CPSC 3125, Operating Systems

Examination 3

October 27. 2021

1 Instructions

The CSU Honor Code applies to this examination. You may not use your textbook, your notes, or any other written or electronic material, including the internet. You may not give or receive aid from another person. Violations of the Honor Code will result in the failure of this examination and dismissal from the course. Submission of your answers to this examination constitutes an acknowledgment of the Honor Code and your compliance therewith.

Please answer the following questions. Please answer in complete sentences. Please answer all questions briefly and succinctly — no long essays permitted. Do not use short answers unless the question specifically calls for short answers. You must exhibit understanding of the material covered by the question.

Submit your answers in accordance with the instructions given orally in class. Late submissions will result in failure of this examination.

2 Questions

1. We will be discussing *parallelization* quite a bit. How would you explain parallelization to your grandmother? What specific kind of task would be a good candidate for parallization?
2. What might be some of the problems when a multi-threaded program returns indeterminate results? There are at least two specific ones mentioned in this chapter. Please be specific in your answer.
3. Briefly define the goals of locking: *mutual exclusion*, *fairness*, and *performance*.
4. Explain the difference between *coarse grained locking* and *fine grained locking*.
5. What does it mean to say that a data structure is *thread safe*?
6. Give a general description of the producer/consumer problem? Give one real life example of a producer/consumer application.
7. How can we use a *binary semaphore* as a lock?
8. How can a semaphore be used to mimic a condition variable?
9. The book states that there are four conditions for a deadlock. For each of these, explain why the condition prevents a deadlock, and how the absence of the condition can contribute to a deadlock.
   (a) mutual exclusion
   (b) hold-and-wait
   (c) no preëmption
   (d) circular wait
10. Why are no locks needed for an event-driven program running on one CPU?