

Slide 1

Administrivia

- Midterm grade summaries e-mailed.

Slide 2

Minute Essay From Last Lecture

- Some people mentioned applications where you want the program to get a small amount of data (parameters for solving an equation, say), and for that files seem to me to be overkill *unless* you want to reuse the input data.
- Some others weren't sure how files could be useful for doing something they actually wanted to do. "Hm!"
- In my thinking — and several people picked up on this — one of the ways in which files are most useful are for working with large volumes of data. Not being an engineer I can't be sure, but perhaps you have situations where you have an instrument that collects a lot of data, which you want to somehow process or summarize? seems to me it makes sense to have the instrument produce a file, which you could write a program to process.
- One person said "homework problems". There is that!

Files in C — Recap/Review

Slide 3

- Files in C based on notion of “streams” (input and output). Sequence of bytes coming in or going out.
- Predefined streams `stdin`, `stdout`, `stderr`. Together with I/O redirection (in shell) these give you a crude way to work with files.
- Or you can create your own streams, connected to files, with `fopen`.
- Read and write with `fgetc`, `fputc`, `fscanf`, `fprintf`. All very much like functions you already know, but with one more parameter (stream).

Sidebar: gnuplot

Slide 4

- A tool I like for both quick interactive plots and nice-looking ones to use in papers is `gnuplot`. Available on most UNIX-like systems and (I think!) optionally for other operating systems. Home page at `gnuplot.sourceforge.net`. Can do 2D and 3D plots, the former with Cartesian or polar coordinates.
- To start it, `gnuplot`. Brings up a command-line interface. Online help available with `help`.

gnuplot, Continued

Slide 5

- Useful commands include `plot` to plot function(s) or data from file(s), `set` to set various things (e.g., x and y ranges).
- Default output to terminal, but with `set terminal` and `set output` you can instead store to a file in various formats.
- Can also put commands (`plot` etc.) in a file and execute batch-style, or with `load`. Useful if you want to regenerate plots when data changes.
- (Examples.)

Floating-Point Revisited

Slide 6

- One of the exam questions asked you about representing 0.1 in (binary) floating point. Only one person got it right. I think this may not have been a fair question, so in scoring the exam I counted it as an extra-credit problem. But why is the answer “can’t be done exactly”?
- Well ...

Numbers in Math Versus Numbers in Programming (Review)

Slide 7

- The integers and real numbers of the idealized world of math have some properties not completely shared by their computer representations.
- Math integers can be any size; computer integers can't.
- Math real numbers can be any size and precision; floating-point numbers can't. Also, some quantities that can be represented easily in decimal can't be represented in binary.
- Math operations on integers and reals have properties such as associativity that don't necessarily hold for the computer representations. (Yes, really!)
- (Two "floating point is strange" example programs.)

File Example(s) Continued

Slide 8

- (Time permitting.)
- Before the exam I mentioned writing a program to "detabify" input. This makes more sense operating on files than on input from stdin. So let's write that . . .

Minute Essay

- How did the midterm compare to your expectations? with regard to length, difficulty, etc.? If you didn't do well, why do you think that happened?
- What tool(s) do you usually use to make plots? (I asked this question in a CSCI class recently, and some students said they only rarely had to do plots. This group may be different?)

Slide 9