Exercise Session
Synchronization / Resource allocation

Operating Systems

Chalmers University of Technoloy – Gothenburg University

Question 1

We have three threads A, B, and C taking care of operations opA, opB, and opC respectively. The threads use three semaphores with the following initial value: semA=1, semB=1, and semC=0.

Thread A:

```
wait(semC);
wait(semB);
opA; // some operation
signal(semB);
signal(semA);
```

Thread B:

```
wait(semA);
wait(semB);
opB; //some operation
signal(semB);
```

Thread C:

```
wait(semA);
wait(semB);
opC; //some operation
signal(semB);
signal(semA);
signal(semC);
```

Are the following executions possible or not and why?

(i) opA opB opC

(ii) opB opC opA

(iii) opC opA opB

(iv) opB opA opC

Question 2

(a) Consider two threads, A and B. Thread B must execute operation opB only after thread A has completed operation opA. How can you guarantee this synchronization using semaphores?

(b) Consider two threads, A and B which must forever take turns executing operation opA and operation opB, respectively. Thread A must be the one that executes opA first. How can you guarantee that using semaphores?

Question 3

Servers can be designed to limit the number of open connections. For example, a server may wish to have only N active socket connections at any point in time. As soon as N connections are made, the server will not accept a new connection until an existing one is released. With pseudocode, describe how semaphores can be used to limit the number of concurrent connections.

Question 4

Show a method that solves the critical section problem for arbitrary number of threads using the atomic TestAndSet instruction that is available in several processor architectures. Use pseudocode in the description and argue about the properties of the solution, with respect to mutual exclusion, progress and fairness. (It is not necessary to describe a solution that guarantees fairness in this question, but if you can, of course it is ok).