Results

100%	10	02:12
Score	Out of 10 points	Time for this attempt

1 2 / 2 points

What is the primary challenge addressed by the Producer-Consumer problem?

○ Ensuring data integrity

✓ ● Preventing race conditions

○ Sharing data between processes

○ Maximizing CPU usage

Feedback

Based on your answer

The main challenge of the Producer-Consumer problem is to ensure that producer and consumer threads can operate concurrently without causing race conditions, which can lead to data corruption or inconsistencies.

2 / 2 points

2 2 / 2 points

Which mechanism is often used to synchronize access to a shared buffer in the Producer-Consumer problem?

○ Busy waiting

○ Priority scheduling

○ Thread preemption

✓ ● Condition variables

2 / 2 points

3 2 / 2 points

What is the major problem with the initial solution for the ProducerConsumer problem presented in the PPT?

○ It fails to block the producer or consumer when necessary.

○ It doesn't use mutexes to protect shared data access.

✓ ● It suffers from busy waiting, wasting CPU cycles.

○ It lacks a clear algorithm for producing and consuming data.

Feedback

Based on your answer

The initial solution in the PPT involves busy waiting, which means threads repeatedly check for conditions to be met, wasting CPU resources while they could be doing other useful work.

2 / 2 points

4 2 / 2 points

How can the problem of busy waiting in the Producer-Consumer problem be effectively resolved?

○ By increasing the thread's privacy to avoid waiting

✓ ● By using condition variables or semaphores

○ By using spinlocks to optimize waiting times

○ By printing debug messages for better monitoring

Feedback

Based on your answer

Condition variables or semaphores can be used to block threads when they cannot make progress, avoiding busy waiting and efficiently utilizing CPU resources.

2 / 2 points

5 2 / 2 points

What library call is commonly used to decrease a semaphore's value, potentially causing a thread to be blocked?

○ sem_post()

✓ ● sem_wait()

○ sem_lock()

○ sem_signal()

2 / 2 points

Fudge Points

Manually adjust the score by adding positive or negative points to this box

0

Final Score **10 / 10**