OPERATING SYSTEMS LAB SESSIONAL 2024

[redacted]

Submit the well-documented and tested C/C++ code along with any necessary instructions for compilation and execution, file should be named i21-XXXX_NAME_SEC_X.c. Your submission should have screenshot of your output on terminal attached.

QUESTION# **[20 Marks]**

Problem Statement: Sorting Rows in Parallel

You are tasked with designing a program that sorts a 6x6 matrix filled with even numbers ranging from 2 to 6 inclusively. The sorting operation is to be carried out in parallel by three child processes, each responsible for sorting two rows of the matrix. After sorting, the parent process will merge the sorted rows and display the final matrix. Additionally, each child process will track and store the number of moves (sorting operations) it performs. The communication between processes should be facilitated using shared memory or pipes.

Detailed Requirements:

1. **Matrix Initialization:**
   - Generate a 6x6 matrix filled with even numbers randomly selected from the range [2, 6] inclusive.
2. **Child Processes:**
   - Implement three child processes using fork.
   - Each child process should be responsible for sorting two rows of the matrix.
   - The sorting of rows should be done in parallel, meaning each child process operates independently of the others.
3. **Sorting Algorithm:**
   - Use a simple sorting algorithm such as Bubble Sort or Selection Sort for sorting the rows.
   - The sorting algorithm should be applied separately to each pair of rows assigned to the child processes.
4. **Pipes:**
   - Utilize pipes for inter-process communication.
   - Child processes should store the number of moves (sorting operations) they perform in pipes.
   - The parent process should receive this information via pipes.
5. **Parent Process:**

- The parent process should wait for all child processes to finish sorting.
- After sorting, the parent process should merge the sorted rows to create the final matrix.
- Display the final matrix on the console.
- Additionally, the parent process should display the number of moves made by each child process before merging the sorted rows.

6. **Error Handling:**
   - Implement error handling to deal with issues such as failed process creation, shared memory allocation errors, or inter-process communication failures.
   - Ensure graceful termination of the program in case of errors.