

Proyecto 2 - (Micro) Sistema de archivos multihilos

Descripción del proyecto:

Se deberá de realizar y desarrollar un programa que pueda obtener, crear y modificar información en el micro sistema de archivos que desarrollo el profesor para la Facultad de Ingeniería, FiUnamFS.

Tendrá que ser capaz de:

1. Listar los contenidos del directorio.
2. Copiar uno de los archivos de dentro del FiUnamFS hacia tu sistema.
3. Copiar un archivo de tu computadora hacia tu FiUnamFS.
4. Eliminar un archivo del FiUnamFS.
5. El programa que desarrollen debe contar, por lo menos, dos hilos de ejecución, operando concurrentemente, y que se comuniquen sus estados mediante mecanismos de sincronización.

Entorno de desarrollo:

Autor: Medrano Solano Enrique

Entorno: Windows 11 Pro

Versión del S.O: 22H2.1000.22H2

Lenguaje de desarrollo: Python3 – Python

Versión: 3.11.10 y 3.12.7 ambas de 64 bits

IDE: Visual Studio Code

Terminal: Git bash | MINGW64

Modo de ejecución: Python proyecto2.py

```
enrique@QUIQUE_PC MINGW64 ~/Documents/UNAM/5to semestre/Sistemas Operativos/sistop-2025-1/proyectos/micro-sist-de-arch-multihilos
/MedranoEnrique (main)
$ python proyecto2.py |
```

Descripción del programa desarrollado:

En este apartado se explicará y documentaran las partes más esenciales del código presentado con el nombre de “proyecto2.py”.

En general el programa consiste en implementar un sistema de archivos simple, llamado FiUnamFS. Este sistema de archivos permite realizar operaciones básicas de administración de archivos dentro de un entorno controlado. Las operaciones principales incluyen listar, copiar desde y hacia el sistema de archivos, y eliminar archivos. Se emplea una interfaz gráfica (GUI) dentro de la terminal en conjunto con un enfoque multihilos para realizar estas tareas en paralelo, gestionando el acceso concurrente al archivo de disco mediante mecanismos de sincronización.

Listar el contenido del directorio: Esta función lo que hace es recorrer el directorio de FiUnamFS y muestra los nombres y tamaños de cada uno de los archivos almacenados, dicha información se obtiene de los clusters de datos que están organizados en una estructura de 64 bytes por entrada como se especificó.

Copiar archivos de FiUnamFS a local: Permite transferir un archivo desde FiUnamFS al sistema local (en este caso mi computadora). La función busca el archivo por su nombre a través de comparaciones, de esta manera cuando se da con él procede a copiar, de igual manera esta diseñado para el caso en donde no exista el archivo y se le haga saber al usuario.

Copia de archivos de local a FiUnam: Dentro de esta función se hace el mismo proceso, pero en viceversa, aunque esta también contempla la verificación de un espacio de entrada libre en el directorio, buscando espacio en los clusters de datos para almacenar el archivo indicado.

Eliminar archivo de FiUnamFS: Función creada para eliminar un archivo seleccionado de FiUnamFS, remplazando su entrada en el directorio principal por un símbolo especial (#), sirviendo como indicador para futuros archivos que se quieran almacenar.

Implementación de una verificación por función de superbloque: Este apartado del código tiene la función de fungir como verificador y controlador de las acciones operacionales del programa, de tal manera que cada que se solicita hacer una operación se verifican los primeros Bytes del archivo del “disco” en este caso de “fiunamfs.img” donde se comprueba que contenga la cadena “FiUnamFS”, de esta manera se evita que siga avanzado el programa para operaciones futuras y de errores imprevistos. Notificando al usuario de cambiar el “disco” a uno válido.

Estructuras de datos y formato de Clusters: en este caso estos dos aspectos cumplen con los requerimientos propuestos, donde el disco queda definido por 1440 KB, y cada cluster de 1024 bytes.

Para el caso de las entradas que ocupan 64 bytes en total están distribuidas de la siguiente manera:

Tipo → pertenece al 1 byte el cual indica si la entrada está ocupada ‘.’ O libre ‘#’.

Nombre → pertenecen a los siguientes 15 bytes, indicando el nombre del archivo.

