National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Faculty of Computing

CS-330 Operating System

BESE – 14B

OPEN ENDED LAB

USER GUIDE

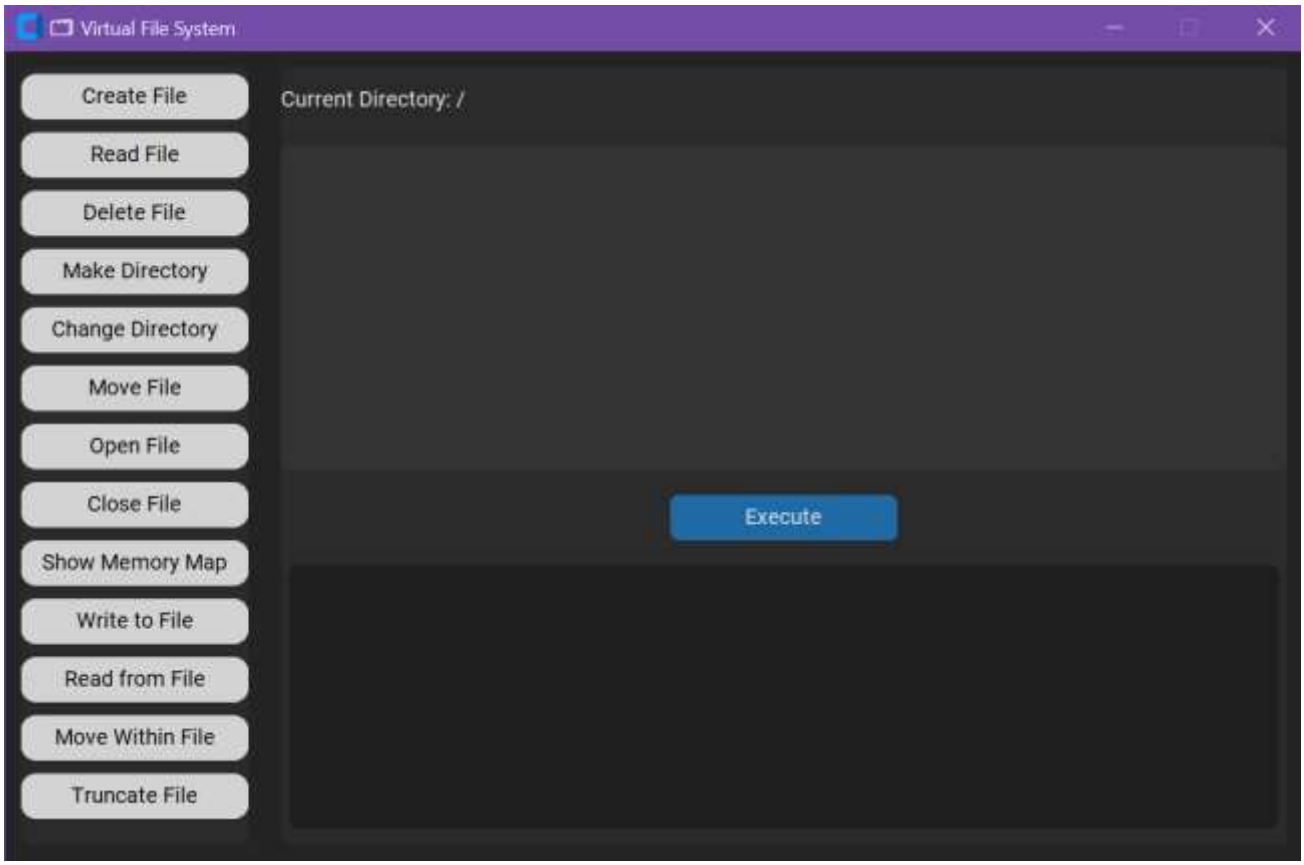
Submission Deadline: 4th May, 2024
Lab Engineer: Mr. Junaid Sajid
Instructor: Engr Taufeeq Ur Rehman

Name	CMS ID
Osama Ayaz	459700
Anoosheh Arshad	459658

Project Link: https://github.com/usama-codes/virtual-file-system

Starting the Application:

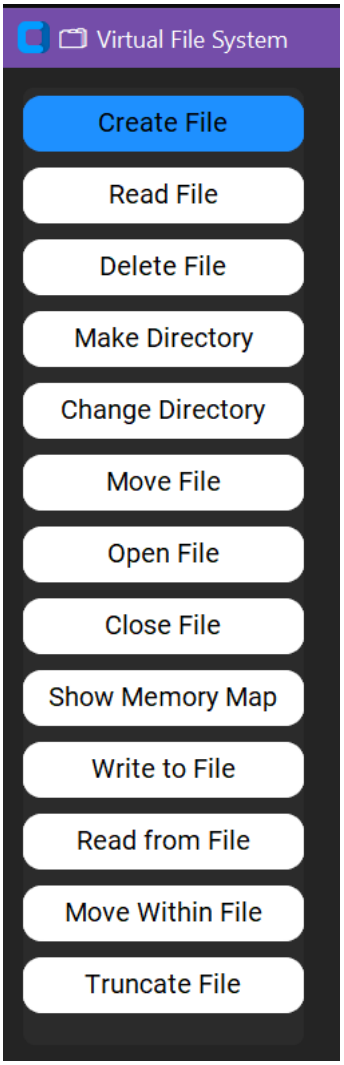
- Before starting, run the SystemInitializer.py to create the cample.dat file (this is our file system!)
- Run the Python file GUI.py



- A modern, dark-mode GUI will open titled **"Virtual File System"**.

