CSCI 315 Operating Systems Design
Final Exam Study Guide

- The final exam focuses on the latter part of the material, but it is also cumulative in the sense that we cannot sidestep some topics that appeared in the midterm exams.
- Review the textbook chapter guides.
- Review the in-class activities.
- Go through labs assignments since the second midterm. Make sure that you have a solid understanding of concepts and topics.
- 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.
    - partition
    - block
    - file
    - file control block
    - file system
    - directory
    - attributes of a file
    - contiguous allocation of file space
    - linked allocation of file space
    - indexed allocation of file space
    - Unix "combined scheme" or inode
    - free space management (via bitmap, linked list, grouping, and counting)
    - buffer
    - symbolic link
    - hard link
    - software cache

2. File system organization: compare the pros and cons of single-level, two-level, tree, acyclic-graph, and general graph structured systems. Identify use cases where each of these structures is desirable.

3. In the context of Unix file systems, what do the file attributes for *owner*, *group*, and *other* express? How do they relate to protection? How do you manipulate these bits using the `chmod` command?

4. How does Unix convert the numerical identifiers for *owner* and *group* stored in an FCB (inode) to human readable strings?

5. Why doesn't the publicly readable file `/etc/passwd` show user passwords? Why is the file `/etc/shadow` made unreadable to all users except for the system administrator?

6. What problems can occur when links to directories are allowed in a directory structure?

7. What is the difference between *hard links* and *symbolic (soft) links*?

8. Compare the methods we studied for space allocation for files. What are the advantages and drawbacks of each?

9. Compare the following methods for organizing the free blocks in a disk: bitmap, linked list, counting, and grouping.

10. Consider the data that is stored in a disk to implement free space management using bitmap, linked list, counting, and grouping. Explain how much data space is required for each method and whether that space can be considered *overhead* (in the sense of taking up space that might otherwise be used for data storage).

11. Consider the following statement: *"The Unix inode offers a structure that serves equally well small and large files."* Does the inode impose limits on file size? Does the inode use much overhead for small files? For very large files?