

CMSI 387

OPERATING SYSTEMS

Spring 2007

Assignment 0315

It's spring break! Time to...write code. And do some exercises.

Not for Submission

1. Read Chapters 6 and 7 in SGG; if you would like supplementary notes on Section 6.9 and some of Chapter 7, see the “Concurrency Control” handout.
2. Read the various “classic” papers that have been given out on concurrent programming and synchronization. A useful exercise is to look at how those papers express their ideas, and to see how those ideas are now expressed in the textbook and other more recent material, such as Wikipedia, documents on current operating systems and APIs, and others. You can also look at the code presented in these papers, and see how that code would translate into the languages that are in use today.

For Submission

Dining Philosophers

Implement a solution to the dining philosophers problem using the POSIX API. Use the given bounded buffer solution as a pattern. Provide well-placed print statements to report the work done by the program and the state of things at any given time.

Commit your code to `/homework/cmsi387/dp` and tag it as `hw-0315`.

Sleeping Barber

Implement a solution to the sleeping barber problem (SGG Exercise 6.11) using the POSIX API, also following the pattern in the bounded buffer sample code. As with dining philosophers, use print statements to confirm correct behavior.

Commit your code to `/homework/cmsi387/sb` and tag it as `hw-0315`.

Exercises

Do the following exercises from SGG; submit your answers in hardcopy.

1. SGG Exercise 6.1 — Prove that Dekker’s critical-section solution for two processes (the first known correct software solution, ca. 1965) satisfies all three requirements for the critical-section problem.
2. SGG Exercise 6.17 — Show how the monitor implementation in the text would change if the `signal()` statement can only appear as the last statement in a monitor procedure (i.e., it behaves like a conventional return statement, in addition to doing its other activities).
3. Name two things that may happen in an *incorrect* critical section solution to the dining philosophers problem.
4. Name two things that may happen in an *incorrect* critical section solution to the sleeping barber problem.

Extra Credit (x2!!!)

That’s quite a chunk of work already, so this extra credit task truly deserves the name “extra credit,” and is worth *two* whole assignments:

Implement a *graphical* critical section solution of dining philosophers using Java/Swing/Java2D or C/GLUT/OpenGL. This graphical version should visually show the philosophers at the table, with their respective plates and chopsticks, and it should also show them thinking, getting hungry, waiting for and picking up chopsticks, and finally eating. If you do this, commit your code to `/homework/cmsi387/dp-gui`, with tag `hw-0315`.