

CSCI 3323 (Principles of Operating Systems), Fall 2017

Review for Final Exam

1 Format of the exam

The exam will be at the scheduled time for the course final, December 8 at noon. The exam will be about twice the length of the first exam (intended to take 75 minutes), but you can use the full three-hour period if you like. As is usually true for my exams, it is “open book / open notes”, which means you can consult paper or electronic copies of the textbook and your notes, sample solutions *from this year only*, your own graded work, and anything on the course Web site. You may not use other books, materials from this course from previous years, a calculator or computer (except as needed to consult allowed sources), or (of course) each other’s papers. As with the midterm, some questions will be similar in format to those in the homework assignments; others will be multiple-choice or true/false. You might be asked to write or comment on code/pseudocode, but it’s unlikely that you’ll be asked to write more than a few lines. There will be some questions on material from the first part of the course (before the midterm), but the emphasis will be on material from the later part of the course.

2 Lecture topics to review

You are responsible for all material presented during lecture; the following is a list of major topics to review:

- History/evolution of operating systems.
- Functions we would like an operating system to provide.
- What functionality we need (and/or can expect) from hardware in order to write a reasonable operating system.
- System calls.
- The process abstraction.
- Implementing processes (what’s involved in a context switch, what kinds of things are in a process table, etc.)
- Processes versus threads.
- Why interprocess communication (IPC) is needed.
- Mechanisms for IPC (shared variables, semaphores, monitors, message-passing).
- IPC problems (mutual exclusion, producers/consumers, dining philosophers) and their solutions.
- Scheduling and scheduling algorithms.
- Deadlocks (what they are, strategies for avoiding them).

- Address space abstraction; virtual (program) addresses versus physical addresses.
- Schemes for managing memory (multiprogramming with variable partitions, paging, segmentation (very briefly)); advantages and disadvantages of each; implementation details at the level of the homework problems.
- “Page faults” (what they are, how they’re handled).
- Page replacement algorithms (what they’re for, which ones work well and why).
- Filesystem abstraction (files, directories, file types, access methods, etc.).
- Basics of filesystem implementation (ways of allocating space for files and their advantages/disadvantages, disk-space management, reliability issues).
- Basics of I/O hardware (devices, device controllers, I/O ports versus memory-mapped I/O, DMA).
- Goals of I/O software.
- Basics of I/O software (programmed I/O versus interrupt-driven I/O versus I/O using DMA).
- I/O software layers and how they work together.
- Security (goals, a little about ways in which systems can be attacked or compromised).

3 Reading to review

You should have read all of chapters 1, 2, 3, 4, and 5, and at least skimmed chapters 6 and 9. Probably the best way to approach reviewing the reading is to skim all of it, paying more attention to topics I covered in class, and (re)read the last (summary) section of each chapter.

4 Other tips

You should also be sure to review all homeworks (and sample solutions) and any non-opinion minute essay questions.