

**INSTITUTE OF TECHNICAL EDUCATION
AND RESEARCH**
(SOA Deemed to be University)

PROJECT REPORT
Console-Based File Explorer in C++
(LINUX OS)



Submitted By

Name: SHREYA SINHA

Registration No.: 2241019538

Branch: ECE

Batch: 13

Code:

```
#include <iostream>
#include <filesystem>
#include <fstream>
#include <string>
#include <sys/stat.h>
#include <io.h>
#include <fcntl.h>
#include <unistd.h>

namespace fs = std::filesystem;

// -----
// Function: List all files/folders
// -----
void listFiles(const fs::path& pathToShow) {
    std::cout << "Current Directory: " << pathToShow << "\n";
    std::cout << "-----\n";

    for (const auto& entry : fs::directory_iterator(pathToShow)) {
        std::string name = entry.path().filename().string();
        if (entry.is_directory())
            std::cout << "[DIR] " << name << '\n';
        else
            std::cout << "      " << name << '\n';
    }

    std::cout << "-----\n";
}

// -----
// Function: Create new file
// -----
void createFile(const fs::path& path) {
    std::ofstream file(path);
    if (file)
        std::cout << "File created: " << path.filename().string() << '\n';
    else
        std::cout << "Error creating file.\n";
}

// -----
// Function: Search for file recursively
// -----
void searchFile(const fs::path& root, const std::string& target) {
    std::cout << "Searching for \"" << target << "\"...\n";
    bool found = false;
    for (const auto& entry : fs::recursive_directory_iterator(root)) {
        if (entry.path().filename() == target) {
            std::cout << " Found: " << entry.path().string() << '\n';
            found = true;
        }
    }
}
```

```

    if (!found)
        std::cout << "File not found.\n";
}

// -----
// Function: Show file permissions
// -----
void showPermissions(const fs::path& p) {
    struct stat info;
    if (stat(p.string().c_str(), &info) != 0) {
        std::cout << "Unable to access file.\n";
        return;
    }

    std::cout << "Permissions for " << p.filename().string() << ":" ;
    std::cout << ((info.st_mode & S_IRUSR) ? "r" : "-");
    std::cout << ((info.st_mode & S_IWUSR) ? "w" : "-");
    std::cout << ((info.st_mode & S_IXUSR) ? "x" : "-");
    std::cout << ((info.st_mode & S_IRGRP) ? "r" : "-");
    std::cout << ((info.st_mode & S_IWGRP) ? "w" : "-");
    std::cout << ((info.st_mode & S_IXGRP) ? "x" : "-");
    std::cout << ((info.st_mode & S_IROTH) ? "r" : "-");
    std::cout << ((info.st_mode & S_IWOTH) ? "w" : "-");
    std::cout << ((info.st_mode & S_IXOTH) ? "x" : "-");
    std::cout << '\n';
}

// -----
// Function: Change file permissions
// -----
void changePermissions(const fs::path& file, int mode) {
#ifndef _WIN32
    int result = _chmod(file.string().c_str(), mode);
    if (result == 0)
        std::cout << "Permissions changed (simulated for Windows).\n";
    else
        std::cout << "Failed to change permissions.\n";
#else
    if (chmod(file.string().c_str(), mode) == 0)
        std::cout << "Permissions changed.\n";
    else
        std::cout << "Failed to change permissions.\n";
#endif
}
// -----
// Main Function
// -----
int main() {
    fs::path current = fs::current_path();
    std::string cmd;
    std::cout << "\nWelcome to Console-Based File Explorer\n";
    std::cout << "Type 'help' to see all commands.\n";
}

