

# Computer Architecture and Operating Systems - Winter 2018

Sambuddho Chakravarty

November 7, 2019

## Assignment 3 (Total points: 45)

Due date: Nov 23, 2018. Time: 23:59 Hrs.

### 1 Multiple reader writer with synchronization (Total points: 45)

You are already aware of the multiple user reader writer problem wherein multiple “writers” can write to a shared queue and multiple “readers” can read data off the shared queue. Your task is to implement the multiple reader–writer problem in such a way that there are no race conditions. While a writer “writes” a certain element to the queue, no “reader” can read that element. The readers may read other elements but not the one current written to. Similarly, only one “writer” can update the queue at a time. Multiple “readers” may read elements off the queue but dequeuing from the queue by the “readers” must be atomic.

You may use various system synchronization primitives for the same, *e.g.* `pthread_mutexes`, `semaphores` *etc.*. If you wish to you may additionally also use IPC primitives which were used in the previous assignment.

#### What To Submit

- Program source code with Makefile
- Write-up describing the following:
  - Description of your code and how you implemented the function – the logical and implementation details.
  - Description of how to compile and test the program
  - The inputs the user should give.
  - Expected output (and how to interpret it).
  - Error values and how to interpret them.

#### Grading Rubric

- Successful compilation your the program – 10 points.

- Correct functioning of the chat system for multi-users – 20 points.
- Correct handling of input errors (atleast two different types of errors should be handled) – 10 points.
- Description of the systems, test cases *etc.* – 5 points.