

Slide 1

Administrivia

- Everything graded except Homework 9. Grade summaries mailed; they should give you an idea where you stand in the course.
- Reminder: Homework 9 due next week.
No homework accepted past May 14. (You can still get partial credit for all assignments, as long as you haven't looked at sample solutions.)
- Sample solutions posted for all homeworks (except of course Homework 9). Might be worth a look even if you got full credit for an assignment.

Slide 2

More Administrivia

- About getting help with Homework 9 (and others, if you're behind):
- ACM tutoring officially ends on the last day of class, though you can ask whether someone can meet with you later.
- I'm tentatively planning to have some office hours next week (times TBA by e-mail). Also I could do "virtual office hours" via Google Hangouts.
And I do respond pretty well to e-mail!
- Questions?

Homework 8 Essays

Slide 3

- Several people said this one was somewhat more straightforward than others. I tend to agree!
- Several people thought their solutions for `update_board()` seemed long and cumbersome. The simple dumb way *is* long. (But sometimes the simple dumb way to do something is not bad.) Sample solution has two shorter ways.

4/24 Video Quiz Answers — Traps for the Unwary

Slide 4

- Most common: Trying to access out-of-bounds array elements.
- Other undefined behavior, including arithmetic overflow.
- Trying to translate Scala into C too directly.
- Arrays not being initialized to zero.
- Problems working with strings (e.g., forgetting null terminator).

4/24 Video Quiz Answers — What Else in C?

Slide 5

- Interacting with hardware:
"Memory-mapped I/O" by setting pointer value explicitly. (Non-standard but sometimes you have to do that.)
- GUIs, graphics, multi-threading / actors . . .
Probably easier in other languages, but possible with libraries outside standard.

4/24 Video Quiz Answers — Uses for `ncurses`

Slide 6

- Simple games?
Some not too hard, though tedious. Example: Conway's Game of Life.
Others trickier; basically anything involving multithreading. Example: Tetris.

Slide 7

4/24 Video Quiz Answers — Multithreading

- Most people in CS2 had some exposure, via parallel collections, actors, and futures. Parallel collections similar to OpenMP; actors and futures may be trickier in C because they involve asynchronous behavior.
- One person using it in Matlab for research project.

Slide 8

4/24 Video Quiz Answers — This and That

- C more understandable than Scala: I tend to agree! With C I can imagine how a very simple compiler could work; with Scala, too much seems like magic.
- What programs would work better in C than in some other language?
Probably low-level stuff that needs to interact with hardware more directly. Classic example is device drivers, other parts of operating system. Often must be non-standard / non-portable.

Just For Fun — “Extreme” ASCII Art?

Slide 9

- 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?)

Quotes of the Day/Week/?

Slide 10

- 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 :)”

Course Topics — Recap

Slide 11

- 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.

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

Slide 12

- 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”.)

Minute Essay

- None really — just sign in.
- Best of luck with your finals, and best wishes for a good summer!

Slide 13