CAPSTONE PROJECT REPORT

LINUX FILE MANAGER USING MODERN C++

Name: SWAROOP KUMAR SATAPATHY
Registration No.: 2241019345

1. Introduction

The Linux File Manager is a terminal-based file explorer developed using Modern C++ (C++17) and the <filesystem> library.
It provides basic file management functions such as listing files, navigating directories, creating, deleting, copying, and searching files directly from the terminal.
The purpose of this project is to strengthen understanding of Linux OS concepts and system-level programming using C++.

2. Objectives

- Design and implement a command-line file management system in C++.

- Apply the concepts of file handling and directory traversal using the <filesystem> library.

- Gain practical knowledge of system-level programming.

- Improve understanding of command-line interfaces and user interaction.

3. Project Description

The project accepts simple text commands such as ls, cd, create, delete, copy, and search to perform file operations.
It provides meaningful feedback to users for success or failure of commands.

Commands Implemented

ls → Lists all files and directories

cd <folder> → Change directory

cd .. → Go to parent directory

create <filename> → Create a new file

delete <filename> → Delete a file

copy <source> <destination> → Copy files

search <keyword> → Search for files

exit → Exit the program

4. System Requirements

Software Requirements:

Linux (Ubuntu / Fedora / WSL)

GCC compiler with C++17 support

Code::Blocks / Visual Studio Code

Hardware Requirements:

Intel i3 or higher processor

Minimum 2 GB RAM

100 MB free disk space

5. Implementation Details

The implementation consists of individual functions to perform each file operation.
The program continuously accepts commands until exit is entered.
It uses string stream for command parsing and <filesystem> for file and directory operations.
The program loop interacts with users and displays relevant success or error messages.

6. Output and Testing

The program was tested on Linux systems and produced correct outputs for various file operations.
Error handling ensures the program remains stable even for invalid inputs.
Screenshots of successful execution are attached at the end of this report.

7. Learning Outcomes

Gained practical experience with Linux file systems.

Learned to use the C++17 <filesystem> library.

Improved knowledge of file handling and error management.

Understood terminal-based programming and input handling.

8. Conclusion

The Linux File Manager successfully demonstrates integration of Modern C++ with Linux system programming.
It allows performing core file operations efficiently from the terminal.
The project meets its educational objective by providing hands-on experience with C++ filesystem operations and Linux OS concepts.

9. Future Scope

Add GUI support.

Implement file compression and permission settings.

Extend to multi-user access management.

Include logging and file history tracking features.

10. References
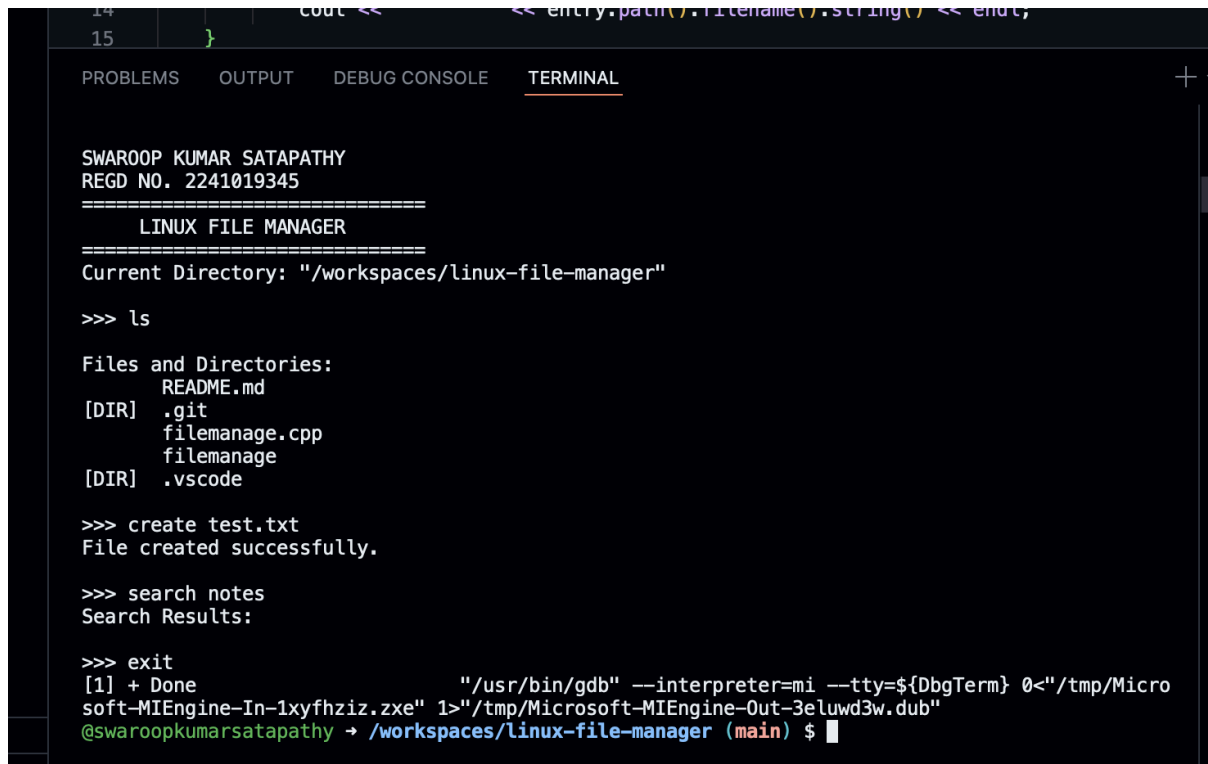
C++ Filesystem Reference

GNU GCC Documentation

Linux Manual Pages (man filesystem, man g++)

W3Schools C++ File Handling Tutorials

Appendix: <u>Screenshots</u>

Name: SWAROOP KUMAR SATAPATHY
Registration No.: 2241019345
Section: 2241024
Wipro Batch: 3

End of Report