



Universidad Tecnológica Centroamericana
UNITEC

Laboratorio FAT

Presentado por:

José Lobo 12241127

Presentado a:

Roman Arturo Pineda Soto

Clase y Sección:

1401 SISTEMAS OPERATIVOS II 2025Q2

Fecha de Entrega:

Tegucigalpa M.D.C. 17 de junio del 2025

Índice

Código Fuente.....	3
Imágenes	3
Enlace de Github	8
Ejecución del Programa	9
La creación de dos archivos	9
La eliminación de uno.....	10
Preguntas de Reflexión	11

Código Fuente

Imagenes

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  // Número total de bloques del "disco"
7  const int DISK_SIZE = 128;
8
9  // Estructura para representar una entrada en la FAT
10 struct FATEntry
11 {
12     int nextBlock; // Índice del siguiente bloque (-1 si es el último)
13     bool isFree;   // true si el bloque está libre
14 };
15
16 // Tabla FAT simulada
17 vector<FATEntry> fatTable(DISK_SIZE);
18
19 // Inicializar la FAT (todos los bloques libres)
20 void initializeFAT()
21 {
22     for (int i = 0; i < DISK_SIZE; ++i)
23     {
24         fatTable[i].isFree = true;
25         fatTable[i].nextBlock = -1;
26     }
27 }
```



```
1 // Eliminar un archivo dado su bloque inicial
2 void deleteFile(int startBlock)
3 {
4     int idBlock = startBlock;
5     while (idBlock != -1)
6     {
7         int siguienteBloque = fatTable[idBlock].nextBlock;
8         fatTable[idBlock].nextBlock = -1;
9         fatTable[idBlock].isFree = true;
10        idBlock = siguienteBloque;
11    }
12 }
```

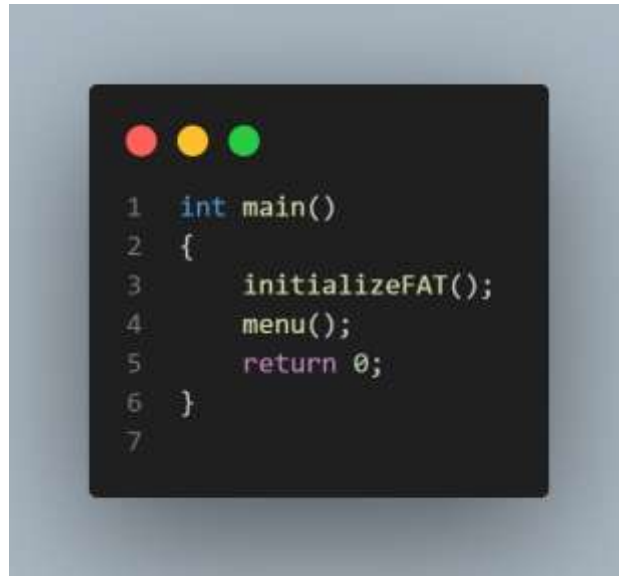
```
1 // Crear un archivo de tamaño dado (en bloques)
2 // Retorna el índice del bloque inicial o -1 si no hay suficiente espacio
3 int createFile(int fileSize)
4 {
5     int filellenado = 0;
6     int idAnterior = -1, idInicial = -1;
7     for (size_t i = 0; i < DISK_SIZE; i++)
8     {
9         if (!fatTable[i].isFree)
10         {
11             continue;
12         }
13         if (idAnterior == -1)
14         {
15             idAnterior = i;
16             idInicial = i;
17             fatTable[i].isFree = false;
18             filellenado++;
19             if (filellenado >= fileSize)
20             {
21                 fatTable[i].nextBlock = -1;
22                 break;
23             }
24             continue;
25         }
26         fatTable[idAnterior].nextBlock = i;
27         fatTable[i].isFree = false;
28         idAnterior = i;
29         filellenado++;
30         if (filellenado >= fileSize)
31         {
32             fatTable[i].nextBlock = -1;
33             break;
34         }
35     }
36     if (filellenado < fileSize)
37     {
38         deleteFile(idInicial);
39         idInicial = -1;
40     }
41
42     return idInicial;
43 }
```

```
1 // Mostrar el estado actual de la FAT
2 void printFAT()
3 {
4     cout << "\nÍndice\tLibre\tSiguiente\n";
5     for (int i = 0; i < DISK_SIZE; ++i)
6     {
7         cout << i << "\t" << (fatTable[i].isFree ? "Si" : "No") << "\t"
8             << fatTable[i].nextBlock << "\n";
9     }
10 }
11
12 // Mostrar los bloques que pertenecen a un archivo
13 void displayFileBlocks(int startBlock)
14 {
15     cout << "\nÍndice\tLibre\tSiguiente\n";
16     int idBlock = startBlock;
17     while (idBlock != -1)
18     {
19         int siguienteBloque = fatTable[idBlock].nextBlock;
20         cout << idBlock << "\t" << (fatTable[idBlock].isFree ? "Si" : "No") << "\t"
21             << fatTable[idBlock].nextBlock << "\n";
22         idBlock = siguienteBloque;
23     }
24 }
```

