



- The machine code part should depend only on the architecture (and 32-bit versus 64-bits counts as part of "the architecture"). But the same compiler running under different operating systems might make different choices?
- Format seems like it would be similar but not identical across operating systems, but if there's no real incentive to standardize maybe it hasn't happened.

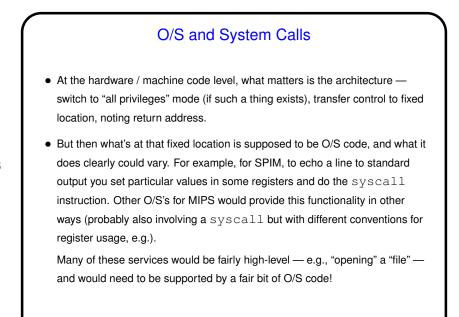
Slide 3

O/S Versus Executable File Contents/Formats

 Part of what's in an executable file is whatever information is needed for the O/S to "launch" the program. Windows .exe files don't run under Linux, right? and ELF executables don't run under Windows? (What about WINE? well, it's an emulator, isn't it?)

Slide 4

- So for example consider references to shared library code Windows DLLs versus UNIX "shared libraries" versus ...).
- Also might matter whether the linker can assume that programs will always be loaded starting at the same address.



Slide 5

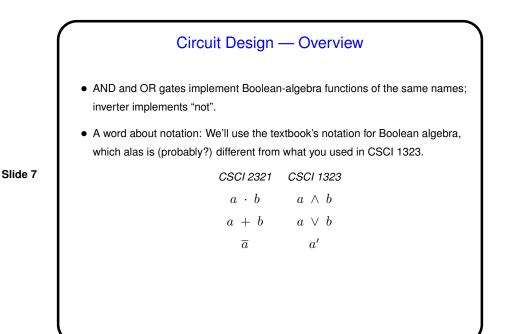
Designing a Processor — Overview

 Goal of Chapter 4 — sketch design of a (hardware) implementation of MIPS architecture in terms of some simple building blocks (AND and OR gates, inverters).

Slide 6

- Key components of the design (Figures 4.1 and 4.2):
 - Something to implement memory.
 - Something to implement instructions: "ALU" (arithmetic/logic unit).
 - Something to implement registers: "register file".
 - Something to implement fetch/decode/execute cycle: "control logic".

The first three together make up the "data path". Analogy — it's a puppet, with "control" pulling its strings.



Implementing Logic Gates — Executive-Level Summary
The ones and zeros of low-level software become two distinct voltages in

- The ones and zeros of low-level software become two distinct voltages in hardware, and the logic of Boolean algebra is implemented using "switches" (things that connect an input to an output, or not, depending on the state of a control input).
- Slide 8
- Currently these switches are (usually?) transistors. In widely-used "CMOS technology", there are two types of switches, one that's good if the input is "one" and one that's good if the input is "zero". These can be combined to implement logic. We looked earlier in the semester at a simple example (inverter). Link to description/explanation on "useful links" page coming soon.

