

Laboratory Assignments

Subject: Design Principles of Operating Systems

Subject code: CSE 3249

Assignment 5: Implementation of synchronization using semaphore:

Objective of this Assignment:

- To implement the concept of multi-threading in a process.
- To learn the use of semaphore i.e., to control access to shared resources.

1. Producer-Consumer problem

Problem: Write a C program to implement the producer-consumer program where:

- Producer generates integers from 1 to 50.
- Consumer processes the numbers.

Requirements:

- Use a shared buffer with a maximum size of 10.
- Use semaphores and mutex to ensure thread-safe access to the buffer.
- Print the number that producer is producing and consumer is consuming.
- Both producer and consumer will continue for 20 iterations

2. Alternating Numbers with Two Threads

Problem: Write a program to print 1, 2, 3 ... upto 20. Create threads where two threads print numbers alternately.

- **Thread A** prints odd numbers: 1, 3, 5 ...
- **Thread B** prints even numbers: 2, 4, 6 ...

Requirements:

- Use semaphores to control the order of execution of the threads.
- Ensure no race conditions occur.

3. Alternating Characters

Problem: Write a program to create two threads that print characters (A and B) alternately such as ABABABABA.... upto 20. Use semaphores to synchronize the threads.

- **Thread A** prints A.
- **Thread B** prints B.

Requirements:

- Use semaphores to control the order of execution of the threads.
- Ensure no race conditions occur.

4. Countdown and Countup

Problem: Write a program create two threads where:

- **Thread A** counts down from 10 to 1.
- **Thread B** counts up from 1 to 10.

Both threads should alternate execution.

Requirements:

- Use semaphores to control the order of execution of the threads.
- Ensure no race conditions occur.

5. Sequence Printing using Threads

Problem: Write a program that creates three threads: Thread A, Thread B, and Thread C. The threads must print numbers in the following sequence: A1, B2, C3, A4, B5, C6 ... upto 20 numbers.

- **Thread A** prints A1, A4, A7, ...
- **Thread B** prints B2, B5, B8, ...
- **Thread C** prints C3, C6, C9, ...

Requirements:

- Use semaphores to control the order of execution of the threads.
- Ensure no race conditions occur.