Slide 1

Administrivia

- Did you get a “this is a test of the class mailing list” e-mail message? If not, come talk to me.

Slide 2

Syllabus / More Administrivia

- One purpose of the syllabus is to spell out policies, especially about:
  - Course requirements and grading.
  - Exam dates (can only be changed if you all agree). “Please plan accordingly” means “don’t schedule something else for these dates”.
  - Late work.
  - Academic integrity.
- Most other information will be on the Web, either on my home page (office hours) or the “course Web page”.
- Part of my job is to answer your questions outside class. E-mail usually works well if office hours don’t.

Slide 3

A Few Words About Computer Use in Class

- Checking your e-mail when you first get here is okay.
- Taking notes online is okay.
- Surfing the Web or playing games during lecture is not okay — fun, but distracts you and maybe your neighbors.
- Remember that I can lock all screens, project what's on one student's screen, etc. — and I will if need be. But I'd rather start by assuming you're all responsible people who will do the right thing!

Slide 4

"Why Do I Have To Take This Course?"

- We could view computer systems (hardware/software) in terms of layers of abstraction:
  - User interface.
  - Operating system / application programs / tools (compilers, e.g.).
  - High-level programming language / ADTs.
  - Machine language / data representations ("it's all 1s and 0s").
  - Hardware (could break this down, maybe, into logical design and EE).
- A goal of a CS degree program is to "demystify" as many of these as we can.

Slide 5

“Why Do I Have To Take This Course?”, Continued

- Relating courses to layers of abstraction:
  - Programming courses — bridge gap between user interface and high-level languages.
  - Operating systems course — bridge gap between user interface / applications programs and hardware.
  - Course on compilers, maybe — bridge gap between application programs and machine language.
  - This course — bridge gaps between application programs and machine language (a bit) and between machine language and hardware.

Slide 6

Course Topics

- Defining and measuring performance.
- Assembly language — MIPS because it’s simple and representative.
- Machine language.
- Hardware (at level of AND/OR gates).

Slide 7

Why Study Assembly / Machine Language?

- Understand the general principles of how things work at this level helps you:
  - Write more efficient programs.
  - Understand operating systems (which also helps you write more efficient programs).
  - Generally understand better what's really happening in the machine.
- It might be fun.

Slide 8

Minute Essay

- What are your goals for this course?
- Based on what I said today and whatever else you know about this course, what do you think you will find most interesting/valuable about it?