```

```

while (true) {
    std::cout << "\n> ";
    std::getline(std::cin, cmd);

    if (cmd == "exit") break;

    // --- Navigation ---
    else if (cmd == "back")
        current = current.parent_path();

    else if (cmd.rfind("cd ", 0) == 0) {
        fs::path newDir = current / cmd.substr(3);
        if (fs::is_directory(newDir))
            current = newDir;
        else
            std::cout << "Invalid directory.\n";
    }

    // --- Directory and File Management ---
    else if (cmd.rfind("mkdir ", 0) == 0)
        fs::create_directory(current / cmd.substr(6));

    else if (cmd.rfind("rm ", 0) == 0)
        fs::remove(current / cmd.substr(3));

    else if (cmd.rfind("mv ", 0) == 0) {
        std::string rest = cmd.substr(3);
        auto pos = rest.find(' ');
        if (pos == std::string::npos) {
            std::cout << "Usage: mv <source> <destination>\n";
            continue;
        }
        fs::rename(current / rest.substr(0, pos), current / rest.substr(pos + 1));
    }

    else if (cmd.rfind("cp ", 0) == 0) {
        std::string rest = cmd.substr(3);
        auto pos = rest.find(' ');
        if (pos == std::string::npos) {
            std::cout << "Usage: cp <source> <destination>\n";
            continue;
        }
        fs::copy(current / rest.substr(0, pos), current / rest.substr(pos + 1),
                 fs::copy_options::overwrite_existing);
    }

    else if (cmd.rfind("new ", 0) == 0)
        createFile(current / cmd.substr(4));

    // --- Listing & Search ---
    else if (cmd == "ls" || cmd == "list")
        listFiles(current);
}

```

```

else if (cmd.rfind("search ", 0) == 0)
    searchFile(current, cmd.substr(7));

// --- Permissions ---
else if (cmd.rfind("chmod ", 0) == 0) {
    std::string rest = cmd.substr(6);
    auto pos = rest.find(' ');
    if (pos == std::string::npos) {
        std::cout << "Usage: chmod <octal> <file>\n";
        continue;
    }
    int mode = std::stoi(rest.substr(0, pos), nullptr, 8);
    fs::path file = current / rest.substr(pos + 1);
    changePermissions(file, mode);
}

else if (cmd.rfind("perm ", 0) == 0)
    showPermissions(current / cmd.substr(5));

// --- Help ---
else if (cmd == "help") {
    std::cout << "\n Available Commands:\n";
    std::cout << "-----\n";
    std::cout << "ls or list           - Show current directory
contents\n";
    std::cout << "cd <folder>          - Enter folder\n";
    std::cout << "back                  - Go up one level\n";
    std::cout << "mkdir <folder>        - Create new folder\n";
    std::cout << "new <file>            - Create new file\n";
    std::cout << "rm <file>              - Delete file\n";
    std::cout << "mv <a> <b>            - Rename or move file\n";
    std::cout << "cp <a> <b>            - Copy file\n";
    std::cout << "search <filename>      - Search file recursively\n";
    std::cout << "perm <file>            - View file permissions\n";
    std::cout << "chmod <mode> <file>    - Change permissions (simulated on
Windows)\n";
    std::cout << "exit                  - Quit program\n";
    std::cout << "-----\n";
}
else
    std::cout << "Unknown command. Type 'help' for options.\n";
}
std::cout << "Exiting File Explorer. Goodbye!\n";
return 0;
}

```

Day 1: Design the application structure and setup the development environment. Start with basic file operations like listing files in a directory.

Output:

```
M /c/Users/win10/Desktop/Project/new_project

win10@SHREYA MINGW64 ~
$ cd /c

win10@SHREYA MINGW64 /c
$ cd Users

win10@SHREYA MINGW64 /c/Users
$ cd win10

win10@SHREYA MINGW64 /c/Users/win10
$ cd Desktop

win10@SHREYA MINGW64 /c/Users/win10/Desktop
$ cd Project

win10@SHREYA MINGW64 /c/Users/win10/Desktop/Project
$ cd new_project

win10@SHREYA MINGW64 /c/Users/win10/Desktop/Project/new_project
$ g++ -std=c++17 file_explorer.cpp -o explorer

win10@SHREYA MINGW64 /c/Users/win10/Desktop/Project/new_project
$ ./explorer

Welcome to Console-Based File Explorer
Type 'help' to see all commands.

> help

Available Commands:
-----
ls or list           - Show current directory contents
cd <folder>         - Enter folder
back                 - Go up one level
mkdir <folder>       - Create new folder
new <file>          - Create new file
rm <file>           - Delete file
mv <a> <b>          - Rename or move file
cp <a> <b>          - Copy file
search <filename>    - Search file recursively
perm <file>          - View file permissions
chmod <mode> <file>   - Change permissions (simulated on Windows)
exit                 - Quit program
-----

> list
Current Directory: "C:\\\\Users\\\\win10\\\\Desktop\\\\Project\\\\new_project"
-----
explorer.exe
file_explorer.cpp
-----
```

