CS 423/523 Assignment 3

Published: Feb. 26, 2016
Due: Mar. 6, 2016 (11:59pm)

Total: 30 points

Please upload your solutions to classes*v2. To do so, please enter classes*v2, then click the “Assignment” button on your left-hand toolbar, and finally click “Assignment3” to upload your assignment.

If you know you are going to submit your assignment late, please let us know in advance (send an email to cs423ta@cs.yale.edu). Solutions will be posted 10 days after the deadline.

Any and all resources may be used as long as you cite them, with the exception of collaborating with other people. Please do not copypasta your definitions from Wiki. We do not really care if you do this, but you are not really learning anything.

If you have ANY questions, please do not hesitate to let us know (email, office hours, etc.)
Part 1: OS definitions (1-3 line answers, 5 x 2 points each = 10 points)

1. Critical section
2. Semaphore
3. Deadlock
4. Starvation
5. Busy waiting

Part 2: Multiple choice (6 x 2 points each = 12 points)

1. Which of the following is correct about a binary semaphore:
   a. Is a hardware-based synchronization mechanism
   b. Requires busy waiting
   c. Both of the above
   d. None of the above

2. Test and set:
   a. Returns a pointer type
   b. Is interruptible
   c. Both of the above
   d. None of the above

3. Which of the following is not the property of Peterson’s algorithm:
   a. Can handle multiple processes case
   b. Instructions, e.g., load and store, should not be interrupted
   c. A reasonable software solution for critical-section handling
   d. None of the above

4. Which of the following is not correct about the semaphore with no busy waiting:
   a. Each semaphore has an associated waiting queue
   b. Block operation places the process on the waiting queue
   c. Wakeup operation directly puts the process in the ready queue
   d. None of the above

5. Which of the following is correct about mutex lock
   a. The value is either 1 or 0
   b. Requires busy waiting
   c. None of the above
   d. Both of the above

6. Which of the following is not correct about semaphore
   a. We can implement a counting semaphore as a binary semaphore
   b. Semaphore offers more sophisticated ways for synchronization than mutex lock
   c. Can be used to implement bounded-buffer problem
   d. Is the same as the purpose of spinlock
Part 3: A longer question (4 x 2 points each = 8 points)

```c
1 do {
2     wait(rw_mutex);
3     ...
4     /* writing is performed */
5     ...
6     signal(rw_mutex);
7 } while (true);
```

Writer algorithm

```c
1 do {
2     wait(mutex);
3     read_count = read_count + 1;
4     if (read_count == 1)
5         wait(rw_mutex);
6     signal(mutex);
7     ...
8     /* reading is performed */
9     ...
10    wait(mutex);
11    read_count = read_count -1;
12    if (read_count == 0)
13        signal(rw_mutex);
14    signal(mutex);
15 } while (true);
```

Reader algorithm

The above code is extracted from the page 42 and 43 of cha5.ppt, i.e., the first version of writer and reader algorithm design for readers-writers problem. Please answer the following questions.

1. What is the key purpose of line 2 and line 6 in the reader algorithm (just answer one purpose)?
2. Why we need line 4-5 in the reader algorithm? In other words, without line 4-5, what may happen?
3. Why we need line 12-13 in the reader algorithm? In other words, without line 12-13, what may happen?
4. Whether this readers-writers design has starvation problem? Why?