Mary Mitchell: mem0250

COMP 3500: Homework 1

Points Possible: 100
Submission via Canvas
There should be no collaboration among students. A student shouldn't share any
project code with any other student. Collaborations among students in any form will be
treated as a serious violation of the University's academic integrity code.
Goals:
- To learn multiple processes.
- To understand critical sections.
- To learn the concepts of semaphores and monitors.

Questions:

1. [60 points] Consider the following program:

```
P1: {                                    P2: {
  shared int x;                            shared int x;
  x = 10;                                  x = 10;
  while (1) {                              while ( 1 ) {
     x = x - 1;                               x = x - 1;
     x = x + 1;                               x = x + 1;
     if (x != 10)                             if (x!=10)
       printf("x is %d",x)                      printf("x is %d",x)
  }                                        }
}                                        }
```

Note that the scheduler in a uniprocessor system would implement pseudo parallel
execution of these two concurrent processes by interleaving their instructions, without
restriction on the order of the interleaving.

1.1. [25 points] Show a sequence (i.e., trace the sequence of interleavings of statements)
such that the statement "x is 10" is printed.

P1: x = x – 1;	**/* x = 9 */**
P1: x = x + 1;	**/* x = 10 */**
P2: x = x – 1;	**/* x = 9 */**
P1: if (x != 10)	**/* true: enter if-body */**
P2: x = x + 1;	**/* x = 10 */**
P1: printf("x is %d", x)	**/* prints "x is 10" */**

1.2. [35 points] Show a sequence such that the statement "x is 8" is printed.
You should remember that the increment/decrements at the source language level are
not done atomically, that is, the assembly language code:

```
LD R0,X /* load R0 from memory location x */
INCR R0 /* increment R0 */
STO R0,X /* store the incremented value back in X */
```

P1: x = x – 1;	**/* x = 9 */**
P2: LD R0, x	**/* load R0 from memory location x */**
P2: DECR R0	**/* decrement RO (R0 contains 8) */**
P1: x = x + 1;	**/* x = 10 */**
P2: STO R0, x	**/* store the decremented value back in x (x = 8) */**

```
P1: if (x != 10)              /* true: enter if-body */
P1: printf("x is %d", x)      /* prints "x is 8" */
```

2. [10 points] What is the difference between binary and general semaphores?

The value of a binary semaphore can range only between 0 and 1 (somewhat similar to how a mutex lock behaves). On the other hand, the value of a general or counting semaphore can range over an unrestricted domain.

3. [10 points] What is a monitor?

Monitors are a high-level abstraction for process synchronization. The monitor type is an abstract data type which includes operations that are provided with mutual exclusion within the monitor. Since it is an abstract data type, the internal variables are only accessible by code within the procedure. The monitor construct ensures that only one process at a time is active within the monitor.

4. [20 points] What operations can be performed on a semaphore?

A semaphore has two main operations: wait() and signal(). The wait operation checks to see if the semaphore indicates any available resources. If there are no resources, then the process must wait (this could be a busy wait depending on the implementation). Once a resource becomes available, the process stops waiting and the semaphore is decremented. The signal operation increments the semaphore, indicating that a resource is now available and the next waiting process can access it.

Submission:

- A heading at the top of your file contains your name and your Auburn UserIDs.
- Submit your solution as a single PDF file named as "hw1.pdf" through Canvas
- File formats other than PDF will not be accepted by Canvas.

Late Submission Penalty:

- Ten percent (10%) penalty per day for late submission. For example, an assignment submitted after the deadline but up to 1 day (24 hours) late can achieve a maximum of 90% of points allocated for the assignment. An assignment submitted after the deadline but up to 2 days (48 hours) late can achieve a maximum of 80% of points allocated for the assignment.
- Assignment submitted more than 3 days (72 hours) after the deadline will not be graded.

Rebuttal period:

- You will be given a period of one week (i.e., 7 days) to read and respond to the comments and grades of your homework or project assignment. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.