CS4328: Homework #3

Due on April, 11, 2019

Mina Guirguis

You may discuss this problem set with other students. However, you must write up your answers on your own. You must also write the names of other students you discussed any problem with. Each problem has a different weight. Please state any assumptions you are making in solving a given problem. Late assignments will not be accepted with prior arrangements. Assignments are due in class.

Problem 1

Consider the following program running on a single processor machine: [**10 pts**]

```
const int n = 50;
int tally;

void total() {
   int count;
   for (count = 1; count<=n; count ++)
      tally++
}

void main() {
   tally = 0;
   parbegin(total(),total()); // parbegin executes the functions in parallel
   write tally;
}
```

(a) What is the lower bound and upper bound on the final value of the shared variable tally in this concurrent program. Assume processes can execute at any relative speed and that the value can only be incremented after it has been loaded into a register by a separate machine instruction.

(b) If we have N processes executing the function total() in parbegin as opposed to 2, what effect will this modification have on the range of final values of tally?

Problem 2

Show that, if the wait() and signal() semaphore operations are not executed atomically, then mutual exclusion may be violated. [**10 pts**]

Problem 3

The Cookie Eater Problem. Consider a system with three cookie eater processes and one agent process. Each cookie eater continuously makes a cookie and eats it. But to make a cookie, three ingredients are needed: milk, flour, and sugar. One of the processes has milk, another flour and the third has sugar. The agent has infinite supply of all three. The agent places two of the ingredients, at random, on the table. The eater who has the remanning ingredient can then make and eat the cookie, signalling the agent upon completion. The agent then puts out two other ingredients and the process repeats. Write a program to synchronize the agent and the eaters.[**20 pts**]

Problem 4

Consider the following snapshot of a system. Answer the following questions: [**20 pts**]

Current available:

R1	R2	R3	R4
2	1	0	0

Current Allocation:

P-R	R1	R2	R3	R4
P1	0	0	1	2
P2	2	0	0	0
P3	0	0	3	4
P4	2	3	5	4
P5	0	3	3	2

Maximum Claim:

P-R	R1	R2	R3	R4
P1	0	0	1	2
P2	2	7	5	0
P3	6	6	5	6
P4	4	3	5	6
P5	0	6	5	2

(a) Compute what each process might still need.

(b) Is this system in a safe or unsafe state? Why?

(c) Is this system currently deadlocked? Why or why not?

(d) Which processes, if any, are or may become deadlocked?

(e) If a request from P3 arrives (0,1,0,0), can that request be safely granted? If granted, what would be the resulting state (safe, unsafe, deadlocked)? Which processes, if any, are or may become deadlocked if this request was immediately granted?

Problem 5

(a) Three processes share four resources units that can be reserved and released only one at a time. Each process needs a maximum of two units. Show that a deadlock cannot occur. [**5pts**]

(b) N processes share M resource units that can be reserved and released only one at a time. The maximum need of each process does not exceed M, and the sum of all maximum needs in less than M+N. Show that a deadlock cannot occur.[**5 pts**]