Condition Variables

CSL301 - Operating System Lab
Class Assignment 10

November 4, 2025

Q1: Identify and Implement Ordered Thread Execution

Problem Description

You are given an incomplete program that spawns three threads — A, B, and C — each responsible for printing its name. However, the threads print in random order (e.g., `C A B A C B`), leading to unpredictable output. Your task is to ensure that the threads print strictly in the order A $\rightarrow$ B $\rightarrow$ C, repeating this pattern for `n` cycles.

Given Code: `q1.c`
Reference Code: `q1_help.c`

Q2: Identify Starvation in Reader–Writer Implementation and Fix

Problem Description

You are given a simplified Readers–Writers problem implementation where:
Multiple readers can read simultaneously.
Writers require exclusive access to the shared resource.
However, upon execution, you will notice that writers are starved — if readers keep arriving, writers may never get a chance to write. Complete the given code by adding appropriate condition variable waits and signals to ensure fairness between readers and writers.

Given Code: q2.c

Q3: Identify and Fix Missing Synchronization in Thread-safe Logger

Problem Description

You are given a multi-threaded logging system in which several worker threads generate log messages, and a single logger thread writes these messages to a file. However, the provided implementation is missing proper synchronization between the workers and the logger. As a result:

- The logger thread may busy-wait, consuming high CPU while checking for new messages, or
- Some log messages may be lost if worker threads finish too quickly.

Your task is to identify and fix this synchronization issue using proper condition variable mechanisms.

Given Code: q3.c

Q4: Identify and Implement Sleep/Wakeup Mechanism

Problem Description

You are given a program that simulates a simplified operating system scheduler, where multiple threads can go to sleep and later be woken up by another thread. However, the current implementation is incomplete — the sleep and wakeup behavior does not work correctly. When you run the program, you may observe that:

- Some threads continuously spin, using CPU while "waiting," or
- Sleeping threads never wake up when the waker tries to signal them.

Each thread in the program has its own structure. Implement the correct behavior using condition variables.

Given Code: q4.c

Q5: Identify Busy-wait and Fix Job Scheduler

Problem Description

You are given a job scheduling system with:

- One dispatcher thread that assigns jobs.
- Multiple worker threads that execute jobs.

Currently, the implementation uses busy-waiting — both the dispatcher and the workers continuously check job counts, wasting CPU time. Identify that this issue is caused by missing condition variable waits, and fix the code.

Given Code: q5.c

Submission Checklist

- Ensure your code compiles without warnings or errors.
- The program should run and produce correct output.
- Submit all files in a single compressed folder.
- Attach screenshots of your final output in the report.

Good Luck!