Slide 1

Administrivia

- Reminder: Homework 5 due today.
- Homework 6 posted; due the week after spring break(!).

  Two problems, mostly straightforward practice working with files in C, plus some practice using command-line arguments.

  My intent is for you to use character-based I/O for one, "formatted" (`fscanf()`, `fprintf()`) for the other.

Slide 2

Minute Essay From Last Lecture

- Most people are either using `make` or at least compiling with `-Wall`. (I recommend the former. If it's not working for you, ask!)
- A few are not. Why would you pass up an opportunity to have the compiler help you find mistakes?

Slide 3

Strings in C — One More Point

- One of the video quiz questions about strings was about the pluses and minuses of representing strings as C does.
- Several people responded with something about having to write code to find a string's length. Do keep in mind that while you can do that (and I did, as an example of working with strings), there *is* a library function `strlen()`!

Slide 4

Recap of Video Lectures

- Files in C.
- Nothing very deep here, though how C deals with errors is different from how more-recent languages do.
- As with other languages I can think of, opening a file for output in C either creates it or overwrites it.
- One thing to maybe note is that C rather encourages reading a character at a time rather than a line at a time as many other languages do.
- Questions?

Slide 5

Files in C — One More Example

- As one more example, with a slight twist:

  Combine `echo-text.c` and `copy-file.c` so the program copies a file if two command-line arguments are given, otherwise echoes.

Slide 6

Practice Problem

- In UNIXworld there's a very old utility `cat` that just concatenates files given as command-line arguments, or echoes what you type if no command-line arguments are given. Write a program that works that way.

Slide 7

Minute Essay

- None really — just sign in, unless questions?