



Universidad Nacional Autónoma de México

Materia: Sistemas Operativos

Profesor: Gunnar Eyal Wolf Iszaevich

Tenorio Martinez Jesus Alejandro

Proyecto: Microsistema de archivos multihilos

- **Introducción**

Este proyecto tiene el propósito de implementar los conocimientos aprendidos en la materia de sistemas operativos, implementando un microsistema de archivos, utilizando hilos para evitar condiciones de carrera.

Un sistema de archivos es un elemento que controla cómo el almacenamiento y recuperación de datos. Sin un sistema de archivos, los datos colocados en un medio de almacenamiento serían un gran cuerpo de datos sin manera de saber dónde termina un dato y comienza el siguiente. En este proyecto en particular se utiliza un sistema de archivos contigua sin subdirectorios, pero este tiene una estructura que permite buscar y escribir archivos.

- El sistema de archivos tiene las siguientes características:
- tiene una longitud de 1440kb,
- todas las cadenas de textos son ASCII 8-bit,
- todos los enteros son representados como valores de 32 bits en formato little endian
- los sectores miden 256b,
- cada clúster mide cuatro sectores,
- no maneja tablas de particiones

Así mismo el primer cluster (superbloque) tiene la siguiente estructura (64b):

- 0–8: Para identificación, el nombre del sistema de archivos.
- 10–14: Versión de la implementación.
- 20–35: Etiqueta del volumen
- 40–44: Tamaño del cluster en bytes
- 45–49: Número de clusters que mide el directorio
- 50–54: Número de clusters que mide la unidad completa

El directorio esta en los cluster del 1 al 4, y el cada entrada que mide 64b, tiene las siguientes características:

- 1–15 Nombre del archivo
- 16–20 Tamaño del archivo, en bytes
- 20–23 Cluster inicial
- 24–37 Hora y fecha de creación del archivo, especificando AAAAMMDDHHMMSS
- 38–51 Hora y fecha de última modificación del archivo, especificando AAAAMMDDHHMMSS

- 52–64 Espacio no utilizado

- **Desarrollo**

La estrategia principal de este programa consiste en “montar” el sistema de archivos como si fuera parte del sistema de archivos del usuario, se utilizó el lenguaje de programación Python para realizar este código.

El código puede proporcionar a varios usuarios los archivos de “FiunamFS”, es por eso que se implementaron hilos, que simulan a estos y se implementó una sincronización para evitar condiciones de carrera

Este programa fue desarrollado en un sistema operativo Windows y pretende simular la terminal de este sistema operativo, así que se recomienda trabajar en este entorno.

- *Ejecutando el código*

Para ejecutar el programa, se necesita tener en una misma carpeta lo siguiente:

- main.py (Archivo Python a ejecutar)
- contenido.py
- copiar_a_fiunamfs.py
- copiar_a_sist.py
- eliminar_archivo.py
- montaje.py
- mostrararchivos.py
- ReconocerEntradas.py
- Sincronización.py
- fiunamfs.img

Así mismo es indispensable tener al menos la última versión de Python instalada y asegurarse poder importar las siguientes bibliotecas:

