National University of Computer and Emerging Sciences, Lahore Campus

	Course:	**Operating Systems**	**Course Code:**	**CS 2006**
	Program:	**BSCS- 4E**	**Semester:**	**Spring 2025**
	Due Date	**4 May, 2025 11:59 PM**	**Total Marks:**	**25 marks**
	Instructor	**Mubashar Hussain**	**TA**	**L226598@lhr.nu.edu.pk**
	Type:	**Assignment 3**	**Questions:**	**2**

Important Instructions:

1. Submit separate code files for each question with your roll number (e.g., Q1_23L1011), along with screenshots of your terminal.
2. **Heavy penalties will be given to all students involved in plagiarism.**
3. Late submission of your solution is not allowed.
4. Your codes will be evaluated thoroughly so make sure you understand all the concepts well.
5. Viva maybe conducted hence it is advised to come prepared
6. In case of any queries, you can always reach out to me via email.

Note: Proper error handling and memory allocation/deallocation carries marks throughout the assignment

---

Question 1: Library Management System using Semaphores and Shared Memory 📚

[15 Marks]

Building upon the system designed in Assignment 2, you are now required to implement the same functionalities using **Shared Memory** and **Semaphores**.

Problem Statement:

You are tasked to create a **server-client model** where the **server** manages the book inventory, and **clients** can **borrow** or **return** books. All operations must be synchronized properly using **semaphores** to avoid race conditions.

Requirements:

- Use **shared memory** to allow access to book data between clients and server.
- Use **semaphores** to manage concurrent access to shared memory.
- Clients will **interact with the server one at a time**:

    - **Clients must wait** (block) using semaphores until the server is ready to serve them.
    - Multiple clients can be started, but only one may be served at a time. Others wait their turn.
- Server will be **persistent.**
- **This should be implemented with 3 semaphores only.**

Server Behavior:

1. Display a welcome message:

```
----------Welcome to Justice Gull Muhammad Library----------
Enter Roll No:  XXLXXXX
```

2. Prompt the user to enter their **Roll Number** in the format (XXLXXXX) where X = 0–9.
    - If the format is incorrect, prompt again until valid input is given.
3. Next, ask the client for their request: borrow or return.

```
Available Actions:
1. Borrow a book
2. Return a book
Enter choice:
```

1. **Borrowing Books:**

Server displays the books currently **available (copies > 0)**:

```
---------- Books Available ----------
[1] Harry Potter
[2] The Hobbit
[3] Calculus 7th Edition
[4] Computer Networks

Enter Book Name:
Enter Number of Copies:
```

- Client is prompted to enter:
    - Book Name (case-sensitive)
    - Number of Copies to borrow

- Server then:
    - Validates the book name (notify client if it doesn't exist)

- Checks availability
- Updates books.txt if books are available:
  - If enough copies exist: confirm borrow and notify client
  - If partial copies available: fulfill partially and notify client
  - If 0 copies: notify client
- If the book doesn't exist in the inventory, give an error.

1. **Returning Books:**

```
---------- Return Book ----------
Enter Book Name:
Enter Number of Copies:
```

- Client is prompted to enter:
  - Book Name (case-sensitive)
  - Number of Copies to return
- Server then:
  - Validates if the book exists
  - Updates the quantity in books.txt and notify client
  - If the book doesn't exist in the inventory, give an error.

Books.txt Format:

BookName/Quantity

```
Harry Potter/10
The Hobbit/7
Calculus 7th Edition/15
Computer Networks/12
Clean Code/0
```

Book names are case-sensitive

NOTE:

- Use **system calls only** for file manipulation: open, read, write, close, etc.
  - Do NOT use ifstream, ofstream, fscanf, fgets, etc.
- Synchronization with **semaphores** is mandatory.
- Proper validation of inputs (roll number, book names, quantities) carries marks.
- Error handling and cleanup/deallocation carries marks.

Question 2: Memory Map Story Correction 🎨 [10 Marks]

Objective:

You are tasked with designing a C/C++ program that corrects a story file that has undergone a color name replacement prank. In the original story, various color names have been replaced with others. For example, "red" has been replaced with "tan", "white" with "olive", and "green" with "black." Your goal is to rectify these substitutions by leveraging **memory mapping** and **multi-threading**.

Problem Statement:

Your program should:

1. Take a **file name** as a **command-line argument**.
2. The file contains a story with color names that have been replaced as follows:
    - "tan" should be replaced with "red".
    - "olive" should be replaced with "white".
    - "black" should be replaced with "green".

3. Use **memory mapping** (mmap()) to load the story into memory efficiently.
4. Implement **two threads**:
    - **Thread 1** processes the **first half** of the story.
    - **Thread 2** processes the **second half** of the story.

5. Each thread should:
    - Replace occurrences of the replaced color names with their correct counterparts.
    - Make modifications **directly in place** within the memory-mapped file.

6. After both threads finish their tasks:
    - The **main thread** should print the updated story to verify the corrections.

NOTE:

- **Only use the open() system call** for file handling.
- You must use **mmap()** to memory-map the file.
- The modifications should be **done in place**, meaning no copies of the data should be made.
- **Two threads** must be used for parallel processing, utilizing **pthread**.
- Do not use any synchronization mechanisms (e.g., semaphores, mutexes) since each thread works on a **separate half** of the file.
- Ensure that modifications made by the threads are appropriately synchronized to the memory map, reflecting the changes in the original file.
- Error handling and cleanup/deallocation carries marks.

Example:

Before: Once upon a time, there was a tan apple and a black tree. The olive sky shone above.

After: Once upon a time, there was a red apple and a green tree. The white sky shone above