Tamaño → 4 bytes, los cuales indican el tamaño del archivo.

Inicio_cluster → 4 bytes, indican el cluster de inicio de cada uno.

Fechas → 14 bytes, indican el formato de la forma YYYYMMDDHHMMSS.

Implementación de multihilos y sincronización:

El programa trata de implementar un diseño de multihilos para realizar las operaciones en paralelo, mejorando la eficiencia en un entorno de multitarea, en este caso para cada uno de los requerimientos básicos, siendo que cada operación de copia, listar y eliminar se ejecuta en su propio hilo, utilizando un mecanismo de bloqueo con (disk_lock) para asegurar que un solo hilo pueda acceder al archivo de disco en un momento dado, de esta manera se evita problemas de concurrencia.

Es por esto que se tiene 3 funciones para cada una de las operaciones básicas, donde se asegura que no haya o evite las interferencias entre hilos concurrentes,

este mecanismo permite que las operaciones de lectura y escritura en el archivo no se empalmen, creando o reescribiendo información en los datos de manera segura.

Ejemplo de programa:

Para el caso del menú se ve así:

```
enric@QUIQUE_PC MINGW64 ~/Documents/UNAH/5to semestre/Sistemas Operativos/sistop-2025-1/proyectos/micro-sist-de-arch-multihilos/MedranoEnrique (main)
$ python proyecto2.py

--- Menú de FiUnamFS ---
1. Lista de contenido del directorio
2. Copiar archivo de FiUnamFS al sistema local
3. Copiar archivo del sistema local a FiUnamFS
4. Eliminar archivo de FiUnamFS
5. Salir
Selecciona una opción: █
```

En caso de querer ver la lista del directorio seleccionar 1:

```
$ python proyecto2.py

--- Menú de FiUnamFS ---
1. Lista de contenido del directorio
2. Copiar archivo de FiUnamFS al sistema local
3. Copiar archivo del sistema local a FiUnamFS
4. Eliminar archivo de FiUnamFS
5. Salir
Selecciona una opción: 1

Contenido del directorio:

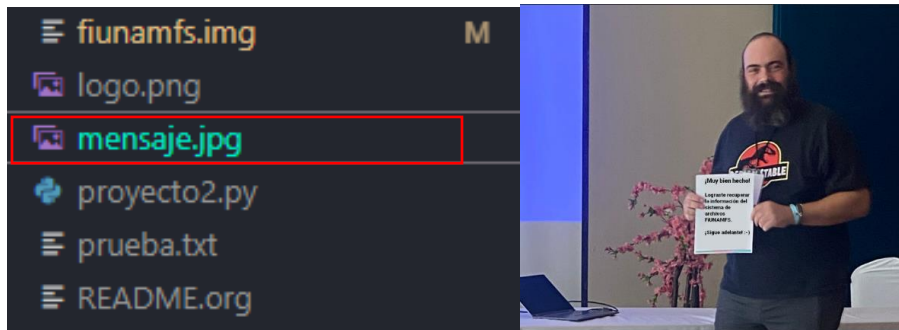
Archivos: README.org    -> Tamaño: 31081 bytes
Archivos: logo.png      -> Tamaño: 126423 bytes
Archivos: mensaje.jpg   -> Tamaño: 254484 bytes

--- Menú de FiUnamFS ---
1. Lista de contenido del directorio
2. Copiar archivo de FiUnamFS al sistema local
3. Copiar archivo del sistema local a FiUnamFS
4. Eliminar archivo de FiUnamFS
5. Salir
Selecciona una opción: █
```

En este caso aparecen los datos y archivos originales proporcionados por el profesor, en caso de querer copiar un archivo se selecciona la opción 2:

```
--- Menú de FiUnamFS ---
1. Lista de contenido del directorio
2. Copiar archivo de FiUnamFS al sistema local
3. Copiar archivo del sistema local a FiUnamFS
4. Eliminar archivo de FiUnamFS
5. Salir
Selecciona una opción: 2
Ingresa el nombre del archivo a copiar de FiUnamFS: mensaje.jpg
Archivo 'mensaje.jpg' copiado al sistema local correctamente!!!
```

Una vez hecho esto, se pasa a ver la carpeta, en este caso en el mismo VS Code se puede ver los archivos generados:



En caso de querer agregar un archivo local hacia FiUnamFS se selecciona la opción 3, para este ejemplo se agregará “prueba.txt” y “proyecto2.py”

```
--- Menú de FiUnamFS ---
1. Lista de contenido del directorio
2. Copiar archivo de FiUnamFS al sistema local
3. Copiar archivo del sistema local a FiUnamFS
4. Eliminar archivo de FiUnamFS
5. Salir
Selecciona una opción: 3
Ingrese el nombre del archivo a FiUnamFS: prueba.txt
Archivo prueba.txt copiado correctamente a FiUnamFS!!!
```

```
--- Menú de FiUnamFS ---
1. Lista de contenido del directorio
2. Copiar archivo de FiUnamFS al sistema local
3. Copiar archivo del sistema local a FiUnamFS
4. Eliminar archivo de FiUnamFS
5. Salir
Selecciona una opción: 3
Ingrese el nombre del archivo a FiUnamFS: proyecto2.py
Archivo proyecto2.py copiado correctamente a FiUnamFS!!!
```

Si se vuelve a observar la lista del directorio (opción 1), ya aparecerá actualizado:

```
Selecciona una opción: 1
Contenido del directorio:
Archivos: README.org -> Tamaño: 31081 bytes
Archivos: prueba.txt -> Tamaño: 139 bytes
Archivos: logo.png -> Tamaño: 126423 bytes
Archivos: proyecto2.py -> Tamaño: 11048 bytes
Archivos: mensaje.jpg -> Tamaño: 254484 bytes
```

Finalmente, si se quiere eliminar un archivo de FiUnamFS, se selecciona la opción 4, para ese ejemplo se eliminará prueba.txt

```
--- Menú de FiUnamFS ---  
1. Lista de contenido del directorio  
2. Copiar archivo de FiUnamFS al sistema local  
3. Copiar archivo del sistema local a FiUnamFS  
4. Eliminar archivo de FiUnamFS  
5. Salir  
Selecciona una opción: 4  
Ingresa el nombre del archivo a eliminar de FiUnamFS: prueba.txt  
Archivo prueba.txt eliminado correctamente de FiUnamFS!!!
```

Se vuelve a ver la lista y ya no aparecerá:

```
--- Menú de FiUnamFS ---  
1. Lista de contenido del directorio  
2. Copiar archivo de FiUnamFS al sistema local  
3. Copiar archivo del sistema local a FiUnamFS  
4. Eliminar archivo de FiUnamFS  
5. Salir  
Selecciona una opción: 1  
Contenido del directorio:  
  
Archivos: README.org    -> Tamaño: 31081 bytes  
Archivos: logo.png      -> Tamaño: 126423 bytes  
  
Archivos: proyecto2.py  -> Tamaño: 11048 bytes  
Archivos: mensaje.jpg   -> Tamaño: 254484 bytes
```

Algunos de los mensajes que anticipan errores para el usuario son los siguientes:

```
--- Menú de FiUnamFS ---  
1. Lista de contenido del directorio  
2. Copiar archivo de FiUnamFS al sistema local  
3. Copiar archivo del sistema local a FiUnamFS  
4. Eliminar archivo de FiUnamFS  
5. Salir  
Selecciona una opción: 6  
Opción ingresada ¡no válida!, por favor intenta de nuevo :)
```

```
--- Menú de FiUnamFS ---  
1. Lista de contenido del directorio  
2. Copiar archivo de FiUnamFS al sistema local  
3. Copiar archivo del sistema local a FiUnamFS  
4. Eliminar archivo de FiUnamFS  
5. Salir  
Selecciona una opción: 2  
Ingresa el nombre del archivo a copiar de FiUnamFS: regresión.docx  
Archivo 'regresión.docx' no encontrado en FiUnamFS.
```

```
--- Menú de FiUnamFS ---  
1. Lista de contenido del directorio  
2. Copiar archivo de FiUnamFS al sistema local  
3. Copiar archivo del sistema local a FiUnamFS  
4. Eliminar archivo de FiUnamFS  
5. Salir  
Selecciona una opción: 3  
Ingrese el nombre del archivo a FiUnamFS: hola.txt  
El archivo hola.txt no existe en el sistema local.
```