CSCI 3323 (Principles of Operating Systems), Fall 2021
Reading Quiz 6

Credit: 20 points.

1 Reading

Be sure you have read, or at least skimmed, Chapters 25, 26, 27, 28, 30, 31, 32, and 34 of the textbook. (Yes, I'm skipping chapters 29 and 33.)

2 Instructions

Answer the questions below using *only* the course textbook — no Web searches. It's okay to talk to classmates about this assignment as you usually do, *but* I want each person to do all the reading. Include the Honor Code pledge in what you turn in, either the full pledge or just the word "pledged". (Please put this in the same document as your answers, so I don't overlook it.) For these quizzes by doing this you are also saying you have at least attempted all the reading it covers.

You may write out your answers by hand and scan them, or you may use a word processor or other program, but please submit PDF or plain text in the "turn-in" folder I set up for you on Google Drive. (So, no word-processor files and no links to other Google Docs. This is a change from how I've asked students to turn in work in previous semesters, meant to reduce both the chance of mistakes on my part and the amount of time I spend managing multiple file formats.)

3 Questions

1. (2.5 points) What are two reasons to use threads in a program?
2. (2.5 points) For locks as described in the textbook, what happens if a thread calls `lock()` and some other thread holds the lock?
3. (2.5 points) Can locks be implemented with just a shared variable assuming the existence of the right machine instruction(s)? If so, what are some names of instructions that make this possible?
4. (2.5 points) How well do spin locks meet the textbook's criteria for locks: Do they guarantee mutual exclusion? How about fairness? And how well do they perform, or does it depend on the situation?
5. (2.5 points) What happens when a thread calls `signal()` on a condition variable nothing is waiting on? is it saved for the next time a thread does a `wait()` on the condition variable?
6. (2.5 points) How do you inspect the value associated with a semaphore? (*Hint:* This might be a bit of a trick question.)
7. (2.5 points) We've mentioned "deadlock" as something that can go wrong with a concurrent algorithm, and the textbook discusses it at some length in the chapter on common concurrency problems. What's the classic example of a deadlock?
8. (2.5 points) What does the textbook give as advice for avoiding deadlocks?