Project 2 Report: Thread-Based Process Simulation and Synchronization

Overview

This project simulates process execution using threads and demonstrates classic synchronization using the **Producer-Consumer** problem. Each thread represents a producer or consumer interacting with a shared buffer.

Implementation Summary

- **Threads**: Two producer and two consumer threads are created.
- **Synchronization**: Used `pthread_mutex_t` for mutual exclusion and `sem_t` for tracking buffer slots.
- **Simulation**: Thread sleep represents CPU burst simulation.
- **Buffer**: Shared circular buffer of size 5.

Sample Input

1 3
2 2
3 1
4 4
5 2

Sample Output Log

[Producer 1] Produced item: 57 at index 0
[Consumer 1] Consumed item: 57 at index 0
...

Challenges Faced

- Preventing race conditions with mutex locking.
- Ensuring correct item order in circular buffer.
- Handling thread synchronization without deadlocks.

Conclusion

The Producer-Consumer implementation successfully demonstrates thread creation and synchronization. Logs clearly show execution steps for each thread, and the buffer is managed without data corruption or access issues.