COMP 310 – Assignment 3, Part 1

Byungwoo Kim
ID 260621422

1. We need to use mutex as well when we already use 2 semaphores in the producer-consumer problem in order to ensure that only one process runs at a time for a specific part (critical section) of the code. This is necessary for the producer-consumer problem because we don't want two different processes changing the value of a shared variable at the same time for a specific part.

2. Yes, it is possible that a consumer with lowest priority suffers from starvation in the 2 semaphore and 1 mutex setup for producer-consumer problem. The purpose of semaphores is to stop some processes at a certain point until another process is done with a specific part to avoid unnecessary changes in the variables. Therefore, if one process is unable to complete the part and trigger the semaphore, then a consumer with lower priority could suffer from starvation.

3. Similar to question 1, the producer-consumer problem deals with shared variables that are sensitive to changes. Any change in the sequence of change to the variable could result in the wrong result if mutex is not used for the critical section.

4. The "full" semaphore is used to stop the producer from producing any more products until any are taken because of "full warehouse", and the "empty" semaphore is used to stop the consumer from taking any more products from the warehouse until any are produced. So it is evident that both are needed, because there are two "point of views" to look at this problem at, the producer's and the consumer's, and it is very important that the processes are controlled with two semaphores.