National University of Computer and Emerging Sciences
Department of Computer Science Chiniot-Faisalabad Campus

Chapter 3 – Assignment

Task 1: Process Termination and Cascading Termination

- Write a C program where the parent process creates two child processes. Implement logic where:
  - The parent waits for the children to finish using wait().
  - If the parent process terminates before the children, the operating system should handle their termination (simulate **cascading termination**).

Questions:

1. Explain the concept of **cascading termination** in operating systems.
2. What happens to a child process if the parent process terminates without waiting for the child (creating a **zombie process**)?

Task 2: Shared Memory - Producer-Consumer

- Implement the **Producer-Consumer problem** using **shared memory** and **semaphores** for synchronization.
- The producer process should create and store items in a shared buffer. The consumer process should consume items from the buffer.
- Ensure proper synchronization to prevent **race conditions**.

Task 3: Message Passing - Producer-Consumer

- Implement the Producer-Consumer problem again, but this time using **message passing** (e.g., **pipes** or **message queues**).
- The producer process should send messages to the consumer process via a pipe or message queue, and the consumer should receive and process them.

Questions:

1. Compare **shared memory** and **message passing** for IPC. What are the advantages and disadvantages of each approach?
2. How do semaphores help prevent **race conditions** in shared memory IPC?

3. What are the differences between **blocking** and **non-blocking** communication in message passing?

Detecting and Handling Race Conditions

- Write a C program to demonstrate a **race condition** using a shared variable between two processes.

Questions:

1. What is a **race condition**? Provide an example based on your program.

Deliverables:

- Source code for all programs.
- A README file that explains how to run each program.
- Answers to the conceptual questions.
- A short report comparing the two IPC methods (shared memory vs. message passing).

Total Grade Breakdown:

- **Correctness of the Code**: 40%
- **Completion of All Tasks and Questions**: 30%
- **Clarity of Explanations in the Report**: 15%
- **Code Structure and Documentation**: 15%
