archivo	Bloque Inicio	Longitud
PATO	22	46
PETO	130	196
PITO	146	1

- Si, si el archivo PETO comienza en el bloque 130 y ocupa 196 bloques el archivo PITO no puede comenzar en el bloque 146
- Solo el acceso al bloque de datos.

14) Suponga un disco con bloques de 512 bytes. Un archivo de 75231 bytes, otro de 7 bytes y otro de 5630 bytes. Un file system con estructura de I-nodos ISAM. Cada I-nodo ocupa 80 bytes y en un bloque de disco se almacenan tantos I-nodos como quepan (enteros). Cada puntero ocupa 4 bytes.

Determinar:

- a. Tamaño total (en bytes) necesario para almacenar los 3 archivos
- b. Cantidad de accesos a disco necesarios para leer los 3 archivos completos
- c. Cantidad de accesos a disco necesarios para leer el último bloque de cada archivo

Solución:

Archivo 1 -> 147 bloques

Archivo 2 -> 1 bloque

Archivo 3 -> 11 bloques

a.

I-Nodos	1	Bloque	
Archivo 1	150	Bloques	
Archivo 2	1	Bloque	
Archivo 3	12	Bloques	

Total	164	Bloques = 83.968 bytes	

b. 164 Accesos

c.

1	Acceso para I-Nodos
3	Accesos para archivo 1
1	Acceso ara archivo 2
2	Accesos para archivo 3

7	Accesos total

15) Suponga un disco con bloques de 512 bytes. Un file system con estructura de I-nodos I-Sam. Los I-Nodos se almacenan en el área de catálogo. Cada puntero ocupa 4 bytes.

Considerando que un bloque es igual a un sector, la asignación es contigua, y que :

- Los archivos del sistema, tablas del sistema, área de catálogo, etc. se encuentran desde el comienzo del disco (bloque 0), y ocupan los primeros 3.747.328 bytes.
- Luego hay un espacio libre de 20.992 bytes.
- A continuación de esto hay un archivo de texto de 7.213 caracteres.
- A continuación existen 19.968 bytes sin usar
- Luego existe un archivo de base de datos de 25.000 registros, con los siguientes campos: código (5 bytes), descripción (9 bytes), e importe (6 bytes). Un registro no puede sub-dividirse en dos sectores. Este archivo posee, además, un bloque de encabezamiento.
- Luego de éste archivo existen 3.304 sectores libres.
- Y por último, un archivo binario cuyo I-nodo contiene 285 punteros en total (tanto a bloques de datos como direcciones)

Se pide :

- Tamaño total del disco (bloques / tamaño).
- Tamaño total ocupado por todos los archivos.
- Construir la tabla de bloques libres (por lista de bloques libres contiguos)..
- Cantidad de accesos a disco necesarios para acceder en forma secuencial al registro 12343 de la base de datos

Solución:

a.

Archivos del sistema:	7319	bloques
Libre	41	
Archivo de texto	16	
Libre	39	
Base de datos	1010	
Libre	3304	
Archivo binario	285	

Tamaño total del disco	12014 bloques	-> 6.007 Kb

b.

Tamaño archivos 1311 bloques -> 655,5 Kb

c. Lista de bloques libres contiguos

Bloque	Cantidad
7319	41
7376	39

8425	3304
------	------

d.

Accesos a bloque de encabezado	1
Acceso a bloques con registros	494
Acceso a indirecciones	5

Cantidad total de accesos	500

16) Suponga un sistema con bloques de 512 bytes y punteros de 2 bytes. Que utiliza asignación indexada (No son I-Nodos ISAM). Se pide:

- a. Cuanto espacio en disco ocupará un archivo de 4096 KB
- b. Cuantos accesos a disco serán necesarios para leer el bloque 766 del archivo
- c. Cuál será el tamaño máximo posible para un archivo?
- d. Cuanto ocuparía dicho archivo con asignación enlazada

a.

4096 kb de datos \Rightarrow 8192 bloques

Bloq, índices 33 bloques

8225 bloques \Rightarrow 4112,5 kb

b.

