Assignment 0422
Alright, back to some programming. For address translation, starter code is provided on the course website so that you can just fill in the blanks.

Outcomes
This assignment will affect your proficiency measures for outcomes 2d and 4a–4f.

Not for Submission
If you have the SGG textbook, you can re-read Chapter 6 for a deeper treatment of synchronization. Details on deadlocks can be found in SGG chapter 7, and memory management is covered by SGG chapters 8–9 in greater detail.

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 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.

Paged Memory Address Translation
Implement logical-to-physical paged memory address translation. To get you started, test harness and header files have been provided on the course website. Just fill in the // TODO comment.

“Extra” Credit: Page Replacement
You will get an additional column for outcomes 2d and 4a–4f for each virtual memory page replacement algorithm that you are able to implement. As with the paged address translation program above, test harness and header files have been provided on the course website, included suggested data structures and // TODO functions:
1. First in first out (FIFO)
2. Least recently used (LRU)