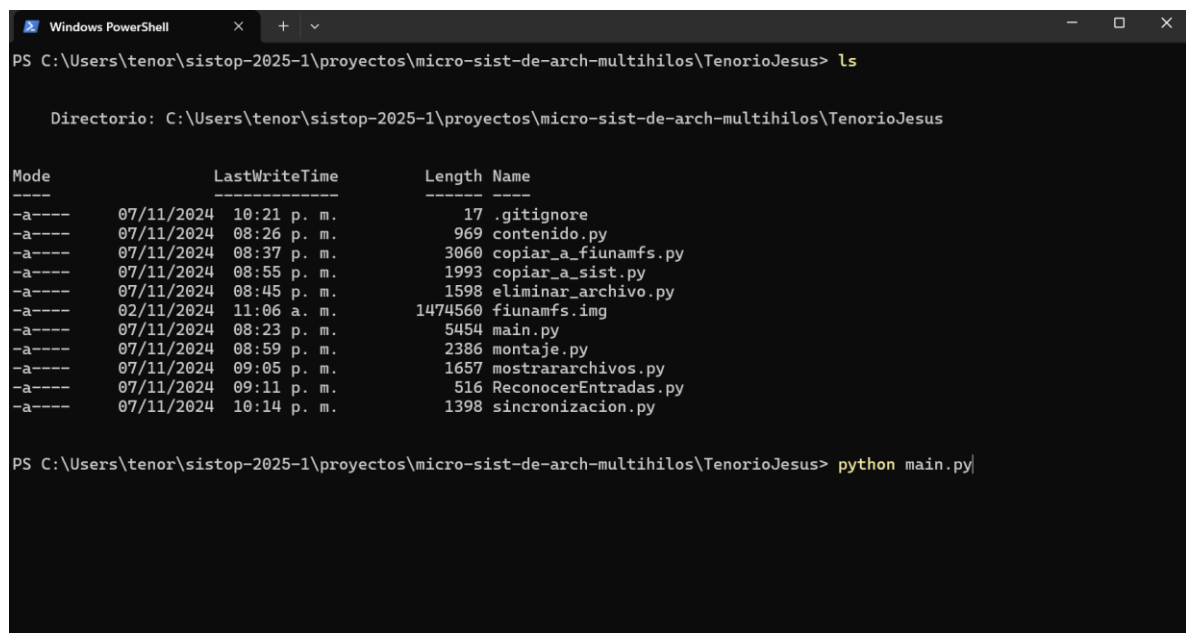
- os*
- struct*
- threading*
- from ReconocerEntradas import Reconocer
- from pathlib import Path
- from datetime import datetime
- from contenido import contenido_archivo

Si no se tienen algunas de las que no están marcadas con “*”, se recomienda verificar que todos los archivos Python anteriormente mencionados estén en la carpeta.

Consideraciones adicionales: Aunque resulte redundante mencionarlo, es preferente que se utilice un sistema Windows, pues algunos comando como “os.system(‘cls’)” podrían generar un error, además de que se pierde la experiencia del programa al “ocultarse del usuario”, pues el programa está diseñado para que quien este ejecutando el programa no se de cuenta que esta ejecutando un programa, sino que sigue usando la interfaz de terminal de Windows.

Al trabajar con directorios, puede que haya problemas con estos dependiendo como se manejen en tu sistema operativo

Una vez que se navegue hasta la carpeta en la terminal, proceder a llamar al codigo “main.py” con Python:



```
Windows PowerShell
PS C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus> ls

Directorio: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus

Mode                LastWriteTime         Length Name
----                -
-a-----         07/11/2024   10:21 p. m.             17 .gitignore
-a-----         07/11/2024   08:26 p. m.             969 contenido.py
-a-----         07/11/2024   08:37 p. m.          3060 copiar_a_fiunamfs.py
-a-----         07/11/2024   08:55 p. m.          1993 copiar_a_sist.py
-a-----         07/11/2024   08:45 p. m.          1598 eliminar_archivo.py
-a-----         02/11/2024   11:06 a. m.       1474560 fiunamfs.img
-a-----         07/11/2024   08:23 p. m.          5454 main.py
-a-----         07/11/2024   08:59 p. m.          2386 montaje.py
-a-----         07/11/2024   09:05 p. m.          1657 mostrararchivos.py
-a-----         07/11/2024   09:11 p. m.           516 ReconocerEntradas.py
-a-----         07/11/2024   10:14 p. m.          1398 sincronizacion.py

PS C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus> python main.py|
```

El programa procederá a montar el sistema de archivos en nuestro sistema y se meterá al sistema como si fuera una carpeta:

```

PS C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus> python main.py

Montando FiUnamFs...

Sistema: FiunamFS

Version: 25-1

etiqueta volumen: 1646292565

Tamño cluster: 1024

Numero de clusteres: 1024

numero de cluster de la unidad completa: 1440
PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS>

```

Una vez dentro del sistema podemos ejecutar instrucciones tradicionales de Windows para interactuar con los archivos.