```

1 // Menú interactivo
2 void menu()
3 {
4     int option, size, startBlock;
5
6     do
7     {
8         cout << "\n--- Menú FAT ---\n";
9         cout << "1. Crear archivo\n";
10        cout << "2. Eliminar archivo\n";
11        cout << "3. Mostrar tabla FAT\n";
12        cout << "4. Mostrar bloques de un archivo\n";
13        cout << "5. Salir\n";
14        cout << "Seleccione una opción: ";
15        cin >> option;
16
17        switch (option)
18        {
19            case 1:
20                cout << "Tamaño del archivo (en bloques): ";
21                cin >> size;
22                startBlock = createFile(size);
23                if (startBlock != -1)
24                    cout << "Archivo creado. Bloque inicial: " << startBlock << "\n";
25                else
26                    cout << "No hay espacio suficiente para el archivo.\n";
27                break;
28
29            case 2:
30                cout << "Bloque inicial del archivo a eliminar: ";
31                cin >> startBlock;
32                deleteFile(startBlock);
33                cout << "Archivo eliminado.\n";
34                break;
35
36            case 3:
37                printFAT();
38                break;
39
40            case 4:
41                cout << "Bloque inicial del archivo: ";
42                cin >> startBlock;
43                displayFileBlocks(startBlock);
44                break;
45
46            case 5:
47                cout << "Saliendo...\n";
48                break;
49
50            default:
51                cout << "Opción inválida.\n";
52        }
53
54    } while (option != 5);
55 }

```



Enlace de Github

<https://github.com/JRafaelLobo/LabFat.git>

Ejecución del Programa

La creación de dos archivos

Creación de archivo de tamaño de 5 bloques

--- Menú FAT ---	Índice	Libre	Siguiente
1. Crear archivo	0	No	1
2. Eliminar archivo	1	No	2
3. Mostrar tabla FAT	2	No	3
4. Mostrar bloques de un archivo	3	No	4
5. Salir	4	No	-1
Seleccione una opción: 1	5	Sí	-1
Tamaño del archivo (en bloques): 5	6	Sí	-1
Archivo creado. Bloque inicial: 0	7	Sí	-1
--- Menú FAT ---	8	Sí	-1
1. Crear archivo	9	Sí	-1
2. Eliminar archivo	10	Sí	-1
3. Mostrar tabla FAT	11	Sí	-1
4. Mostrar bloques de un archivo	12	Sí	-1
5. Salir	13	Sí	-1
Seleccione una opción: 3	14	Sí	-1

Creación de archivo de tamaño de 7 bloques

--- Menú FAT ---	Índice	Libre	Siguiente
1. Crear archivo	0	No	1
2. Eliminar archivo	1	No	2
3. Mostrar tabla FAT	2	No	3
4. Mostrar bloques de un archivo	3	No	4
5. Salir	4	No	-1
Seleccione una opción: 1	5	No	6
Tamaño del archivo (en bloques): 7	6	No	7
Archivo creado. Bloque inicial: 5	7	No	8
--- Menú FAT ---	8	No	9
1. Crear archivo	9	No	10
2. Eliminar archivo	10	No	11
3. Mostrar tabla FAT	11	No	-1
4. Mostrar bloques de un archivo	12	Sí	-1
5. Salir	13	Sí	-1
Seleccione una opción: 3	14	Sí	-1

La eliminación de uno.

Eliminación del archivo 0

```
--- Menú FAT ---
1. Crear archivo
2. Eliminar archivo
3. Mostrar tabla FAT
4. Mostrar bloques de un archivo
5. Salir
Seleccione una opción: 2
Bloque inicial del archivo a eliminar: 0
Archivo eliminado.

--- Menú FAT ---
1. Crear archivo
2. Eliminar archivo
3. Mostrar tabla FAT
4. Mostrar bloques de un archivo
5. Salir
Seleccione una opción: 3
```

Índice	Libre	Siguiente
0	Sí	-1
1	Sí	-1
2	Sí	-1
3	Sí	-1
4	Sí	-1
5	No	6
6	No	7
7	No	8
8	No	9
9	No	10
10	No	11
11	No	-1
12	Sí	-1
13	Sí	-1

Preguntas de Reflexión

1. ¿Qué analogía existe entre la FAT y la tabla de páginas en memoria virtual?
Estas mismas funcionan de la misma manera en que las páginas de memoria, ya que divide la memoria en bloques. El sistema FAT separa los archivos en bloques cada uno apuntando a la siguiente hasta que llegue al último archivo. Mientras que la tabla de páginas en memoria virtual funcionan de la misma manera en que junta datos para mover enteramente un bloque de memoria, el cual va a ser utilizada.
2. ¿Qué tipo de fragmentación puede observarse en FAT?
Puede ocurrir muy seguido la fragmentación externa, el cual se refiere a la separación de ubicaciones cercanas del archivo. Sin embargo, también pasa en menor medida la fragmentación interna, el cual sucede cuando un archivo no completa el clúster, sin embargo eso es poco significativo.
3. ¿Cómo se podría mejorar el acceso a archivos en FAT?
La fragmentación externa del disco afecta bastante en los tiempos de acceso, por ello es importante reorganizar físicamente los archivos del disco. Se puede incrementar los tamaños de los clusters, ya que eso implica menos recorrido en la tabla FAT.