CISC 324 Assignment 2

Fall 2022

Student #:20206993

Dajung Yoon

Question 1: False

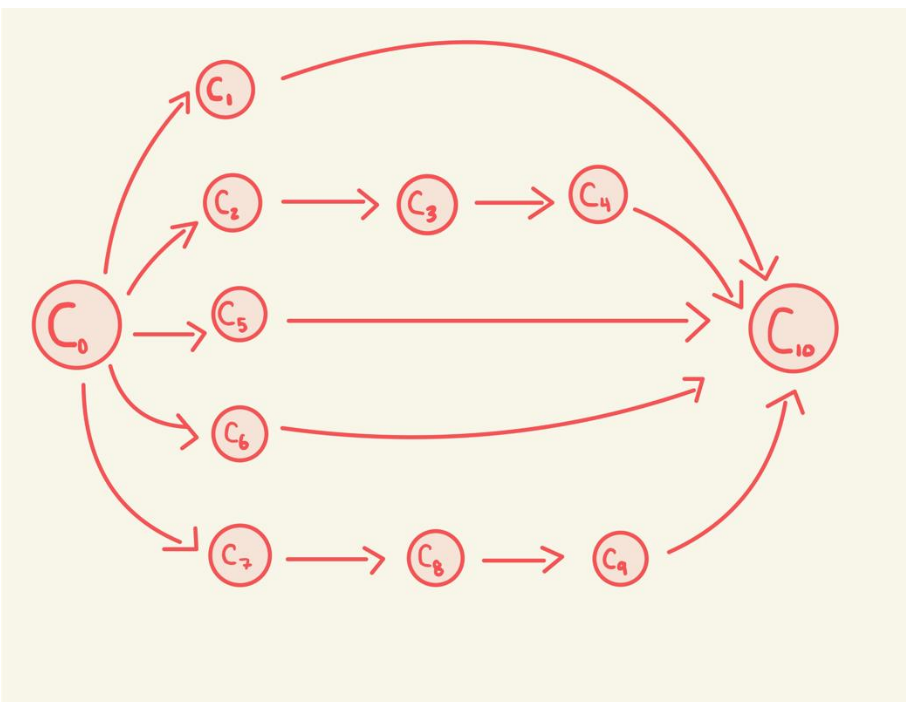
Question 2: d) all of the above

Question 3: One of the problems with semaphores is that since it is a complicated operation, the wait and signal operation must be in the correct order. Or else, there will be deadlock.

If the semaphore use busy waiting, then that is another issue. Busy waiting is when a process in critical section, and there is another process trying to enter the critical section, that process will have to wait until it is not taken by the previous process. Busy waiting also wasted CPU cycles that other processes could be using. When Busy waiting, incorrectly used: hard to detect timing errors/violation of critical section problem criteria.

Another issue is when the semaphore does not use busy waiting, then there is a possibility to deadlock and starvation. Starvation occurs when there is one or more process in the program that is blocked from getting access to a section. Due to that reason, the process is "starved" and cannot make progress. Deadlock is a part of starvation, when two or more process's are waiting on a condition that cannot be satisfied.

Question 4:



Question 5:

Yes a process can be interrupted during the execution, it would be switching a process from the running state to the ready state. You have to save the context of that process. If you are not saving the context of any process P , then after some time, when the process P comes in the CPU for execution again, the process will start executing.

This could happen for the following reasons:

1. If a process with high priority is in ready state. If this is the case, then the execution of the running process will stop and then the high priority process will be put in the CPU to be executed.
2. If there is a transition needed between the user mode and kernel mode, then context switching is required. If the kernel is preemptive, then this will allow the process to run. If it is non-preemptive, then it is free from race conditions and only one process is alive.

Question 6:

In the RAG diagram, P1, R1, P3, R3, and P5 is in a state without deadlock. However, P2, R2, P4, and R4 are in a state with deadlock. Since the right process's in the RAG in deadlock, if there ever is an instance when P1, P3, or P5 request R2 or R4, then they would be put in deadlock. The process is not satisfied because not all the process's has the required resources. Therefore, the following system is in deadlock.