ECSE 427: Assignment 3 Report

# Part 1: Theoretical Questions

1. ***When we already use 2 semaphores in producer-consumer problem why is there a need for mutex?***

While the 2 semaphores are used to synchronize the count of the shared queue amongst threads, they only cause threads to wait at the boundary conditions, I.e. only empty or full queue. A mutex is needed to ensure that no two or more threads can read/write to the queue at the same time.

1. ***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.***

No, the producer-consumer problem works on a first come first served (FCFS) system. However if priority were to be introduced should the queue be filled and new higher priority consumers arrive faster than the queue can be emptied the lower-priority consumer may starve as each other consumer will be serviced first.

1. ***Though binary semaphores avoid starvation and mutexes don’t, why is it recommended to use mutex in producer-consumer to secure critical section?***

Mutex’s have the property of ownership, i.e. the thread that locked it is the only one that can unlock it. This is opposed to binary semaphores, and as such it prevents any other thread from opening up the critical section. This from a security standpoint is very valuable.

1. ***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”***

F