

Slide 1

### Administrivia

- One purpose of the syllabus is to spell out policies (next slides).
- Most other information will be on the Web, either on my home page ([here](#), office hours) or the course Web page ([here](#)).

A request: If you spot something wrong with course material on the Web, please let me know!

Slide 2

### Course FAQ

- "What will my grade be based on?" (See syllabus.)
- "When are the exams?" (See syllabus.)
- "What happens if I can't turn in work on time, or I miss a class?" (See syllabus.)
- "What's your policy on collaboration?" (See syllabus.)

### Course FAQ, Continued

- “When is the next homework due?” (See “Lecture topics and assignments” page.)
- “When are your office hours?” (See my home page.)

Slide 3

Note that part of my job is to answer your questions outside class, so if you need help, please ask! in person or by e-mail or phone. Some office hours will be “open lab” (times TBA). At those times I’ll be in one of the classrooms/labs ready to answer questions.

### Why Is This Course Required?

- “ACM says so” (i.e., curriculum recommendations include course on operating systems). Why? Well . . .
- To be a “computer scientist”, need to have a broad understanding of computer systems — and operating system is a key part of a computer system.
- Knowing something about how operating systems work helps you write efficient code.
- Many of our courses “demystify” parts of computer systems (e.g., PAD I/II and Computer Design); so does this course.
- It might even be fun . . .

Slide 4

Slide 5

### What Is An Operating System?

- Definition by example?
- Definition(s) from operating systems textbooks?

Slide 6

### What Is An Operating System? Continued

- Definition by example:
  - Recent: Windows, Linux, UNIX, BeOs, OS X (Mac), . . .
  - Older: MULTICS, VMS, MVS, VM/370, . . .
  - (Also special-purpose o/s's for special-purpose hardware — e.g., video-conferencing system.)
- Definition(s) from operating systems textbooks:
  - Something that provides “virtual machine” for application programs and users (“top down”).
  - Something that manages computer’s resources (“bottom up”).
- Another view — key part of bridging gap between what hardware can do (not much, but very fast) and what users want.

### What The Hardware Can Do

- CPU: fetch machine instruction from memory, execute; repeat.
- Disk: read data from / write data to location on disk.
- And so forth — very primitive.

Slide 7

### What The Software Must Do

- Programs students usually write in PAD I/II:
  - Define and manipulate data structures.
  - Do arithmetic/logical calculations.
  - Read stdin / write stdout.
  - Call GUI/graphics library routines.
- The magic cloud (operating system):
  - Read from keyboard, write to screen.
  - Manage what's on screen — windows, taskbar, etc.
  - Run multiple applications “at the same time”.
  - Manage disk contents — files, directories/folders.
  - Share the machine with other users.

Slide 8

## Course Overview

- Brief history of operating systems.
- Review of what hardware can do, what operating system must/should do.
- Discussion of major functions of operating system — problem(s) to be solved, solutions:
  - Process management.
  - Memory management.
  - I/O management.
  - Filesystem management.

Focus on principles rather than details.

Slide 9

## Minute Essay

- What are your goals for this course?
- What operating systems have you used/installed/experienced?
- Anything else you want to tell me? about the course, what you did this summer, ...?

Slide 10