1. Purpose

This homework builds practical understanding of synchronization methods through implementation and performance evaluation.

2. Description

The goal of this homework is to compare the relative efficiency of different synchronization methods by implementing a shared bounded-buffer (queue) using various techniques discussed in class. A fixed number of items should be produced and consumed for each implementation, from which you will generate statistics to report on your findings.

Common Attributes
1. \( N \) producer threads that generate integers uniformly randomly to add to the queue
2. Sum of produced integers (for debugging)
3. \( M \) consumer threads that take integers off the queue
4. Sum of consumed integers (for debugging)
5. Barriers should be used to synchronize the start of all threads

Queue Implementations
1. Unnamed semaphore
2. Named semaphore (multiple processes not required)
3. Condition variable

Desired Metrics
1. Program throughput (items/second)
2. Average CPU utilization (Use /proc/stat to get this information)

Bonus (in addition to other comparisons)
Use local buffering on the producer/consumer to reduce contention for the queue.

3. Report

Your report (one per group) should contain a description of your code and anything interesting you learned while developing it. Compare each of the queue implementations for different values of \( N \) and \( M \). Draw conclusions about the relative strengths and weaknesses of each approach based on your experiments.