LSU

LOUISIANA STATE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CSC 4103: Operating Systems
Spring 2022
DUE MARCH 21 @ CLASS TIME
NO LATE SUBMISSIONS

****** MIDTERM IS MARCH 23 ******

****** MLFQS DUE MARCH 23 ******

Written Assignment Rules

- All work must be your own. You are not to work in teams on this assignment. You are not to use materials from previous offerings of this course. Do not look online for solutions. Do your own work!
- Your solution **must** be typed or it will not be graded.
- Turn in a printed copy of your homework on the due date.

(1)

Prove that Peterson's algorithm (Slide # 19 in Slide Set # 3) correctly ensures two-process mutual exclusion.

(2)

Prove that the Bakery algorithm (Slides 24-28 in Slide Set # 3) correctly ensures n-process mutual exclusion.

(3)

Write (in pseudocode) a **STRONG WRITERS** solution to the readers-writers problem using semaphores. You must indicate if waiting readers must wait until ALL waiting writers have proceeded (STRONG STRONG writers) or not (just STRONG writers).

(4)

The "solution" to the two-process mutual exclusion problem below was actually published in the January 1966 issue of *Communications of the ACM.* It doesn't work! Your task: Provide a detailed counterexample that illustrates the problem. As in lecture, assume that simple assignment statements (involving no computation) like `turn = 0` are atomic.

```
Shared Data:

          blocked: array[0..1] of Boolean;
          turn: 0..1;

          blocked[0] = blocked[1] = false;
          turn = 0;

Local Data:

          ID: 0..1;   /* (identifies the process;
                         set to 0 for one process,
                         1 for the other) */

Code for each of the two processes:

          while (1) {

               blocked[ID] = true;
               while (turn <> ID) {
                    while (blocked[1 - ID]);
                    turn = ID;
               }

               << critical section >>

               blocked[ID] = false;

               << normal work >>

          }
```