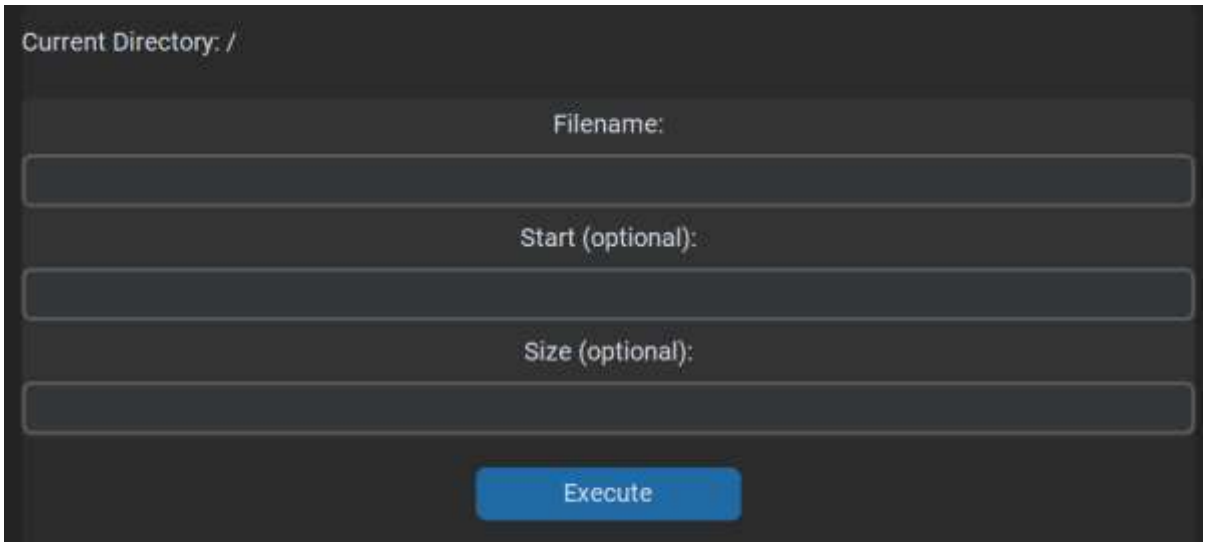
Choosing Operations:

- You will see various **buttons** to do the different tasks of your need
- Select **what you want to do**:
  - **Create File:** Create a new file inside the current directory and put some content inside it.
  - **Read File**: Read and show the content of an existing file (from the curret directory.
  - **Delete File**: Delete an existing file in the current directory (cannot delete a folder with this!).
  - **Make Directory**: Create a new empty directory (folder).
  - **Change Directory:** Move into another directory (or go up to parent/root).
  - **Move File**: Move a file from current directory into another directory.
  - **Open File**: Open a file in memory (to read/write operations).
  - **Close File:** Close the opened file (frees it from memory).
  - **Show Memory Map**: Show the overall structure: which files/directories exist and where.
  - **Write to File**: Add (write) some content to an opened file.
  - **Read from File:** Read a specific part of a file (start, size).
  - **Move Within File**: Move a part of the file's data to another position inside the same file.
  - **Truncate File**: Shorten or extend a file to a specific size.



Entering Fields:

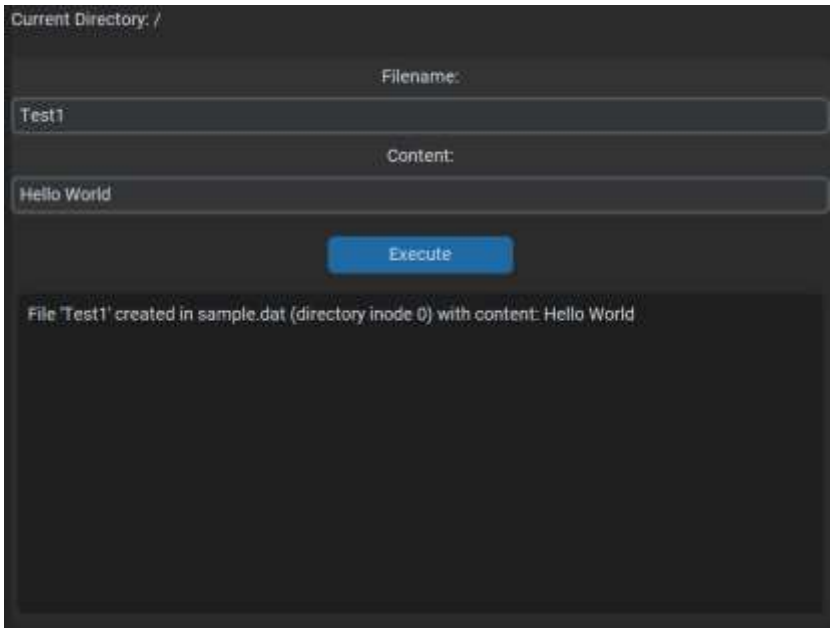
- After selecting an operation, the GUI will show the **required fields** (like **filename**, **start, size** etc.).
- **Fill in the details (**shown for truncate file)**:**



