

Homework 2

1. What is the role of the `contentionscope` for pthreads? How does it affect the execution of pthreads?
2. Assume you have a system that does not provide a `usleep(unsigned long usec)` call to suspend the execution of the thread for a given amount of time, say in μ secs. How would you implement this function using condition variables? (Describe your implementation in form of a C/C++ code snippet.)
3. [1] Which of the scheduling algorithms listed below could result in starvation?
 - (a) First-come, first-served
 - (b) Shortest job first
 - (c) Round robin
 - (d) Fixed priority
4. [1] Servers can be designed to limit the number of open connections. For example, a server may wish to have only N connections active at any time. As soon as N connections are established, the server will not handle more connections until at last one existing connection is closed. How would you use semaphores to limit the number of concurrent connections? Assume that you have a function called `HandleNextIncomingConnection()` and one called `CloseExistingConnection()`, which are called before and at the end of handling a connection. How would you add the necessary synchronization code to limit the number of concurrent connections to at most N ?
5. [3] Ten processes share a critical section implemented by using a semaphore x . Nine of these processes use the code `x.P(); <critical section>; x.V()`. However, one process erroneously uses the code `x.V(); <critical section>; x.P()`. What is the maximum number of processes that can be in the critical section at the same time?

6. You just started a new position at Banana, Inc., and your new boss knows that you have done well in CSCE 313. Your first assignment therefore is to design the entry protocol for Banana, Inc.'s unisex bathroom¹. Your boss specifies the following two requirements:

- (a) There cannot be men and women in the bathroom at the same time.
- (b) There should never be more than three employees wasting company time in the bathroom.

In the interest of both Banana, Inc. and of its employees, your solution should avoid deadlocks.

Describe the implementation of your solution by adding the necessary synchronization to the female code listed below. (The male code should be similar.)

```
// declare and initialize semaphores here

for(;;) { // forever
    // this code is for female employees. male code is similar.
    <here the employee is at his/her desk>

    // here comes the code for the employee to access the bathroom.

    <here the employee is in the bathroom>

    // here comes the code for the employee to leave the bathroom.

}
```

Hint: I would help to have the following semaphores: `empty` to denote that the bathroom is empty, and either gender can enter, and `women.multiplex` and `men.multiplex` to limit the number of users in the bathroom.

Bonus: It is likely that your solution does not address starvation. A long line of men can come and enter while a woman is waiting, and vice versa. You get 5 BONUS POINTS if your solution does not allow this to happen.

¹A version of this problem appeared in Andrew's book *Concurrent Programming*.

References

- [1] A. Silberschatz, P. Galvin, and G. Gagne, *Applied Operating Systems Concepts*, John Wiley & Sons, Inc., New York, NY, 2000.
- [2] Deitel, Deitel, and Choffnes, *Operating Systems*, Pearson / Prentice Hall, 2004.
- [3] D. M. Dhamdhere, *Operating Systems, a Concept-Based Approach*, McGraw Hill, 2009.