Se necesitan 5 accesos (4 para el índice + 1 para el dato)

c.

16 bits \rightarrow apuntan 65536 bloques de 512 bytes \rightarrow 32 MB

d.

$8192 * 512 / 510 = 8225$ bloques \Rightarrow 4112,5 Kb

17) Suponga un sistema con bloques de 1024 bytes y punteros de 4 bytes. Que utiliza I-Nodos ISAM. Se pide:

- a. Cuanto espacio en disco ocupará un archivo de 8192 KB?
- b. Cuantos accesos a disco serán necesarios para leer en forma secuencial todo el archivo?
- c. Cuál será el tamaño máximo posible para un archivo?
- d. Cuanto ocuparía dicho archivo con asignación doblemente enlazada?

Solución:

- a. 8225 bloques \approx 8192 bloques de datos + 33 bloques de direcciones
- b. 8225 accesos
- c. 16,06 GB
- d. $8192 \times 1024 / 1016 = 8256,5 \approx 8257$ bloques

18) Suponga un sistema que utiliza un file system con bloques de 1 kb y punteros de 4 bytes. Contiene un archivo de 3 Mb. Se pide:

a. ¿Cuánto espacio en disco ocuparía si se utilizara, asignación indexada, (NO I-Nodos ISAM) y cuánto si se utilizara asignación doblemente enlazada?

b. Cuántos accesos a disco serán necesarios para leer el último bloque del archivo en cada file system ? Si supone alguna estructura en memoria aclárelo.

Solución:

a. Asignación indexada : 3072 bloques de datos
 13 bloques índice

 3085

Asignación doblemente enlazada: $3072 \times 1024 / 1016 = 3097$

b. Indexada : 14 accesos

Doblemente enlazada: 1 acceso

19) Suponga un sistema que utiliza un file system con bloques de 1 kb y punteros de 4 bytes. Contiene un archivo de 4 Mb. Se pide:

a. ¿Cuánto espacio en disco ocuparía si se utilizaran I-Nodos ISAM? ¿Cuánto, si se utilizara asignación enlazada?

b. ¿Cuántos accesos a disco serán necesarios para leer el último bloque del archivo en cada file system ? Si supone alguna estructura en memoria aclárelo.

Solución:

a. I – Nodos

	4096	bloques de datos
1	Ind. Simple	
	16	Ind. Doble

4113 bloques en total

Enlazada

$4096 * 1024 / 1020 = 4112.06 \Rightarrow 4113$ bloques

b. I-Nodos: 3 accesos (suponiendo el I-Nodo en memoria)

Enlazada: 4113 accesos

20) Considere que se está utilizando un S.O. de la familia Unix, con una implementación de I-Nodos ISAM con direcciones de 4 bytes y bloques de 1024 bytes. Cada I-Nodo se almacena en un bloque de disco. Se pide:

a. Cuál es el espacio total en disco requerido para almacenar un archivo de 4 MB?

b. Cuantos accesos a disco serán necesarios para leer el último bloque del archivo?

a. Se necesitan 4114 bloques de disco de 1024 bytes, o sea 4113 kb.

4096 bloques de datos

1 bloque para el I-Nodo

1 bloque para los punteros de la indirección simple

16 bloques para los punteros de la indirección doble

b. Se necesitan 4 accesos

1 para leer el I-Nodo

1 para leer el primer bloque de la indirección doble

1 para leer el segundo bloque de la indirección doble

1 para leer el bloque de datos (el último del archivo)

21) Un sistema trabaja con bloques de 1 kb y punteros de 4 bytes. Contiene un archivo de 4 Mb. Se pide:

a. ¿Cuánto espacio en disco ocupará dicho archivo con asignación indexada, (NO I-Nodos ISAM) y cuánto si se utilizara asignación doblemente enlazada?

b. Cuántos accesos a disco serán necesarios para leer el último bloque del archivo en cada file system ? Si supone alguna estructura en memoria aclárelo.

a. Asignación indexada: 4096 bloques de datos + 17 bloques índices = 4113 bloques total

Asignación doblemente enlazada: $4096 * 1024 / 1016 = 4129$ bloques

b. Asignación indexada: 17 accesos a índice + 1 acceso a dato = 18 accesos en total

Asignación doblemente enlazada: 1 acceso

22) Se dispone de un disco formateado con un file system del tipo I-Nodos, con bloques de 1KB, con punteros de 32 bits. En este FS existe un proceso que se encarga de acumular la información procesada del sistema de integraciones. Este proceso corre todos los días a las 22hs y realiza lo siguiente: lee cada uno de los archivos en el directorio temporal y los concatena al archivo final.

