

Administrivia

- Comments in code for sorted-linked-list example and provided code for Homework 6 updated (improved I hope!).
- Homework 6 due date revised to May 12. Not accepted late. That is also the “not accepted past” date for revised versions of all homeworks.
- Homework 3 grades mailed today; sample solution posted. Additional grades and sample solutions coming as I have them.

Slide 1

More Administrivia

- I plan to have office hours during reading days and finals, but not every day. I will send e-mail when I've settled details.
- In some courses I put together a last-minute extra-credit assignment. For this course I usually don't but would consider doing so. More about that at minute-essay time.

Slide 2

One More Example — C Really Does Vary

- I've been saying that C isn't necessarily the same everywhere — some flexibility in implementation.
- Sample program `sizes.c` illustrates this — run on a 32-bit system (`hardy`) and a 64-bit system (any of our other Linux machines) and compare output.

Slide 3

Sorted Linked List Example and Homework 6

- Example code is meant to be a fairly straightforward recursive implementation of a basic linked data structure, expressed in C; only the details of how to do this in C should be novel or difficult.
(One thing that might need more explanation is the choice of parameters for the print function. The idea is to let the caller decide some details — print to standard output or a file? all on one line or each on its own line?)
- Homework 6 is meant to give you a chance to try to something similar for binary search trees, and the example is meant to be helpful as a possible model.

Slide 4

Grades

Slide 5

- Most of grade based on homeworks, with a few points for attendance.
- If you've turned in all the homeworks, more or less on time, and your code compiles and passes your tests, and you've attended class, you will likely make an A.
- If you've turned in all or most of the homeworks, but some of them didn't work, you're welcome to submit revised versions of anything I haven't graded yet. I'd rather grade working code!
- If you didn't turn in a homework, it's not too late to get *some* points (maximum of half credit, but better than zero!).

Course Topics — Recap

Slide 6

- 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 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 — sign in. Also tell me whether you're interested in an extra-credit assignment.

Slide 9