

# CS4320 (Principles of Operating Systems): Homework 1

**Assigned:** September 12, 2000.

**Due:** September 19, 2000, by 5pm.

**Credit:** 30 points.

## Reading

Be sure you have read chapter 1, and section 4 of chapter 6.

## Problems

Turn in hardcopy answers (either handwritten or generated by your favorite word-processing or text-formatting program) to the following problems.

1. (4 points) Exercise 1.4 from Silberschatz and Galvin.
2. (4 points) Exercise 1.8 from Silberschatz and Galvin.
3. (4 points) In class we discussed a semaphore-based solution for the “critical-section” problem (also known as the “mutual exclusion” problem). (The same solution is presented in slightly different form on p. 168 of Silberschatz and Galvin.) Would this solution still work if the semaphore *wait* and *signal* operations were not executed atomically? If you answer “yes”, briefly explain why. If you answer “no”, provide a counterexample.
4. (4 points) In class we discussed a semaphore-based solution to the bounded-buffer problem. (The same solution is presented in slightly different form on p. 173 of Silberschatz and Galvin — figures 6.12 and 6.13). Would this solution still work (i.e., would it still meet the requirements we set for the original solution) if the order of the two *signal* operations in both producer and consumer processes were reversed? i.e., if producers first perform *signal(full)* and then *signal(mutex)*, and similarly for consumers. If you answer “yes”, briefly explain why. If you answer “no”, provide a counterexample.
5. (7 points) Write a semaphore-based solution to the problem described in Exercise 6.7 from Silberschatz and Galvin. Explain briefly why you believe your solution is correct. (You do not need to do a “proof” similar to those in class, although you may if you like.)

*Hint:* You may want to structure your answer in terms of two kinds of processes, a barber and customers, outlined thus:

```
barber:
    while (true) {
        get customer from waiting room
        serve customer
    }
customer:
    if (waiting room full)
        do nothing
    else {
        enter waiting room
        wait to be served
    }
```

Remember to specify initial value(s) for any semaphore(s) used in your solution.

6. (7 points) Write a semaphore-based solution to the problem described in Exercise 6.8 from Silberschatz and Galvin. Explain briefly why you believe your solution is correct. (You do not need to do a “proof” similar to those in class, although you may if you like.)