ECSE 427: Assignment 3

Daniele Bercovici, 260627845

April 10, 2018

Contents

1 Question 1

When we already use 2 semaphores in the producer-consumer problem, why is there a need for mutex?

Having a mutex gives mutual exclusion to the queue. Essentially acting as a lock, where either the producer or the consumer has the key. It synchronizes the access to the resource, unlike semaphores that are a are a signaling mechanism, there is ownership associated with it.

2 Question 2

Is it possible that a consumer with lowest priority suffers from starvation in the 2 semaphore and 1 mutex setup for producer-consumer. Explain the situation

Yes it is possible especially if there are a lot of consumers compared to producers. The higher priority consumer will consistently be chosen over the lower priority. A better way is FIFO or round robin.

3 Question 3

Though binary semaphores avoid starvation and mutexes don't, why is it recommended to use mutex in producer consumer to serve critical section?

Mutexes are used because only that thread may unlock it (ownership). You will acquire the mutex (lock), enter critical section, and release mutex (unlock) all in the same thread. While using a semaphore, you can make a thread wait on a semaphore (say thread A), until another thread (say thread B) completes the task, and then sets the Semaphore for thread A to stop the wait, and continue its task.

4 Question 4

Why do producer-consumer need to have 2 semaphores namely "Full" and "empty". What complications may arise if we use only one semaphore for "Full" and rely on computed complementary value for "empty"

"Full" is used to block the consumer if buffer is empty, "empty" is used to block producer if buffer is full. Since items in the buffer are a counted resource and empty buffer slots are also a counted resource, two semaphores are necessary. There is a redundancy because the two semaphore counts are not independent. Their sums remain constant and equal to the buffer size. However, the redundancy is not complete enough to allow the elimination of one of the semaphores. If only one semaphore "Full" is used, a complication that might arise is that perhaps before the updated value of empty can be calculated there is an interrupt. This can result in a race condition.