

Trabajo Corto 4 (%)

Fecha de Entrega:

# Sistema de Archivos Simple

## Descripción de la tarea

En esta tarea, los estudiantes deberán implementar un sistema de archivos simple en C que emule las operaciones básicas de un sistema de archivos. Este sistema manejará operaciones como crear archivos, escribir en ellos, leer su contenido, eliminar archivos y listar el contenido del directorio. Los estudiantes deben simular la estructura de un sistema de archivos básico utilizando una tabla de archivos y bloques de datos.

## Especificaciones

### Estructura del Sistema de Archivos

- El sistema simulado debe dividirse en bloques de un tamaño fijo (por ejemplo, 512 bytes por bloque).
- Los archivos deben estar representados por estructuras de control que almacenen información como el nombre del archivo, el tamaño, y los bloques ocupados.
- El sistema debe simular un directorio raíz donde se almacenarán los archivos.

### Operaciones Soportadas

- CREATE <nombre\_archivo> <tamaño\_bytes>**: Crea un archivo con el nombre dado y asigna espacio en los bloques simulados para almacenar el archivo.
- WRITE <nombre\_archivo> <offset> <data>**: Escribe datos en el archivo, comenzando en el offset especificado. Si el archivo no existe o no hay suficiente espacio, se debe retornar un error.
- READ <nombre\_archivo> <offset> <tamaño>**: Lee una cantidad específica de bytes desde el offset de un archivo dado.
- DELETE <nombre\_archivo>**: Elimina el archivo del sistema y libera los bloques que ocupaba.
- LIST**: Lista todos los archivos almacenados en el sistema, mostrando su nombre y tamaño.

## Simulación de los Bloques

- Se debe implementar un manejo de bloques de memoria simulados para la escritura y lectura de datos.
- Implementar una tabla de archivos que mantenga el seguimiento de qué bloques están ocupados y qué archivos los utilizan.

## Manejo de Errores

El sistema debe manejar errores como:

- Falta de espacio en los bloques simulados.
- Intentar leer o escribir en un archivo que no existe.
- Sobrepasar los límites de un archivo.

## Restricciones

- Limitar el número máximo de archivos y el tamaño máximo de cada archivo. (Por ejemplo, un total de 100 archivos y 1 MB de almacenamiento simulado).

## Ejemplo de ejecución

```
> CREATE archivo1.txt 1000
> WRITE archivo1.txt 0 "Hola , mundo"
> READ archivo1.txt 0 11
Salida: "Hola , mundo"
> LIST
archivo1.txt - 1000 bytes
> DELETE archivo1.txt
> LIST
(no hay archivos)
```

## Criterios de evaluación

- **Funcionalidad (40%)**: Implementación correcta de las operaciones solicitadas (CREATE, WRITE, READ, DELETE, LIST).
- **Gestión de Bloques (25%)**: Correcta simulación del manejo de bloques de memoria y gestión del espacio de almacenamiento.
- **Manejo de Errores (15%)**: Correcto manejo de los errores, como archivo no encontrado, falta de espacio, entre otros.
- **Estructura del Código y Comentarios (10%)**: Código limpio, bien estructurado y comentado adecuadamente.

- **Documentación (10%):** Documentación clara que explique el funcionamiento del código y las decisiones de diseño.

## **Entregable**

Debe presentar un archivo comprimido que incluya:

- El código fuente del sistema de archivos.
- Un archivo de texto con ejemplos de uso y pruebas que incluyan la creación, escritura, lectura, eliminación y listado de archivos.
- Un informe breve explicando las estructuras de datos utilizadas y las principales decisiones de diseño.