**Operating Systems**

**Interactive File Management System**

**Simulating Core File Operations**

**By:**

**Sai Kiran  
Kaustubh Adhav**

**Harsh Patel**

**Dhruvik Patel**

# Project Report

## 1. Overview

This project implements a file system GUI using Dear ImGui, a graphical interface library. The application enables users to perform various file system operations, including file and directory management, disk usage monitoring, and file modification. The GUI is implemented in C++ and uses OpenGL for rendering.

## 2. Implemented Functions

|  |  |
| --- | --- |
| Operation | Description |
| Create Directory | Creates a new directory with the specified name. |
| Delete Directory | Deletes the specified directory. |
| Create File | Creates a new file with the specified name. |
| Delete File | Deletes the specified file. |
| Write to File | Writes the provided content to the specified file. |
| Read File | Reads and returns the content of the specified file. |
| Get File Info | Retrieves detailed information about the specified file. |
| List Directory Contents | Lists all files and subdirectories within the specified directory. |
| Rename File/Directory | Renames a file or directory from the old name to the new name. |
| Move File/Directory | Moves a file or directory to a specified destination. |
| Copy File | Copies a file to a specified destination. |
| Change File Permissions | Updates the permissions of a specified file. |
| Get Disk Usage | Returns the total, free, and used disk space in the current directory. |

## 3. Libraries and Their Uses

### Standard C++ Libraries

* <iostream>: For console input/output.
* <fstream>: For file handling (reading and writing files).
* <string>: To handle string operations.
* <vector>: To manage dynamic arrays and store collections.
* <filesystem>: To interact with the file system.
* <sys/stat.h> and <sys/types.h>: To manage file and directory attributes.

### Dear ImGui

* Purpose: A graphical user interface library for rapid prototyping and building interactive interfaces.
* Uses: Creating UI elements for file system operations, managing themes and styles, and rendering interfaces.

### GLFW

* Purpose: A library for managing OpenGL contexts, handling windows, and user input.
* Uses: Managing the application window and capturing user interactions.

### OpenGL

* Purpose: For rendering graphics.
* Uses: Handles low-level graphics rendering for the Dear ImGui interface.

### POSIX APIs

* <unistd.h> and <dirent.h>: To interact with the operating system for file and directory operations.

## 4. Detailed File Analysis

### file\_operations.cpp

Main Libraries Used: <filesystem>, <sys/stat.h>

Purpose: Implements core file system operations.

#### Key Functions:

* create\_directory: Creates a new directory.
* delete\_directory: Deletes an existing directory.
* list\_directory\_contents: Lists files and subdirectories within a specified directory.
* create\_file: Creates a new file in a directory.
* delete\_file: Deletes a specified file.
* write\_to\_file: Writes data to a file.
* read\_file: Reads content from a file.
* change\_permissions: Updates file permissions.

### file\_operations.h

Main Libraries Used: None directly used but includes <string> and <vector>.

Purpose: Defines the prototypes for functions implemented in file\_operations.cpp.

### main.cpp

Main Libraries Used: Dear ImGui, GLFW, OpenGL, <string>, <vector>

Purpose: Sets up Dear ImGui context, integrates GUI with file operations, and handles user input.

### Makefile

Main Libraries Mentioned: GLFW, OpenGL, Dear ImGui

Purpose: Automates the build process across different platforms.

### README.md

Purpose: Serves as project documentation providing setup steps and project structure details.

## 5. Steps to Build and Run the Program

1. Install the necessary libraries:

bash  
sudo apt update  
sudo apt install gcc pkg-config g++ build-essential libglfw3-dev libgl1-mesa-dev libx11-dev libxrandr-dev libxi-dev libxxf86vm-dev libxcursor-dev cmake

1. Clone Dear ImGui repository:

bash  
git clone --recursive https://github.com/ocornut/imgui -b docking

NOTE: The `imgui` folder should be at the project root (one directory above FileSys\_GUI).

1. Build the project:

bash  
cd file\_system\_gui  
make

1. Run the GUI interface:

bash  
./file\_Sys\_gui

## 6. Directory Structure for This Project

├── FileSys\_GUI  
├── imgui  
├── Output\_Screenshots  
└── README.md

## 7. Test Screenshots

Below are screenshots of the GUI in operation. Refer to the project folder for full-resolution images.

A screenshot of a computer

Description automatically generated