En el directorio temporal se encuentran los siguientes archivos:

```
ls -l /tmp/files
```

```
-rw-r--r--      9358   Aug  4 19:49   integradorA.txt
-rw-r--r--     15129   Aug  4 14:51   integradorB.txt
-rw-r--r--     20909   Aug  4 19:28   IntegradorC.txt
```

```
ls -l /tmp/final
```

```
-rw-r--r--    123456  Aug  3 22:01   integradores.txt
```

1. Indique el tamaño del archivo integradores.txt y la cantidad de bloques que ocupa (demostrando cómo calculó estos datos), luego de que se ejecute el proceso del día 4 de Agosto.
2. Dibuje cómo es la estructura del i-nodo que contiene al archivo integradores.txt
3. ¿Existe fragmentación interna en el archivo? Explique por qué y en cuántos bloques afecta.
4. ¿Cuál es el tamaño máximo que puede tener un archivo en este FS?

23) En cierto SO, se dispone de la siguiente tabla de directorios (TD):

Nombre	i-nodo	Bloques de dato	Tamaño
/tmp/2023-02-11.log	1	9	
/tmp/2023-02-12.log	2	1035	
/tmp/2023-02-13.log	3	5354	
/tmp/2023-02-14.log	4	2050	

Los i-nodo soportan direcciones de 16 bits y el tamaño de bloque es de 1KB.
En un determinado momento un usuario ejecuta el siguiente comando:

```
$ head -c 20000 /tmp/2020-02-11.log > /tmp/sample.txt
```

- ¿Cómo queda la TD luego de la ejecución del comando anterior?
- Dibuje la estructura del i-nodo 3.
- ¿Cuántos bytes ocupan cada uno de estos archivos, incluyendo el archivo creado en el punto A?, indique cuántos bytes corresponden a datos y cuántos a la administración del archivo dentro del FS. Utilice la unidad adecuada (Bytes, KB, MB, GB). No considere fragmentación interna para sus respuestas

Nota:

- head -c X toma los primeros X bytes del archivo y
- > /tmp/sample.txt crea un nuevo archivo

24) En cierto sistema operativo que trabaja con ext3 y con i-nodos ISAM. Los i-nodo soportan direcciones de 32 bits y el tamaño de bloque es de 2KB. Una base de datos guarda la información de sus tablas en i-nodos (un i-nodo por tabla). La tabla Transaccion tiene la siguiente estructura:

- id_transaccion: 2 bytes
- fecha: 4 bytes
- id_producto: 2 bytes
- id_sucursal: 2 bytes
- id_payer: 2 bytes
- importe: 4 bytes
- id_metodo_pago: 2 bytes

Cada registro de la tabla se guarda a continuación del anterior y todo el registro debe estar completo (no se puede tener parte de un registro en un bloque y el resto en otro bloque).

Antes de la ejecución del proceso de proceso que realiza el merge de todas las sucursales la tabla contaba con 56.789 registros. La ejecución del proceso indicó que se agregaron 34.567 registros.

Indique:

- a. ¿Cómo era la estructura del i-nodo antes del update de la tabla?
- b. ¿Cómo es la estructura del i-nodo después del update de la tabla?
- c. ¿Cuántos ocupa el i-nodo que representa la tabla Transaccion?, indique cuántos bytes corresponden a datos y cuántos a la administración del archivo dentro del FS. Utilice la unidad más adecuada (Bytes, KB, MB, GB)
- d. ¿Cuál es la cantidad máxima de registros que puede tener la tabla?