

Slide 1

### Administrivia

- (I had said in class previously that I wanted to have one more assignment, somewhat open-ended, but — no.)
- All work turned in to me has been graded, so we're pretty much done here except for attendance today. I just e-mailed you a summary of your grades with a somewhat tentative letter grade ("tentative" in that I'm uncertain about averages near grade boundaries and for now I'm estimating on the conservative side, and also in that the average doesn't include today's "attendance point").

Slide 2

### Minute Essay From Last Lecture

- Homework 7 — many people struggled, but at least one reported that it helped in understanding pointers. (A couple of people, however, found it not so tough. "Hm!?)
- (Might be worth noting that this group did much better with this assignment than students in previous years — but in previous years I've also had to rush through the needed background.)

### Full-Screen Text-Mode Programming

Slide 3

- As you know(?), the C standard library defines text input/output via “streams”, but these are line-oriented.
- But at least one text-mode programs you’ve used (`vim`) allows for other kinds of text I/O.
- How do they do that? various ways, none portable in the sense of working with all standard-conforming implementations of C, but the `ncurses` library is fairly standard in UNIXworld.
- (Examples on “sample programs” page.)

### Other Useful(?) C Libraries

Slide 4

- `readline` to provide command history and some simple editing. (Probably not standard C but probably widely available. `man readline` for some info.)
- “X11 library” for fairly low-level graphics programs. (I’m not finding a good man-page starting point, but your favorite search engine ...) Several third-party toolkits that build on it. Nothing completely portable, alas, but remember that C’s supposed to be implementable on a very wide range of hardware.

### Just For Fun — “Extreme” ASCII Art?

- Some of you may have heard of “ASCII art”? a truly over-the-top example, from quite a while ago, can still be found, via  
`telnet towel.blinkenlights.nl`  
(to interrupt control-] then “quit” or control-d — although this doesn’t seem to work in a terminal window??)  
(For a while recently the site wasn’t working. Seems okay now?)
- (What some people choose to do with their time can be — interesting?)

Slide 5

### Course Topics — Recap

- Basic C programming, for people who already know how to write programs in some other language. Especially useful (I think!) for those who start in a very abstract/high-level language.
- Review of the Linux/UNIX command-line environment and command-line development tools.
- Review of basics of computer arithmetic and data representation. A little more about floating-point representation.

Slide 6

## Why Learn C? (For Java/Python/Scala Programmers — Recap)

Slide 7

- Scala and Python (and Java, less so) provide a programming environment that's nice in many ways — lots of safety checks, nice features, extensive standard library. But they hide a lot about how hardware actually works.
- C, in contrast, has been called “high-level assembly language” — so it seems primitive in some ways compared to many other languages. What you get (we think!) in return for the annoyances is more understanding of hardware — and if you do low-level work (e.g., operating systems, embedded systems), it may well be in C. (Performance *may* also be better, though “measure and be sure”.)

## Quotes of the Day/Week/?

Slide 8

- From a key figure in the early days of computing:  
“As soon as we started programming, we found to our surprise that it wasn't as easy to get programs right as we had thought. Debugging had to be discovered. I can remember the exact instant when I realized that a large part of my life from then on was going to be spent finding mistakes in my own programs.” (Maurice Wilkes: 1948)
- From someone in a discussion group for the Java programming language:  
“Compilers aren't friendly to anybody. They are heartless nitpickers that enjoy telling you about all your mistakes. The best one can do is to satisfy their pedantry to keep them quiet :)”

## Minute Essay

- None really — just sign in (unless you have parting remarks?).

Slide 9