Assignment 0405
This assignment is yet another “classic” operating systems programming task dealing with process synchronization, supported by a smattering of exercises.

Outcomes
This assignment will affect your proficiency measures for outcomes 1c, 1d, 3e, and 4a–4f.

Not for Submission
• If you haven’t done so already, read SGG Chapters 3–5.
• Read Chapter 6 of SGG and the synchronization papers given out in class.

For Submission
The Dining Philosophers Problem
Implement a solution to the dining philosophers problem using POSIX threads and semaphores. The bounded buffer sample code may be used as a basis for your solution. In addition, Sections 6.6.3 and 6.7.2 provide outlines for solving the problem.

Make sure to include well-placed output statements to report what’s happening in your program and the state of things at any given time — that’s how we’ll know whether your solution is working. Include code that announces, loudly, when the rules are violated. The assert function will be useful here.

Commit and push your code to your git repository under homework/dining-philosophers.

Exercises
1. Answer SGG Exercises 5.1, 5.2, 5.5, 5.11, 5.12, and 6.27.
2. Name two things (for a total of four observations) that may happen in incorrect critical section solutions to the dining philosophers and sleeping barber (SGG Exercise 6.39) problems. Write up your answers in any widely-readable file format. Commit and push your answers to homework/dining-philosophers/doc.

Extra Credit
Dining Philosophers in Java and the Sleeping-Barber Problem
You will get automatic + proficiencies in the assignment’s outcomes if you also:
• Successfully implement a Java-based solution to the dining philosophers problem using Java’s synchronized declaration (i.e., monitors). Commit and push this to homework/dining-philosophers/java.
• Successfully implement a solution to the sleeping barber problem (SGG Exercise 6.39). Commit and push this to homework/sleeping-barber.