Review for CS-450 and CS-550 Mid-Term Exam

Fall 2005

Below are the principal areas of coverage that we have had so far, through last Tuesday's class. If you know the listed material well, you should get a high score on the exam.

NOTE: Page numbers for Nutt's text and detailed chapter/section identifiers are not provided. You should be able to look those up for yourself.

1. Which design goals for an Operating System are most important in which computer environments?

2. Sequence of historical progression of OS concepts and of memory layout in conjunction with hardware developments, and critical OS terms.

3. The four principal functions implemented by the kernel of the Operating System: Process, Thread, and Resource Management, Device Management, File Management, and Memory Management (Nutt fig. 3.10).

4. Multi-Programming/Multi-Tasking and Multi-Processing: understanding what each concept means and what are the differences between them.

5. Concurrency and Simultaneity: understanding what each concept means and what are the differences between them.

6. The differences between Hard- and Soft-Real-Time environments and the constraints that they impose upon system performance.

7. Review of hardware, principal events in boot sequence, and understanding both why and how is the modern OS event- (i.e., interrupt-) driven. The Fetch-Decode-Execute Cycle and its augmentation to enable the recognition and handling of interrupts.

8. The processing of interrupts and of traps; the difference between a trap and an interrupt.

9. Storage hierarchy, storage devices, including especially disk drives and magnetic tape, and Main Memory.

10. Privileged instructions, processor modes and the control of processor mode, and the role of mode change in the servicing of system calls and interrupts.

11. Groups of functions provided by the OS; and services provided by the OS. The Application Program Interface (API), and the Command-Line Interpreter.

12. System calls and their execution sequence.

13. Device-independent and Device-dependent portions of the Device Manager; in particular, mechanism for incorporating reconfigurable device drivers (Nutt's Figure 5.8-5.19).

14. Sequencing of execution of disk accesses: First-Come-First-Served, Shortest Seek-Time first, Scan, Look, Circular-Scan and Circular-Look. **In particular, be able to figure out, for a given sequence of track access request, how the request sequence would be executed under each algorithm.**

15. The Job or Process, possible process states, permissible state transitions.

16. Contents of the Process Control Block (PCB) and of the Process Image for single-threaded processes.

17. The sequence of events in Context-Switching.

18. Creation and termination of processes; parent processes and child processes: *UNIX* fork(), wait(), and join()

19. Scheduling, especially three kinds of Process Scheduling (Long-Term, Medium-Term, and Short-Term, which kind(s) **must** an operating system have and which are present in only some, but not all, operating systems.

20. Threads, thread structures, relationship between multiple threads belonging to a single process, advantages of multiple threads for a single process, and especially the differences between User-Level threads and Kernel-Level threads.

21. CPU (short-term) scheduling, goals of CPU scheduling, circumstances under which the short-term scheduler is invoked, and operations of the scheduler.

22. Various algorithms used for CPU scheduling: First-Come First-Served (FCFS), Shortest Job First (SJF, both its preemptive and its non-preemptive variants), Round Robin Scheduling, Priority

Scheduling (both preemptive and non-preemptive), Multi-Level Queue Scheduling, and the prediction of next CPU-burst time (simple arithmetic averaging of all previous bursts, assumption that next burst time will be identical to previous burst time, and at least a qualitative (but NOT a quantitative) understanding of exponential averaging). **In particular, be able to trace out the execution of various processes in accordance with the various CPU-scheduling algorithms.**

23. Two hardware synchronization approaches: Test-and-Set, and Swap or Exchange. Also, OS-provided semaphores and their use, in particular the counting semaphore.

24. Deadlock and starvation, **the four conditions necessary for deadlock to occur, and the four strategies for dealing with deadlock (Deadlock Prevention, Deadlock Avoidance, Deadlock-Detection-and-Resolution, and Disregard-for-the-Possibility-of-Deadlock).** The banker's algorithm.