**Assignment 6**

**Due**: 11:55PM Oct. 30, 2023

**Submission**: Complete your submission on Blackboard.

Email submission will NOT be accepted. If you cannot complete the assignment before the deadline due to Covid-19, you must contact the instructor **BEFORE** the deadline. Otherwise, **late submission will NOT be accepted**.

**Total points**: 20 points

1. Analyze the following code (10 points)

// process 0

flag[0] = true; // I want in

while (flag[1]) // p1 is in

{

flag[0] = false; // defer to p1

// delay

flag[0] = true; // try again

}

//… code of critical section

flag[0] = false; // I’m out

// process 1

flag[1] = true;

while (flag[0])

{

flag[1] = false;

// delay

flag[1] = true;

}

//… code of critical section

flag[1] = false;

1.a. What is the problem in the above pseudocode?

(a) mutual exclusion is violated (b) deadlock (c) live lock

**Live Lock**

1.b. Write the detailed execution sequence that causes the above problem.

A math equations on a lined paper

Description automatically generated

2. Use the hardware instruction “testset” (Refer to slide 33 in Chapter 5 about the definition on testset) to implement mutual exclusion. Fill in blanks in the following code. (4 points)

Shared variable x= 0

Process P1()

{

while ( !testset(x) )

// do nothing

// critical section code here

x = 0

// remaining code…

}

3. Complete the following table. The initial value of semaphore s is 1. The semaphore s is using a first-in-first-out order. If the blocked queue is empty, write down “N/A”. (6 points)

|  |  |  |
| --- | --- | --- |
| Action | The new value of semaphore s | Blocked Queue |
| Thread A calls “wait(s)” | 0 | N/A |
| Thread B calls “wait(s)” | -1 | B |
| Thread C calls “signal(s)” | 0 | N/A |

END