1- Listar Archivos: para listar los archivos se utiliza el típico comando “ls”

```

PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS> ls

Eperando el recurso...

Mode                LastWriteTime         Length Name
----                -
10
11
11
-a---l      2024\10\24    19:50:56      31081 README.org
-a---l      2024\10\24    19:50:56     126423 logo.png
-a---l      2024\10\24    19:50:56     254484 mensaje.jpg

PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS>

```

Al introducir “ls”, el sistema comenzara a verificar si algún otro usuario esta utilizando el recurso “fiunamfs.img”

2- Copiar un archivo de FiUnamFS a nuestro sistema: Para copiar un archivo se utiliza el típico comando de Windows “copy”

```

Mode                LastWriteTime         Length Name
----                -
10
11
11
-a---l      2024\10\24    19:50:56      31081 README.org
-a---l      2024\10\24    19:50:56     126423 logo.png
-a---l      2024\10\24    19:50:56     254484 mensaje.jpg

PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS> copy "C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS\logo.png" "C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\"

```

```

Eperando el recurso...

Archivo 'logo.png' copiado a 'C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\\logo.png'

PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS>

```

El programa primero verifica la disponibilidad del disco ante cualquier otro hilo

Así, el archivo se copea a una ruta cualquiera dada, en este caso movimos el archivo logo.png hacia la carpeta del proyecto.

Podemos ir a verificar yendo a la carpeta. Y eso se hace con “cd ..”, recordar que es como se haría en la terminal de Windows típicamente

```
PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS> cd ..

Desmontando FiunamFs...

PS C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus> ls

Directorio: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus

Mode                LastWriteTime         Length Name
----                -
d-----          07/11/2024 11:25 p. m.           35  __pycache__
-a-----          08/11/2024 12:23 a. m.            969  contenido.py
-a-----          07/11/2024 08:26 p. m.          3060  copiar_a_fiunamfs.py
-a-----          07/11/2024 08:37 p. m.          1993  copiar_a_sist.py
-a-----          07/11/2024 08:55 p. m.          1598  eliminar_archivo.py
-a-----          02/11/2024 11:06 a. m.         1474560  fiunamfs.img
-a-----          08/11/2024 12:24 a. m.          126423  logo.png
-a-----          07/11/2024 11:34 p. m.           5427  main.py
-a-----          07/11/2024 08:59 p. m.           2386  montaje.py
-a-----          07/11/2024 09:05 p. m.           1657  mostrararchivos.py
-a-----          07/11/2024 09:11 p. m.            516  ReconocerEntradas.py
-a-----          07/11/2024 10:14 p. m.           1398  sincronizacion.py

PS C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus>
```

Tener en cuenta que ahora existe el archivo logo.png

3- Copiar un archivo del sistema a FiunamFs: para copiar un archivo el típico comando de windows “copy”

```
PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS> copy "C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\montaje.py" "C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS\"
```

El programa verifica la disponibilidad del disco por medio de mecanismo de sincronización.

```
Eperando el recurso...

Archivo 'montaje.py' escrito exitosamente en FiUnamFS.
PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS>
```

Al ejecutar esta parte, si leemos nuevamente los archivos de FiunamFS, conseguiremos observar que ya existe el archivo especificado.

```
Eperando el recurso...

Archivo 'montaje.py' escrito exitosamente en FiUnamFS.
PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS> ls

Eperando el recurso...

Mode                LastWriteTime         Length Name
----                -
10
11
-a---l      2024\10\24      19:50:56      31081 README.org
-a---l      2024\11\08      00:30:05      2375 montaje.py
-a---l      2024\10\24      19:50:56      126423 logo.png
-a---l      2024\10\24      19:50:56      254484 mensaje.jpg

PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS>
```

4- Hilos y sincronización:

El programa “main.py” esta repleto de sincronización entre hilos (usuarios) pero es evidente que no vamos a estar escribiendo varios archivos en ambos sistemas para evidenciar el funcionamiento del programa, así que usaremos un ejemplo practico que permite usar “ls” y la sincronización entre procesos para evidenciar que “main.py” debería funcionar si se ejecutara de manera paralela. Al ejecutar sincronización.py, después de montar el programa varios usuarios intentaran leer el recurso FiunamFS, pero solo uno lo puede hacer a la vez.

```
PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS> cd ..

Desmontando FiunamFs...

PS C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus> python sincronizacion.py
C:\Users\tenor\AppData\Local\Programs\Python\Python311\python.exe: can't open file 'C:\\Users\\tenor\\sistop-2025-1\\proyectos\\micro-sist-de-arch-multihilos\\TenorioJesus\\sincronizacion.py': [Errno 2] No such file or directory
PS C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus> python sincronizacion.py

Montando FiUnamFs

Sistema: FiunamFS

Version: 25-1

etiqueta volumen: 1646292565

Tamaño cluster: 1024

Numero de clusteres: 1024

numero de cluster de la unidad completa: 1440

Usuario solicitando 'ls': Esperando el recurso...

Usuario solicitando 'ls': Esperando el recurso...
```

```

Usuario solicitando 'ls': Esperando el recurso...

Usuario solicitando 'ls': Esperando el recurso...

Usuario solicitando 'ls': Esperando el recurso...
Mode                LastWriteTime         Length Name
----                -
10                  .
11                  ..
-a---l      2024\10\24    19:50:56      31081 README.org
-a---l      2024\11\08    00:30:05      2375 montaje.py
-a---l      2024\10\24    19:50:56     126423 logo.png
-a---l      2024\10\24    19:50:56     254484 mensaje.jpg

Mode                LastWriteTime         Length Name
----                -
10                  .
11                  ..
-a---l      2024\10\24    19:50:56      31081 README.org
-a---l      2024\11\08    00:30:05      2375 montaje.py
-a---l      2024\10\24    19:50:56     126423 logo.png
-a---l      2024\10\24    19:50:56     254484 mensaje.jpg

Usuario solicitando 'ls': Esperando el recurso...
Mode                LastWriteTime         Length Name
----                -

```

```

Usuario solicitando 'ls': Esperando el recurso...
Mode                LastWriteTime         Length Name
----                -
10                  .
11                  ..
-a---l      2024\10\24    19:50:56      31081 README.org
-a---l      2024\11\08    00:30:05      2375 montaje.py
-a---l      2024\10\24    19:50:56     126423 logo.png
-a---l      2024\10\24    19:50:56     254484 mensaje.jpg

Usuario solicitando 'ls': Esperando el recurso...
Mode                LastWriteTime         Length Name
----                -
10                  .
11                  ..
-a---l      2024\10\24    19:50:56      31081 README.org
-a---l      2024\11\08    00:30:05      2375 montaje.py
-a---l      2024\10\24    19:50:56     126423 logo.png

```

- 5- Eliminar Archivo: Para eliminar un archivo se puede escribir “del “seguido del nombre del archivo a eliminar, tal y como se hace en la terminal de Windows estándar


```

PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS> ls

Eperando el recurso...

Mode                LastWriteTime         Length Name
----                -
10
11
-a---l      2024\10\24    19:50:56      31081 README.org
-a---l      2024\11\08    00:30:05      2375 montaje.py
-a---l      2024\10\24    19:50:56     126423 logo.png
-a---l      2024\10\24    19:50:56     254484 mensaje.jpg

PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS> del README.org

Eperando el recurso...

Archivo 'README.org' eliminado exitosamente.

```

Volviendo a checar los archivos en la carpeta FiunamFS...

```

PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS> del README.org

Eperando el recurso...

Archivo 'README.org' eliminado exitosamente.
PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS> ls

Eperando el recurso...

Mode                LastWriteTime         Length Name
----                -
10
11
-a---l      2024\11\08    00:30:05      2375 montaje.py
-a---l      2024\10\24    19:50:56     126423 logo.png
-a---l      2024\10\24    19:50:56     254484 mensaje.jpg

PS: C:\Users\tenor\sistop-2025-1\proyectos\micro-sist-de-arch-multihilos\TenorioJesus\FiunamFS> |

```

En conclusión: El sistema de archivos elaborado pretende simular la terminal de windows con el fin de que el usuario no se de cuenta que esta usando un programa, el código también me ayudó a reforzar los temas de sistemas de archivos y su organización en el medio lógico y físico, también fue bastante retador, pues al no estar tan familiarizado con el lenguaje, me costó utilizar funciones de Python.