COMP30640 Operating Systems: Quiz 6

Exercise 1 – Synchronisation

Discuss Dijkstra's conditions for a true solution to the problem of synchronisation on a critical section.

Sample Solution 1

Dijkstra identified 3 conditions:

1) Mutual exclusion: one process at most inside the CS at any time
2) Progress
   - A process in execution out of a CS cannot prevent other processes from entering it
   - If several processes are attempting to enter a CS simultaneously the decision on which one goes in cannot be indefinitely postponed
   - A process may not remain in its CS indefinitely (neither terminate inside it)
3) Bounded waiting (no starvation): A process attempting to enter its CS will eventually do so

Exercise 2 – Semaphores

Consider a semaphore S that is initialised to 1, and a critical section (CS) shared by processes A, B, and C which is protected by that semaphore.

1) What happens when a process A executes P(S) (that is to say, when it probes that semaphore)? (answer in terms of the value of the semaphore S and the effect on process A).
2) While process A is inside the CS, what happens when process B executes P(S) (answer in terms of the value of the semaphore S and the effect on processes A and B)?
3) While process A is still inside the CS, what happens when process C executes P(S) on that semaphore (answer in terms of the value of the semaphore S and the effect on processes A, B and C)?

Sample Solution 2

1) S is decremented from 1 to 0. A continues into the critical section. No other processes are affected.
2) Since S==0, B blocks and is added to a queue of processes waiting on S. A is already in the critical section, and there is no effect on it.
3) Since S==0, C blocks and is added to a queue of processes waiting on S (along with B). A is still in the critical section and there is no effect on it.

Exercise 3 – Semaphores (2)

Discuss whether the following statements are true or false, with respect to a semaphore S with associated functions P(S) and V(S). Give examples (or counterexamples) to clarify your answers.

1) P(S) always enqueues the calling process, waiting for the critical section to be free.
2) S cannot be modified by an operation other than P(S) and V(S).
3) The initial value of S can only be zero for binary semaphores.
4) S cannot take values greater than one

Sample Solution 3

1) False, the calling process is only enqueued when another process is inside the critical section.
2) True, from the definition of semaphore the nonnegative integer variable is protected.
3) False, see for instance the producer/consumer algorithm with semaphores.
4) False, see counting semaphores