CSCI 315 Operating Systems Design
Midterm Exam 2 Study Guide

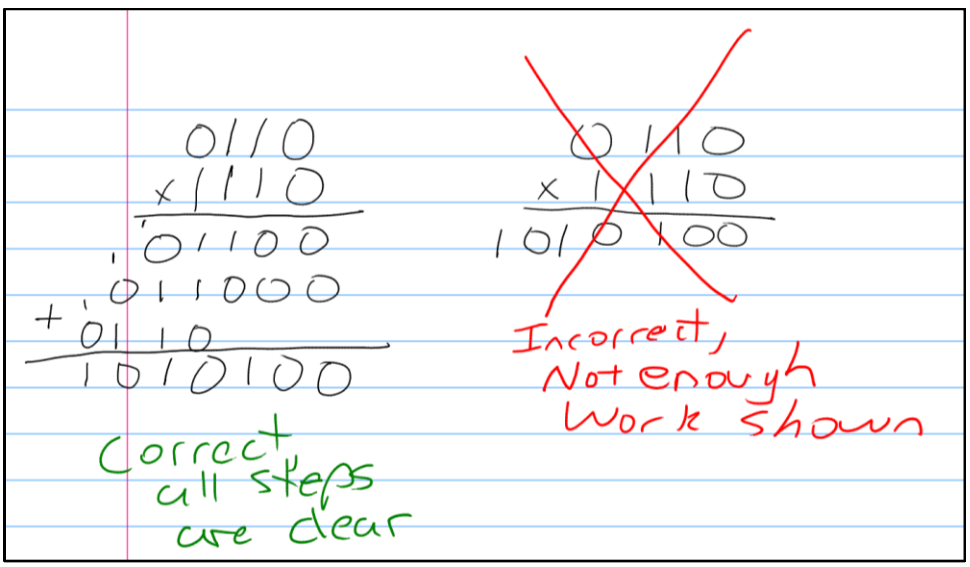
Fall 2020

Exam Instructions

This exam is timed. You must complete the exam in a 180-minute duration for a 52-minute exam. The timer will start when you click the button and download the exam file. The exam will be made available on October 30th, Friday ~~10:00~~ **9:00** am and due at November 3rd,Tuesday ~~10:00~~ **9:00** am.

We will be using the same setting as we did in our first exam, the Moodle Timed Quiz.

The exam will be open notes and open book with a specified amount of time to complete. As such, when there is any calculation, or code comprehension, or writing of any code in the exam, you must show the detailed steps you followed to arrive at your answer. For example, below shows two ways to compute 6 x 14 in binary. One will receive full credit, the other will receive no credit. When in doubt, error on the side of showing more work than you think you need. (This is a random sample question, not CSCI 315 question.)



You must work on your own. Do not discuss the exam with anyone else. The instructor will be available online to answer any questions.

Exam Study Guide

- Review all assigned readings from the textbook and external readings.
    - SGG (10th edition): 8.1 through 8.7, 9.1 through 9.5, 10.1 through 10.8
    - SGG (9th edition): 7.1 through 7.6, 8.1 through 8.6, 9.1 through 9.8

- Go through labs, quizzes, and activities. Make sure that you have a solid understanding of the topics they address. The relevant C programming is an important part of this phase of the course. While we do not expect students to memorize the details of system calls and functions, the concepts and their basic C syntax are required.

- This document doesn't mean to give an exhaustive coverage of what might appear in the exam, but it will be useful as a self-check list for your preparation.

---

1. Identify the following concepts.
    - deadlock
    - resource allocation graph
    - necessary conditions for deadlock
    - deadlock prevention and deadlock avoidance
    - safe state in the context of deadlock
    - safe sequence in the context of deadlock
    - first-fit, best-fit, worst-fit
    - internal fragmentation, external fragmentation
    - frame, page
    - physical address, logical address
    - TLB
    - page table, hierarchical page table
    - effective access time (EAT) in page-based memory management
    - virtual memory
    - demand paging
    - page fault
    - copy-on-write
    - swapping
    - thrashing
    - memory mapped file
    - valid bit, dirty bit, reference bit
    - allocation of frames (global vs. local, fixed vs. priority)
    - effective access time in virtual memory system
    - Belady's anomaly
    - FIFO page replacement algorithm
    - optimal page replacement algorithm
    - LRU page replacement algorithm and its various approximations
    - memory compressions
    - kernel memory

2. What are the four necessary conditions for a deadlock to occur?
3. Given a set of processes, resources, and resource requests, draw the resource allocation graph after each request and state whether or not a deadlock is possible.
4. In general, a cycle in a resource allocation graph indicates only the possibility of a deadlock. Under what special condition does it indicate the existence of a deadlock?
5. What are the different ways of dealing with deadlock?
6. How does the **safety** version of the Banker's algorithm work? You should be able to carry out a hand execution of the algorithm.
7. In contiguous memory allocation schemes, it is possible to use first-fit, best-fit, and worst-fit strategies. State a justification for each of these strategies. Is there one that works better or worse than the others? Explain your answer.
8. Identify the circumstances in which internal fragmentation and external fragmentation happen.
9. What are the advantages and disadvantages of memory management schemes such as: overlays, swapping, and virtual memory? What is the impact that each of these schemes have on the usability of the system (from a programmer's perspective) and on the implementation of the system?
10. In the context of a paging system, what is a logical address? What is a physical address?
11. Describe what is a *Translation Lookaside Buffer* (TLB), what data it contains, and how it is used.
12. Describe the impact that the use of a TLB can have on the performance of a virtual memory system.
13. Explain how a physical address is determined in a paged system using a TLB. Be specific about what the values are and how they are used.
14. Given the logical address in a virtual memory system, describe how it is translated to a physical address when there is: a single page table, a hierarchical page table.
15. Discuss the pros and cons of using a single page table and a hierarchy of page tables.
16. Consider a paged system where addresses are byte addresses, and pages consist of 16 4-byte words. If the desired page is in frame 5, and the offset is byte 4 in the frame, what is the corresponding physical address?
17. In a system with a TLB, assume the following:
    a. the memory access time is 150 nsec.
    b. the TLB access time is 25 nsec.
    c. the TLB hit rate is 80%.
    Computing the effective access time for memory in this type of scenario.
18. What are the motivations for using virtual memory in an operating system?
19. Identify the benefits of using virtual memory in a multi-user, multi-programmed operating system.
20. Identify the steps the OS takes in handling a page fault.
21. Describe a mechanism by which one frame of physical memory can belong to the logical address space of multiple different processes.
22. Construct a scenario in which it is better to pre-load all the pages of a process than to allow pages to be loaded on demand.