National University of Computer and Emerging Sciences
Lahore Campus

Operating System Lab (CL2006)

Date: October 14, 2024

Lab Instructor(s)

Ms. Saba Tariq

Mr. Hassan Raza

Sessional-I Exam

Total Time (Hrs): 2:30
Total Marks: 55
Total Questions: 2

_____ _____   _____
Roll No   Section   Student Signature

Instructions:
- Follow file naming scheme discussed by instructors in Lab.
- Any type of Plagiarism, either from online resources or previous labs will result in **ZERO** marks.
- Late submission will cause negative marking.
- **Print output of each underlined instruction**.

Do not write below this line

Attempt all the questions.

Q1: Create three processes: **Create_pipe.c, Librarian_process.c** and **student_process.c.**   [Total= 30 marks]

Instructions:
1. **Create_pipe.c:**   [5marks]
    The parent process creates two named pipes: **request_pipe** for communication from student_process to librarian_process and **response_pipe** for communication in the reverse direction.
2. **Book Inventory File:**   [2 marks]
    - Create a file called library_inventory.txt containing a list of books and their quantities (e.g.,
      "BookA: 3
       BookB: 2
       BookC: 5").
3. **Student_process.c :**   [3+3+3= 9 marks]
    - This process simulates a student trying to either borrow or return books.
    - It opens the request_pipe and sends a request, which is a structured message such as "**borrow BookA**" or "**return BookB**".   [3 marks]
    - It then listens for a response from **response_pipe**, indicating whether the request was successful or if any error occurred (e.g., "Success: Borrowed BookA" or "Error: BookB not available").   [2 marks]
    - Repeat this process for 5 different requests.   [3 marks]

4. **Librarian_process.c :**   [3+4+2+5= 14marks]
    - This process is responsible for managing the book inventory.
    - It opens library_inventory.txt and reads the current book quantities.   [3 marks]
    - It continuously listens for requests on request_pipe.
        - When a request such as "borrow BookA" is received, it checks if the book is available in sufficient quantity.   [4 marks]

- If available, it decrements the count, updates library_inventory.txt, and sends a success response (e.g., "Success: Borrowed BookA").   **[4 marks]**
- If not available, it sends an error message (e.g., "Error: BookA not available").   **[2 marks]**
- If the request is "return BookA", it increments the count in the inventory file and sends a success response.   **[5 marks]**
  The librarian_process should handle these gracefully, sending appropriate error messages back to student_process.

Q2: You are required to write a program that takes a **2D array** of integers as **command-line arguments**. The first argument specifies the number of rows, and the second specifies the number of columns. The following arguments represent the elements of the 2D array (row-major order). The parent should create 3 or 4 child processes exactly, each responsible for processing different rows of the array. Pipes should be used for communication between parent and child processes.   **[Total= 25 marks]**

Instructions:

1. **Input Format**   **[4 marks]**
   - First argument: Number of rows `R`.
   - Second argument: Number of columns `C`.
   - The remaining arguments represent the 2D array of size `R x C`, provided in a row-major order.
     Example: For a 4x4 matrix
       "1 2 3 4
       5 6 7 8
       9 1 2 3
       4 5 6 7"
     the input will be: ./program 4 4 1234 5678 9123 4567

2. **Child Process 1:**   **[1+2+2= 5 marks]**
   - This child will process the first row of the 2D array.
   - Reverse the elements of row (e.g., 1 2 3 4 becomes 4 3 2 1).
   - Send the reversed rows back to the parent process using a pipe.

3. **Child Process 2:**
   - This child will process the second row of the 2D array.   **[1+2+2= 5 marks]**
   - For each element in these rows, find the square of the number (e.g., for 5 6 7 8, square of digits = 25 36 49 64).
   - Send the squared row back to the parent process using a separate pipe.

4. **Child Process 3:**   **[1+2+2= 5 marks]**
   - This child will process the final row of the 2D array.
   - Compute the sum of all elements (e.g., for 9 1 2 3, sum of digits = 15).
   - Send the results back to the parent process using another pipe.

5. **Parent Process:**
   - Parent will wait for all Childs to return.   **[2+2+2= 6 marks]**
   - The parent process will read the results from each child through pipes and print.