Day 2: Implement file and directory navigation features. Enable the user to move through directories

Output:

```
> ls
Current Directory: "C:\\Users\\win10\\Desktop\\Project\\new_project"
-----
    explorer.exe
    file_explorer.cpp
-----

> back

> ls
Current Directory: "C:\\Users\\win10\\Desktop\\Project"
-----
[DIR]  important
[DIR]  new_project
-----

> cd new_project

> ls
Current Directory: "C:\\Users\\win10\\Desktop\\Project\\new_project"
-----
    explorer.exe
    file_explorer.cpp
-----

> exit
Exiting File Explorer. Goodbye!

win10@SHREYA MINGW64 /c/Users/win10/Desktop/Project/new_project
$ |
```

Day 3: Add file manipulation capabilities (copy, move, delete, create).

Output:

```
win10@SHREYA MINGW64 /c/Users/win10/Desktop/Project/new_project
$ g++ -std=c++17 file_explorer.cpp -o explorer

win10@SHREYA MINGW64 /c/Users/win10/Desktop/Project/new_project
$ ./explorer

Welcome to Console-Based File Explorer
Type 'help' to see all commands.

> ls
Current Directory: "C:\\\\Users\\\\win10\\\\Desktop\\\\Project\\\\new_project"
-----
    explorer.exe
    file_explorer.cpp
-----

> mkdir new1.txt

> ls
Current Directory: "C:\\\\Users\\\\win10\\\\Desktop\\\\Project\\\\new_project"
-----
    explorer.exe
    file_explorer.cpp
[DIR]  new1.txt
-----

> rm new1.txt

> ls
Current Directory: "C:\\\\Users\\\\win10\\\\Desktop\\\\Project\\\\new_project"
-----
    explorer.exe
    file_explorer.cpp
-----
```

Day 4: Implement file search functionality within the file explorer

Output:

```
> ls
Current Directory: "C:\\\\Users\\\\win10\\\\Desktop\\\\Project\\\\new_project"
-----
explorer.exe
file_explorer.cpp
hello.py
-----
> cp hello.py hello1.py
> ls
Current Directory: "C:\\\\Users\\\\win10\\\\Desktop\\\\Project\\\\new_project"
-----
explorer.exe
file_explorer.cpp
hello.py
hello1.py
-----
> mv hello1.py hello2.py
> ls
Current Directory: "C:\\\\Users\\\\win10\\\\Desktop\\\\Project\\\\new_project"
-----
explorer.exe
file_explorer.cpp
hello.py
hello2.py
-----
> search hello2.py
Searching for "hello2.py"...
Found: C:\\Users\\win10\\Desktop\\Project\\new_project\\hello2.py
> search hello5.py
Searching for "hello5.py"...
File not found.

>
```

Day 5: Add file permission management features

Output:

```
mv <a> <b>           - Rename or move file
cp <a> <b>           - Copy file
search <filename>     - Search file recursively
perm <file>          - View file permissions
chmod <mode> <file>   - Change permissions (simulated on Windows)
exit                  - Quit program
-----
> new a.txt
File created: a.txt

> ls
Current Directory: "C:\\\\Users\\\\win10\\\\Desktop\\\\Project\\\\new_project"
-----
a.txt
explorer.exe
file_explorer.cpp
hello.py
hello2.py
-----
> perm a.txt
Permissions for a.txt: rw-rw-rw-
> chmod 755 a.txt
Permissions changed (simulated for Windows).
> perm a1.txt
Unable to access file.
> perm a.txt
Permissions for a.txt: rw-rw-rw-
> exit
Exiting File Explorer. Goodbye!
win10@SHREYA MINGW64 /c/Users/win10/Desktop/Project/new_project
$ |
```