ECSE 427 – Assignment 3, Part 1

Alex Hale
ID: 260672475

1)

a. If the mutex variable were not present in the producer-consumer solution, there would be no issues in the case of a single producer and a single consumer. However, if there are multiple producers and/or multiple consumers, there is a possible race condition that could result in two producers writing into the same slot, or two consumers reading from the same slot. To stop this from happening, critical sections are implemented where a producer is adding to the buffer and where a consumer is reading from the buffer. These critical sections, protected by the mutex variable, ensure that no two producers and no two consumers add to or remove from the same slot in the buffer.

b. Starvation is possible in this scenario. The 2 semaphore + 1 mutex solution to the producer-consumer problem has no authority on the priority of processes. If multiple processes with high priority are drawing from the buffer at a rate greater than or equal to the rate at which the producers are adding to the buffer, then the lowest-priority process(es) could starve. The semaphores and mutex have no effect on this problem.

c. A mutex variable is under the control of the process that is using it. No other process can give away control of the mutex variable. Binary semaphores are not strictly controlled by the process holding them, so other processes can give away control. Therefore, a mutex is better for locking down a critical section (because the process is guaranteed not to be interrupted during its execution of the critical section), while a binary semaphore is more useful for synchronization tasks.

d. The producer in the producer-consumer problem is also a sort of consumer, in that it consumes the empty slots of the buffer. Therefore, two semaphores are required to keep track of the two counted resources: items in the buffer, and empty buffer slots. If only one semaphore is used, the algorithm will fail because the producer won't stop producing when the buffer is full, and if the consumer waits on an empty buffer, it could block the producer from adding to the buffer.