Mini-version File Manager Development Project

Team size: 1 to 2 people

Q&A contact: Chen Jingwen 12532511@mail.sustech.edu.cn

Project Overview

In the digital age, the file system serves as the core bridge for interaction between the operating system and users. Whether it is managing daily office documents or organizing code files during the development process, efficient file operation tools are relied upon. This project requires the development of a terminal-based mini file manager (MiniFileExplorer) that supports core operations such as file browsing, creation, deletion, and searching similar to these in Linux.

Project Requirements and Implementation Details

Basic Requirements (80 points)

Directory Initialization and Switching (25 points)

- **Startup Initialization (10 points)**

1. When the program starts, it defaults to loading the current working directory (obtained through the getcwd() function) and displays the current directory path in the terminal (format: Current Directory: /home/user/project).

2. It supports specifying the initial directory through command-line parameters, such as ./MiniFileExplorer /home/user/docs. If the specified directory does not exist, it prompts "Directory not found: /home/user/docs" and exits.

- **Directory Switching Operation (15 points)**

1．It supports entering "cd [directory path]" to switch directories (such as cd ..) /.. Switch to the parent directory of the parent directory (cd ..\..), and switch to the test subdirectory under the current directory (cd ./test)).

2．Verify the legality of the directory when switching: If the target directory does not exist, prompt "Invalid directory: [target path]"; if the target path is a file rather than a directory, prompt "Not a directory: [target path]".

3．Support inputting "cd ~" to quickly switch to the current user's home directory (in Linux it is /home/[username]).

File/Folder Basic Operations (30 points)

- **File/Folder Creation (10 points)**

1．Input "touch [filename]" to create an empty file (such as "touch note.txt"), if the file already exists, prompt "File already exists: note.txt".

2．Input "mkdir [folder name]" to create an empty folder (such as "mkdir data"), if the folder already exists, prompt "Directory already exists: data".

- **File/Folder Deletion (10 points)**

1．Input "rm [filename]" to delete the specified file, and a second confirmation is required before deletion (prompt "Are you sure to delete [filename]? (y/n)"), only input "y" to execute the deletion, other inputs cancel the operation.

2．Input "rmdir [folder name]" to delete the specified empty folder, if the folder is not empty, prompt "Directory not empty: [folder name]"; if the folder does not exist, prompt "Directory not found: [folder name]".

- **Directory Content List Display (10 points)**

1．When inputting "ls", display all files and folders in the current directory in a list format, distinguish the display types (folder names followed by /, such as data/; file names are displayed normally, such as note.txt).

2．The list should include 4 columns: "Name, Type, Size (bytes), Modification Time", and the format should be aligned (example as follows):

```
Name            Type      Size(B)      Modify Time
```

```
data/          Dir     -           2024-05-20 14:30:00
note.txt       File    1024        2024-05-20 15:10:23
```

Terminal Interaction and Information Query (25 points)

Command Interaction and Help (10 points)

1. After the program starts, a command prompt is displayed: Enter command (type 'help' for all commands): , allowing continuous command input.

2. When 'help' is entered, all supported commands and their descriptions are listed (such as 'cd [path]: Switch to target directory', 'ls: List all files and directories').

3. When 'exit' is entered, the program exits and prompts "MiniFileExplorer closed successfully".

File Information Query (15 points)

1. When 'stat [file name/folder name]' is entered, detailed information of the target is displayed: type (file/folder), path, size (file in bytes, folder as "-"), creation time, modification time, access time.

2. If the target does not exist, it prompts "Target not found: [target name]"; if no target is specified when inputting (only 'stat' is entered), it prompts "Missing target: Please enter 'stat [name]'".

Advanced Requirements (up to 40 points)

File Search Function (10 points)

1. When 'search [keyword]' is entered, search for files and folders whose names contain the keyword in the current directory and all its subdirectories (case-insensitive).

2. Search results are displayed in "path, type" format, such as:

```
Search results for 'note' (2 items):
/home/user/docs/note.txt (File)
/home/user/data/old_note/ (Dir)
```

3. If no matching result is found, display "No results found for '[keyword]'".

- **File copying and moving (15 points)**

1. File copying: Enter "cp [source file path] [target path]", to copy the source file to the target path (e.g., cp ./note.txt ../backup/). If a file with the same name already exists in the target path, prompt "File exists in target: Overwrite? (y/n)", and input "y" to overwrite.

2. File moving: Enter "mv [source file/folder path] [target path]", to move the source file/folder to the new path (e.g., mv ./data ../archive/). Supports renaming (e.g., mv note.txt new_note.txt, renames the current directory's note.txt to new_note.txt).

3. Verify legality during operation: If the source or target path does not exist, prompt "Source not found"; if the target path is illegal, prompt "Invalid target path".

- **Directory size calculation and sorting (15 points)**

1. Directory size calculation: Enter "du [folder name]", to calculate the total size of this folder and all its sub-files (unit: KB/MB, automatically adapted, e.g., total size of 1024 bytes displays as "1 KB", total size of 2048000 bytes displays as "2 MB"), and prompt "Total size of [folder name]: 5 MB".

2. List sorting: Enter "ls -s", to sort the list in descending order of file/folder size (folders are calculated based on the total size of sub-files, and empty folders are placed at the end); enter "ls -t", to sort by modification time in descending order (the most recently modified file is placed at the top), and the columns of "name, type, size, modification time" remain aligned after sorting.

Scoring Criteria

Classification	Scoring Criteria	Points
Basic Requirements (80 points)	Directory initialization and switching (directory loading upon startup, support for command line parameters, validation of cd command legality)	25

	File/Folder Basic Operations (touch/mkdir creation, rm/rmdir deletion, ls listing display)	30
	Terminal Interaction and Information Query (help/exit commands, detailed information query with stat, friendliness of error messages)	25
Advanced Requirements (Up to 40 points)	File Search (keyword matching, subdirectory traversal, result display)	10
	File Copy and Move (implementation of cp/mv commands, confirmation of overwrite, path legality validation)	15
	Directory Size Calculation and Sorting (size statistics with du command, sorting with ls -s/-t, unit adaptation)	15
Document Requirement	Clarity of design concept, complete function description, logical problem-solving approach	Extra 5 points

Submission Requirements

- **Source Code:** All source files related to the project
- **Project Documentation:** A concise and clear document that includes the purpose and usage of the functions you have implemented.