



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

FACULTAD DE INGENIERÍA

**DIVISIÓN DE INGENIERÍA
ELÉCTRICA**

Sistemas Operativos

Proyecto 2

ALUMNOS:

Cuevas Quintana Amir

De la Rosa Flores Fernando

PROFESOR;

Gunnar Wolf

GRUPO 6

2025-1



Objetivo:

- Realizar un microsisistema de archivos multi hilo de la Facultad de Ingeniería, empleando mecanismos de sincronización, en el cual se permiten 4 operaciones a un archivo, listar los archivos del disco, copiar un archivo del disco a la máquina local, pasar un archivo de la máquina hacia el disco y eliminar un archivo.

Entornos y dependencias:

Librerías:

```
import os
import struct
from threading import Lock, Semaphore
import tkinter as tk
from tkinter import filedialog, messagebox, simpledialog
```

- El programa fue realizado en Python 3.12.3, usando el IDE Visual Studio Code de Microsoft, usando un Sistema Operativo Windows 11.
- Se debe tener en cuenta que igual se debe instalar la librería tkinter 8.6.13 para la GUI, las demás ya vienen con Python.

Buscando en Internet, para una distribución Debian de Linux, en esta página se muestra como instalar Python para Debian 11 [Instalar Python 3.11.1 en Debian 11 y derivados - weblinus](#) y [python - How to install Tkinter on debian sid? - Stack Overflow](#)

Especificaciones:

- El sistema de archivos cabe en un diskette tradicional de 1440 Kilobytes, este se divide en sectores de 256 bytes que a su vez se va a dividir en clusters (Cada cluster tiene 4 sectores) y todos los enteros serán representados como enteros de 32 bits, no se maneja una tabla de particiones y FiUnamFS es un sistema de archivos plano, terminando el espacio del directorio, tenemos totalmente espacio libre.

Solución:

Para el primer cluster lo manejaremos como un superbloque, este se va a fragmentar de la siguiente forma:

0 - 8: Validación de versión y nombre del directorio

```
Comprobación del sistema de archivos... Nombre: FiUnamFS, Versión: 25-1
```

Nuestro código está ejecutado en Visual Studio Code y podemos ver un mensaje de validación en la terminal, en código, lo que hacemos es tomar el primer elemento del superbloque y desempaquetarlo así como traducir su código ascii para poder realizar la validación pertinente, en nuestro caso igualmente hacemos uso de una GUI para seleccionar directamente el archivo desde una interfaz al ejecutar en donde en el encabezado viene la instrucción de que hacer.

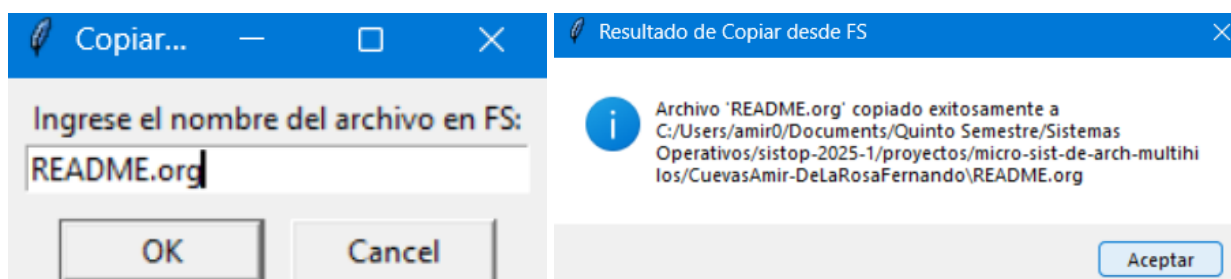
Como se puede observar en la imagen previa, tenemos 3 archivos de diferentes tamaños con su cluster inicial y sus fechas en formato 'AAAAMMDDHHMMSS'.

Ejecución:

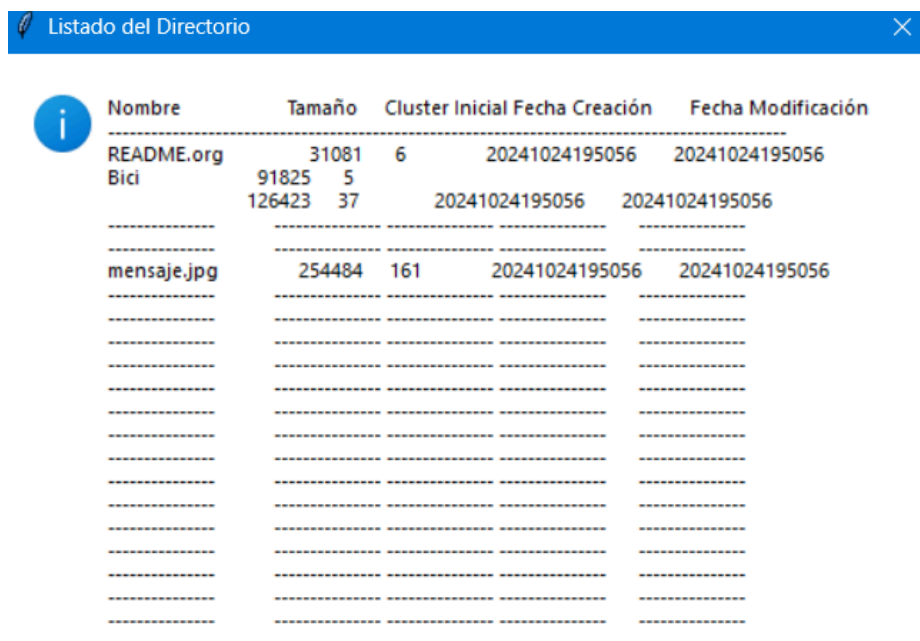
Para poder hacer uso del programa basta con ejecutar el archivo llamado 'FiUnamFS.py' desde la terminal o con ayuda de algun IDE, para Linux, encontramos este enlace [How to execute python file in linux - Stack Overflow](#)

No es necesario hacer modificaciones al código ya que el mismo programa con su GUI hace que el programa puede ser utilizado.


Para copiar un archivo desde FS hacia la máquina local, basta con dar click al botón de la interfaz y colocar el nombre del archivo en el repositorio de FiUnamFS, y después seleccionar la carpeta hacia donde queremos que este se copie.




Para copiar un archivo de manera local hacia FS basta con seleccionarlo y asignarle un nombre para el directorio




Y por último, también podemos eliminar archivos del listado simplemente colocando su nombre en el recuadro que la GUI nos proporciona.



Listado del Directorio



	Nombre	Tamaño	Cluster Inicial	Fecha Creación	Fecha Modificación
	README.org	31081	6	20241024195056	20241024195056
	logo.png	126423	37	20241024195056	20241024195056
	mensaje.jpg	254484	161	20241024195056	20241024195056

Mecanismos de sincronización utilizados:

Hacemos uso de un lock y un semáforo, un lock *self.bloque_acceso* es para sincronizar el acceso al sistema de archivos y es utilizado para las operaciones (Mover, listar y borrar), y el semáforo *self.bloqueo_superbloque* controla el acceso a la sección del superbloque.

Consideración:

El programa fue hecho haciendo uso de Windows 11 y el IDE Visual Studio Code, preferiblemente el codigo de *FiUnamFS.py* y *fiunamfs.img* deben de estar en la misma carpeta, no es necesario que a estos se les hagan modificaciones, basta con directamente compilar el archivo .py