Slide 1

Administrivia

- Corrected versions of Homeworks 2 and 3 will be on the Web later today. I will send mail.
- Reading for (this past) Monday slightly revised — 2.16 and 2.17 are worth looking at but not required; similarly for 2.15 (on CD).

Slide 2

Minute Essay From Last Lecture

- With dynamic linking it's harder to track down errors? (I'm skeptical.)
- Executables can be really big? (Um, no? true *without* dynamic linking, no?) or library becomes part of executable, so updating library doesn't update all users? (again, no?)
- If you change the library API (remove a function, overload, etc.)? (Yeah. But surely no sane system . . .)
- Dynamic linking might involve some runtime overhead? (Maybe, but might be one-time startup cost.) or otherwise be inefficient? (Maybe.)
- Security risk? (Maybe.)
- (Also see my proposed answer, in the notes for last time.)

Slide 3

This and That

- In a previous class we mentioned “decoding” a machine-language instruction. Should we do an(other) example?
- Textbook presents extended example (sort). Skim as an example of using MIPS instructions.
- Textbook goes into some detail about compiling C code to loop through an array, showing a version that uses indices and one that uses pointers. Skim it as another example, but the take-home message is that whether C programmers need to consider such things — probably not with a sufficiently “good” (optimizing) compiler.
- Anything else we should talk about before moving on to the next chapter?

Slide 4

Minute Essay

- None — quiz.