CSE 344

Homework #1

Due Date: 23 March 23:59

- **This assignment will be tested and run on Debian 11 (64-bit) in VirtualBox.**
- **Identical or highly similar submissions will receive -100 points.**
- **A report with screenshots is mandatory. No report = 0 points.**
- **Late submissions will not be accepted.**

Scenario: You will develop a Secure File and Directory Management System in C programming language using Linux system calls. The program must handle file and directory operations, use process creation (fork()), and support logging.

Requirements

Your program should support the following operations:

1. **Create Files and Directories (15 points)**

The user should be able to create a directory and create a file with the following commands:

createDir "folderName"

- If the directory already exists, the program should print:

Error: Directory "folderName" already exists.

createFile "fileName"

- If the file already exists, the program should print:

Error: File "fileName" already exists.

- If the file is successfully created, it should store the creation timestamp inside the file.

2. **List Files and Directories (15 points)**

The user should be able to list all files and directories using:

listDir "folderName"

- If the directory does not exist, the program should print:

Error: Directory "folderName" not found.

To list files by extension, use:

listFilesByExtension "folderName" ".txt"

- If no files match the extension, print:

No files with extension ".txt" found in "folderName".

Listing operations should be handled using a separate process (fork()).

3. Read and Update File Content (15 points)

To read a file's content:

readFile "fileName"

- If the file does not exist, print:

Error: File "fileName" not found.

To append content to a file:

appendToFile "fileName" "new content"

- This operation must use file locking to prevent simultaneous writes.
- If the file is read-only or locked, print:

Error: Cannot write to "fileName". File is locked or read-only.

4. Delete Files and Directories (15 points)

To delete a file:

deleteFile "fileName"

- If the file does not exist, print:

Error: File "fileName" not found.

- If deletion is successful, print:

File "fileName" deleted successfully.

To delete an empty directory:

deleteDir "folderName"

- If the directory is not empty, print:

Error: Directory "folderName" is not empty.

File and directory deletion should be performed using a separate process (fork()).

5. **Logging and Report Generation (15 points)**

- Every operation should be logged in log.txt. Example log entries:

[2025-03-10 14:35:21] File "test.txt" created successfully.
[2025-03-10 14:36:45] Directory "backup" deleted.

- The user should be able to display logs with:

showLogs

6. **Help and Usage Guide (10 pointss)**

When the program is run without arguments, it should display all available commands and descriptions:
fileManager

Example output:

Usage: fileManager <command> [arguments]

Commands:
createDir "folderName" - Create a new directory
createFile "fileName" - Create a new file
listDir "folderName" - List all files in a directory
listFilesByExtension "folderName" ".txt" - List files with specific extension
readFile "fileName" - Read a file's content
appendToFile "fileName" "new content" - Append content to a file
deleteFile "fileName" - Delete a file
deleteDir "folderName" - Delete an empty directory
showLogs - Display operation logs

7. Testing Scenario

Students should test their programs using this step-by-step scenario:

1. Create a directory named testDir.
2. Create a file example.txt and write "Hello, World!" inside it.
3. List all files in testDir.
4. Read example.txt, then append "New Line" to it.
5. Delete example.txt and display the logs.

Grading (Total 100 Points)

Category	Description	Points
1. Create Files and Directories	createDir and createFile commands work correctly	15
2. List Files and Directories	listDir and listFilesByExtension work correctly	15
3. Read and Update File Content	readFile and appendToFile work correctly (with file locking)	15
4. Delete Files and Directories	deleteFile and deleteDir work correctly	15
5. Logging	log.txt records all operations and showLogs works correctly	15
6. Help and Usage Guide	fileManager lists all commands	10
7. Code Quality & Compilation	Code is well-structured, uses makefile, and make clean works	10
8. Report *(Mandatory)*	Screenshots and step-by-step documentation	20

Additional Rules:

- Code does not compile → -100 points
- Missing Makefile or make clean → -30 points
- No report (or missing screenshots) → -100 points
- No fork() usage → -50 points
- Missing or broken features → -15 points per feature
- Late submissions → Not accepted

Report Format

Your report must include:

1. Title Page (Student Name, ID, Course, Homework #)

2. Introduction (Explain what the program does)
3. Code Explanation (Explain each function and how it works)
4. Screenshots (For each command tested)
5. Conclusion (Challenges faced, solutions, and final thoughts)

For your questions, please contact: **zbilici@gtu.edu.tr**