National University of Computer and Emerging Sciences, Lahore Campus

	Course:	**Operating Systems**	**Course Code:**	**CS 2006**
	Program:	**BSE (5A)**	**Semester:**	**Fall 2024**
	Due Date	**15 September, 2024 11:59 PM**	**Total Marks:**	**100 marks**
	Type:	**Assignment 1**	**Page(s):**	**3**

Important Instructions:

1. Submit the solution in a folder as your roll number.
2. **You are not allowed to copy solutions from other students. We will check your code for plagiarism using plagiarism checkers. If any sort of cheating is found, heavy penalties will be given to all students involved.**
3. Late submission of your solution is not allowed.

Question 1: Custom Shell **[50 marks]**

You have recently been brought on as a software developer at a startup that is building an innovative educational platform for teaching programming skills. One of the key features requested by the product team is an interactive learning environment where students can practice using command-line tools in a controlled, guided setting. To implement this, the team needs a program that can interpret user commands, simulate a typical terminal environment, and provide immediate feedback.
Your role is to develop a command execution program in C that will serve as the core of this learning tool. The program should:

1. Accept user input where they may type commands like cp ./OS ../newOS. This input should be captured and stored properly, such as in a character array or string object.
2. The program will perform tokenization of the input and separate the command and its arguments.
3. It should then create a child process to execute the given instruction, using system calls such as execvp or execlp.
4. Your program must support commands like ls, cp, cd, pwd, mv, mkdir, rmdir, rm, touch, and grep. (You can find more details on these commands https://www.hostinger.com/tutorials/linux-commands .)
5. After executing a command, it should prompt the user to enter another instruction, allowing for continuous interaction.
6. The program should terminate when the user types "exit."

Question 2: Automated Server Deployment with Process Coordination and Error Handling [25 marks]

Your team is developing an automated server deployment tool that needs to execute multiple stages to set up a new server environment. The stages include System Update, Software Installation, Configuration Setup, and Security Checks.

Your task is to create a C program that:

1. **Stage Execution:** Launches a separate process for each deployment stage in sequence. Each stage should simulate its activity using sleep.
2. **Process Coordination:** Waits for each stage to finish before moving on to the next one (**Hint**: Use waitpid() to synchronize and wait for each child process to complete). If a stage encounters an issue (simulated by an error code), logs the error and retries up to three times.
3. **Error Handling:** If a stage fails after three retries, logs the failure and terminates the deployment process.
4. **Completion Report:** Provides a report on the completion status, detailing which stages were successful and which failed, including the number of retries.

Requirements:

- Employ process management techniques to handle creation, synchronization, and error handling. **(Hint**: Use system calls such as fork(), waitpid(), and exit() appropriately to manage the processes.)
- Simulate a failure with a 15% probability for each stage.

Question 3: Secure Data Handling in a Cybersecurity Context [25 marks]

As a cybersecurity analyst at a data protection firm, you are tasked with securely transforming a critical file, classified_data.txt, to ensure it is safe for further analysis in an ongoing investigation. The file has potentially compromised information that needs to be either decrypted or redacted to maintain data confidentiality.

Your Task:

- Develop a C/C++ program that securely handles data transformation using a parent-child process interaction model.

Details:

1. **Data Transformation Options:**
    - **Decrypt Data (Option 1):** Perform a reverse Caesar cipher, shifting each character backward by one position (if a decryption key is available).
    - **Redact Sensitive Information (Option 2):** Replace sensitive data patterns, like email addresses or credit card numbers, with "[REDACTED]".
2. **Program Flow:**
    - Prompt the user for input and output file names.
    - Ask the user to choose a transformation method (Decrypt or Redact).
    - Create a child process to securely perform the selected transformation.
    - Ensure sensitive data remains secure during the operation.
    - Have the parent process handle writing to the file and generate a summary report.

Requirements:

- Handle invalid file names and incorrect transformation choices.
- Display appropriate error messages for any encountered issues.

Sample Scenarios:

- **Decrypt Data**: If classified_data.txt contains “Uijt jt b dpogjefoujbm nfttbhf” and a shift of 1 is applied, the output should be “This is a confidential message” in output.txt.
- **Redact Data:** If classified_data.txt contains “Credit Card: 4111-1111-1111-1111”, replace the credit card with “[REDACTED]” in output.txt.

Ensure secure and efficient data handling to maintain data integrity and confidentiality during transformation.