Exercise Session
Synchronization / Resource allocation

Operating Systems, EDA093 - DIT400

Hannaneh Najdataei

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).

Question 5

A two way east-west road contains a narrow bridge with only one lane. An eastbound (or westbound) car can pass over the bridge only if there is no oncoming car on the bridge. Traffic may only cross the bridge in one direction at a time, and if there are ever more than 3 vehicles on the bridge at one time, it will collapse under their weight. In this system, each car is represented by one thread, which executes the procedure OneVehicle when it arrives at the bridge.

```
OneVehicle(Direction direc) {
    ArriveBridge(direc);
    CrossBridge(direc);
    ExitBridge(direc); }
```

direc gives the direction in which the vehicle will cross the bridge

Write the procedures *ArriveBridge* and *ExitBridge*. *ArriveBridge* must not return until it safe for the car to cross the bridge in the given direction (it must guarantee that there will be no head-on collisions or bridge collapses). *ExitBridge* is called to indicate that the caller has finished crossing the bridge; *ExitBridge* should take steps to let additional cars cross the bridge.