Slide 1

Administrivia

- Reminder: All homeworks, reading quizzes past due. By no means too late to get at least partial credit, *as long as you haven't looked at sample solutions*.
- Sample solutions available now for homeworks; sample solutions for reading quizzes coming soon.

  I'm making some headway on grading, and hoping to get more done over the weekend.

Slide 2

More Administrivia

- Midterm to be available Monday; turn in by following Monday. Review sheet with rules/format and topics available on course Web site. (This is also today's topic.)
- You may well have questions you'd have asked if this lecture had been live. Please make a note of them and put them in your minute essay!

Slide 3

Midterm — Rules

- Open book / open notes. Not however "find answers wherever you like".
- No collaboration with others. Okay to ask me questions. I'm hoping to do better with office hours next week, but if not then I will be available by e-mail during those times.
- Timed, one continous block of up to 3 hours (though intended to take no more than 1.5 hours). Up to you when to do this during the designated week.
- You will start from a PDF (shared on Google Drive) and e-mail me a PDF (see review sheet for options). Note that you should *not* look at the exam until you're ready to actually do it.
- Yes, I'm trusting you a lot here! However, my feeling is that most Trinity students *are* trustworthy.

Slide 4

Midterm — Format

- Some multiple-choice and true-false questions. If you think a question is ambiguous you can explain why, and that can get you part credit.
- Some short-answer questions.
- Probably no writing of code or pseudocode, but very likely questions about pseudocode (e.g., "this code is meant to do X; does it work?").

Slide 5

Midterm — Topics

- History/evolution of operating systems.

  No detailed questions here, but maybe something about how operating systems grew from nothing to hugely complex, following similar patterns in mainframe, PC, etc. worlds.

- Functions O/S should provide.

  Again no detailed questions, though maybe something about two views (from programmer side, from hardware side).

- What's needed from hardware.

  What's needed to write an O/S that can defend itself, and protect each user/process from others?

- System calls.

  What's their purpose? How do they work?

Slide 6

Midterm — Topics, Continued

- Processes.

  Things that need to be stored on a per-process basis to make the "virtual CPU" idea work; how processes differ from threads.

- IPC.

  Different synchronization mechanisms — how do they differ both from abstraction side and from implemention side.

  Basic/classical problems (mutual exclusion) — how solutions work.

  You might be given pseudocode and asked to comment on it.

Slide 7

Midterm — Topics, Continued

- Scheduling and scheduling algorithms.

  No very detailed questions; mostly focus will be on conceptual understanding, such as when different algorithms are appropriate.

- Deadlocks.

  Again nothing very detailed, no more so than homework problem.

Slide 8

Minute Essay

- Questions?