CSCI 315 Operating Systems Design
Final Exam Study Guide

- The final exam is cumulative. In order to prepare thoroughly for this exam, you should review the material for the two midterm exams (see their respective study guides).
- Review all assigned readings from the textbook.
- Go through labs assignments. Make sure that you have a solid understanding of the topics they address. The concepts in C systems programming covered in labs and project are also important for this exam.
- 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. Be able to explain the following concepts.
    - partition
    - file
    - file system
    - directory
    - attributes of a file
    - contiguous allocation of file space
    - linked allocation of file space
    - indexed allocation of file space
    - file allocation table
    - inode
    - FCB
    - PCB
    - free space management linked list)
    - polling
    - interrupt
    - DMA
    - symbolic link
    - hard link
    - process
    - thread
    - atomic operations

- deadlock, livelock
- spinlock
- internal fragmentation, external fragmentation
- physical address, logical address
- TLB
- page table, hierarchical page table
- Belady's anomaly
- valid bit, dirty bit
- thrashing

1. File system organization: compare the pros and cons of single-level, two-level, tree, acyclic, and generalized graph structures.

2. In the context of UNIX file systems, what do the file attributes for *owner*, *group*, and *other* express? How do they relate to protection?

3. How are numeric values of user id and group resolved when one runs **`ls`** to see the attributes to a file?

4. How does the directory represent a file, a symbolic link, and a hard link?

5. What problems can occur if links to directories are allowed in a directory structure?

6. List three methods for space allocation for files, and explain for each how the operating system keeps track of the blocks in a file. What are the advantages and drawbacks of each method?

7. Given the clear description of how the space for a file in secondary storage is represented, determine the maximum file size in that specific file system.

8. Consider the diagram for the "combined system" for file space allocation used in Unix. Describe what motivated the addition of the inode fields "direct blocks", "single indirect", "double indirect", and "triple indirect".

9. Describe the motivation for the creation of "log based" file systems.

10. How can free space for a file system be organized? Explain the advantages and disadvantages of the methods we have studied.

11. Explain how the interrupt system works by giving an example of what happens when a device is ready to transmit data to the CPU. Be specific and explain the steps involved.

12. How does *direct memory access* (DMA) affect the performance of a data transfers in a computer system? How does it compare to CPU-managed I/O?

13. Describe the sequence of steps for a DMA data transfer.

14. Explain how abstraction and encapsulation are involved in the design of the operating system.

15. Explain the how the elements of the storage hierarchy fit together (physical memory, cache memory, virtual memory, mass storage) and how each level contributes to computing in general.

16. Identify the circumstances in memory and secondary storage in which internal fragmentation and external fragmentation happen.

17. 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?

18. Describe what a*Translation Lookaside Buffer* (TLB) is and how it used. Describe the impact that the use of a TLB can have on the performance of a virtual memory system.

19. What are the motivations for using virtual memory in an operating system?

20. Describe the "big picture" purpose of an operating system.

21. Explain how each of the following scheduling algorithms works. Be able to discuss how each one affects CPU utilization, job throughput, turnaround time, waiting time, and response time. Discuss how preemption might apply to each of the algorithms and what effects it might produce on the performance of the system.

- first come, first served

- shortest job first
- shortest remaining time first
- round robin
- priority
- multilevel queue

18. How does the use of *preemption* and *priority levels* affect the behavior and the performance of CPU scheduling algorithms?

19. What is the difference between *user mode* and *kernel mode*? How does the system keep track of which mode it's operating in?

20. Enumerate five different methods for interprocess communication. Given various methods for interprocess communication, match them agains scenarios when they are more appropriate for use in a program.

21. List the states that a process may be in. Draw a diagram showing the possible transitions between theses states and identifying what causes state transitions.

22. From the perspective of the operating system, explain what is involved in the creation of a UNIX process (that is, what data structures are used, what is the sequence of operations in the creation process). Identify what data structures in the system are necessary to manage this process through its lifetime (with regards to process scheduling and management, memory management, and file system management).

23. Given the man page of a given system call or library function, identify what it does, what parameters it receives, what return values it produces, and how it indicates errors to the programmer.

24. What happens during a process context switch? Why can frequent context switches be a problem?