

# CMSI 387-01

## OPERATING SYSTEMS

Spring 2013

### Assignment 0411

This assignment is yet another “classic” operating systems programming task dealing with process synchronization, accompanied by a trip through some seminal computer science literature.

#### Outcomes

This assignment will affect your proficiency measures for outcomes *2d* and *4a–4c* (The Dining Philosophers Problem), *3b* (All of Time and Space) and *4d–4f* (both).

#### 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

##### All of Time and Space

Continue our journey through seminal operating systems literature by using library resources (digital or otherwise) to find at least three additional publications (also digital or otherwise) that are related to and/or build upon the work given out in class. For each found publication:

- State its connection to the other work in our small collection—the connection must be explicit (i.e., cited). Connections may be to the other work that you find, but at least one must point to something given out in class.
- State what it does new or differently from the prior cited works.
- Give a brief assessment of the quality of the work. Consider the authors, the publication venue, the level of detail, references cited, etc.

You are not expected to fully understand everything in every piece that you find, but should have *some* kind of idea.

Commit and push this mini-annotated bibliography in any widely-readable format to *homework/os-literature*. Provide full citations for your found work. If you can supply full-text copies of these items, that would be great also.

##### 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 of SGG, as well as all kinds of other sources on the Internet, 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*.