

Tarea programada 2 – mini simulación de disco

Debe programar una **pequeña simulación de como se almacenan los archivos en un file system**. Tome en cuenta lo siguiente:

- El **disco puede ser una matriz de tamaño $a*b$** . También puede ser una lista enlazada o un vector.
- Cada **entrada de la matriz representa un bloque de tamaño N bytes**
- El **primer bloque de esta matriz NO se debe utilizar**, ya que es el sector cero del disco. Se puede dejar vacío o utilizar un carácter en específico para denotar que está reservado.
- Debe elegir una **porción de ese disco para almacenar los nodos i** correspondientes a cada archivo. Recuerde la estructura de los nodos i vista en clase.
 - Puede hacer listas enlazadas para cada nodo i y tener las referencias a cada una en un bloque de disco.
 - Puede tomar el primer bloque para el encabezado del archivo y los otros 7 bloques para direcciones donde se encuentra el siguiente bloque de datos.
- Debe elegir **otra porción del disco para agregar los bloques de datos de los archivos**.
- El programa debe **permitir consultar los documentos almacenados**, agregar un nuevo documento o agregar a uno existente.

Puntos que **NO se deben programar** para simplificar el programa:

- NO se necesita la **opción de borrar archivos o parte de un archivo**.
- **NO se va a trabajar con directorios**.
- **NO se debe agregar nada en el sector cero del disco**.
- **NO necesita estructura para bloques libres (bitmap o lista)**
- Los **archivos NO pueden crecer sin límite**. Es decir, el **máximo de un archivo puede ser lo que una sola estructura de nodos i de 8 bloques pueda manejar**.

Para la entrega:

- La tarea se puede realizar en parejas
- Tiene un valor de 10%
- La fecha de entrega: 4 de diciembre
- Se puede realizar en C, C++, C# o Python
- El programa debe **tener un "Read Me" para saber como correr el programa**, la salida esperada y cualquier restricción.

Ejemplo de como se puede ver este file system

Sector 0	Referencia al nodo l (1)	Referencia al nodo l (2)						
Datos 1	Datos 1	Datos 1	Datos 1	Datos 1	Datos 1	Datos 2	Datos 2	Datos 2
Datos 2	Datos 2	Datos 2						

Otro ejemplo, distinta implementación

Sector 0	Nodo l (1) Encabezado -> (2)	Dirección de disco (60) -> (3)	Dirección de disco (61) -> (4)	Dirección de disco (62) -> (5)	Dirección de disco (63) -> (6)	Dirección de disco (64) -> (8)	Dirección de disco (65) -> (-1)	Nodo l (2) Encabezado -> (10)
Dirección de disco (66) -> (11)	Dirección de disco (67) -> (12)	Dirección de disco (68) -> (13)	Dirección de disco (69) -> (14)	Dirección de disco (70) -> (15)	Dirección de disco (71) -> (16)	Dirección de disco (72) -> (-1)		
Datos 1	Datos 1	Datos 1	Datos 1	Datos 1	Datos 1	Datos 2	Datos 2	Datos 2
Datos 2	Datos 2	Datos 2						