COSC 439: Operating Systems Project

Title: Basic Operating System Shell Development

Objective: The objective is to design and develop a basic operating system shell capable of executing commands and performing fundamental system operations effectively. It's imperative to stress that the implementation of each command must be directly integrated into the shell program. Using terminal middleware to execute commands, such as popen, ShellExecute(), CreateProcess(), and similar APIs, is strictly prohibited. All functionalities should be seamlessly integrated and executed within the shell itself, ensuring a comprehensive understanding and control of system operations. This shell must be implemented using C++.

Features to Implement:

1. **Change Directory (cd):** Implement functionality to switch the default directory to the specified path. If no directory argument is provided, display the current directory. Report errors for non-existing directories and update the PWD environment variable accordingly.
2. **Clear Screen (clr):** Develop a command to clear the display screen.
3. **List Directory Contents (dir):** Create a command to list the contents of a specified directory.
4. **Environment Variables (environ):** Implement a command to display all environment strings.
5. **Echo Command (echo):** Develop a command to display a comment on the screen, followed by a new line. Reduce multiple spaces/tabs to a single space.
6. **Help Manual (help):** Display the user manual using the more filter.
7. **Pause Operation (pause):** Implement a command to pause shell operations until 'Enter' is pressed.
8. **Quit Command (quit):** Develop a command to exit the shell.
9. **File Permissions (chmod):** Develop functionality to modify file permissions for users, groups, and others.
10. **Change Ownership (chown):** Implement a command to change the ownership of a file or directory, allowing a user to specify the new owner and/or group.
11. **List Files (ls)**: Display a list of files and directories in the current directory.
12. **Print Working Directory (pwd)**: Print the current working directory to the standard output.
13. **Concatenate and Display Files (cat)**: Read files sequentially and output their contents to the standard output.
14. **Create Directories (mkdir)**: Create one or more directories specified by the user.
15. **Remove Directories (rmdir)**: Remove one or more directories specified by the user.
16. **Remove Files (rm)**: Delete specified files or directories.
17. **Copy Files (cp)**: Copy files or directories from one location to another.
18. **Move Files (mv)**: Move files or directories from one location to another.
19. **Create Empty Files (touch)**: Create empty files or update access/modification times of existing files.
20. **Search Text Patterns (grep)**: Search for specified patterns in files or input streams and display matching lines.
21. **Word Count (wc)**: Count the number of lines, words, and characters in a file or input stream.

Requirements:

1. **Progress Report:** Submit a progress report outlining encountered challenges, how you have solved them, the current status, and forthcoming steps. Upon submission, feedback will be given for project adjustment based on the provided feedback. **(1 pt)**

2. **Code Implementation:** Develop a working shell integrating the outlined features **(6 pts)**.
3. **Technical Report:** Prepare a detailed report that includes **(5 pts)**:
   - **Introduction to the Project:** Define objectives, importance, and the scope of the system.
   - **Features Description:** Detail each command's implementation and its significance in executing system operations efficiently.
   - **Implementation Details:** Discuss technical insights, challenges faced, and significant decisions made during the development. List the commands supported and their syntax.
   - **Results Analysis:** Showcase outcomes from each feature and their execution output.
   - **Conclusion:** Summarize key findings.
4. **Presentation:** In person presentation that focuses on the technical aspects of the project. Utilize PowerPoint slides to highlight project goals, algorithms employed, implementation details, evaluations, challenges encountered, and insights gained. Additionally, ensure the presentation includes a live demonstration of the project to provide a practical illustration of its functionality. **(5 pts)**
5. **Retrospective and Contribution Report:** Reflect on the Operating Systems (OS) course, summarizing significant lessons learned, their practical relevance, and their impact on understanding OS principles. Additionally, list your own contributions as well as those of your teammates towards the project. **(1 pt)**.

Deadlines:

- **Progress Report:** November 17, 2025
- **Presentation and Demo:** December 3–8, 2025 (In person)
  - Presentations will take place in the professor’s office (YR 456) or in the library (YR 454).
  - Each group will present together, and all group members must be present.
  - The professor may ask questions or request modifications to the project or source code to verify that the work was done by the students and not generated by GenAI or copied from online sources.
  - Time slots will be provided via Calendly, and students should book a slot according to their convenience.
- **Source Code Submission:** December 8, 2025
- **Technical Report:** December 11, 2025
- **Retrospective and Contribution Report:** December 11, 2025