.

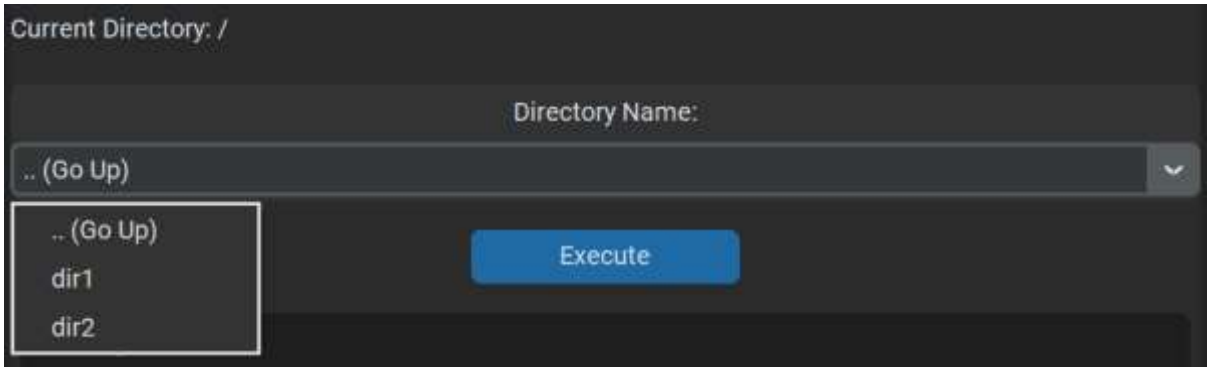
Executing:

- Press the **"Execute"** button.
- Your action will be performed.
- Console outputs (status like "file created", "file moved", "file not found", errors, etc.) will appear in the **bottom output window**.



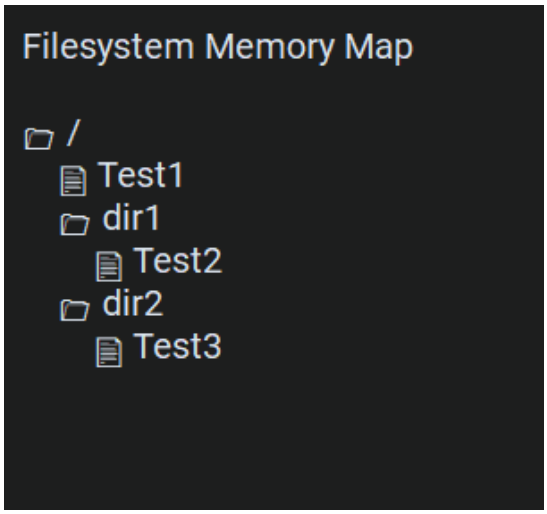
Current Directory:

- Your **Current Directory** is shown right above the dropdown.
- Whenever you **Change Directory**, the label updates.



Special Things to Know:

- **After executing**, your choices in the fields are **automatically cleared** for a new operation.
- **Directories** can only be created or navigated into if they already exist.
- **Only files can be read, written, deleted**, etc.
- **If you want to go back to the root**, in Change Directory, select ".. (Go Up)" option.
- You can **view the filesystem memory** at any point using "Show Memory Map"



Directory Structure:

The directory structure of the Virtual File System (VFS) is hierarchical, utilizing inodes and directory entries to efficiently manage files and directories. Here's an overview:

1. Root Directory

- The root directory is the entry point of the file system, represented by a special inode.
- It contains a list of directory entries, each pointing to either a file or a subdirectory.

2. Directories

- Directories are special files that store lists of entries, where each entry links to a file or another directory.
- Directories are identified by a flag in their metadata, indicating whether they contain files or other directories.

3. Files

- Files are also represented by inodes, similar to directories, but with a flag indicating they are not directories.
- A file’s inode contains metadata and pointers to data blocks where the file’s content is stored.

4. Hierarchical Structure

- The system's hierarchical structure is achieved by linking directories and files through directory entries.
- Each directory can contain both files and subdirectories, enabling a nested file organization.

5. Directory Operations

- Common operations on directories include creating new directories, navigating between directories, and listing the directory contents.
- Navigation is achieved by updating the reference to the current directory, allowing users to move through the file system.