Class 01: An Introduction to the Course

Held: Monday, 24 January 2011

Summary: We begin the class with a short overview and then a short lab.

Related Pages:

- EBoard.
- Lab: GNU/Linux I.

Notes:

- Welcome to CSC 161.
- Many of you are prospective CS majors. Let me know if you want to be added to the CS students mailing list, which includes summer opportunities and such.
- Some of you struggled in CSC 161. We're setting up small tutoring groups now. Let me know if you'd like to be in one. (I'll also try to email you if I think you might want to be in one.)
- Due tomorrow morning: Assignment 0 (an introductory survey).
- Due Wednesday evening: Assignment 1 (Linux stuff that follows up on the first three days of class).
- This Thursday at 4:15: Microsoft on opportunities.
- Next Thursday at 4:30: Summer opportunities in CS.

Overview:

- Course goals.
- Texts, required and otherwise.
- A few syllabus notes.
- Lab 1: GNU/Linux 1.

Course Goals

- Learn a new paradigm of programming: *Imperative Programming*
  - Key operations: assignment and explicit sequencing of actions.
- Learn some lower-level concepts in computer science.
  - Memory management.
  - Data representation.
- Learn C
  - Small and powerful
  - A nearly ideal language for learning the prior concepts
  - A syntax that has been copied way too often
  - There's a reason it's survived thirty years.
- Learn some tools that every CS major should know.

    - Linux
    - Emacs
    - Make
- Learn a bit of the philosophy of Unix
    - There's a reason it's survived thirty years.
- Revisit some key data structures and algorithms from this new perspective
    - Lists
    - Dictionaries
    - Searching and sorting
- And more!

Think like an old-school computer scientist.

Textbooks

- Kernighan and Ritchie: *The C Programming Language*
    - The first book written on C (and, for awhile
    - Reflects the language: Small, elegant, powerful.
    - Treats you like an adult. (For better or for worse.)
    - Helps you think like a real C programmer.
    - Something you'll want to keep.
- Mecklenberg: *Managing Projects with GNU Make*
    - Much more information than you'll need about Make.
    - Available free online.
- Raymond: *The Art of Unix Programming*
    - Helps you think like a Unix programmer.
    - Available free online.
- Harbison and Steele: *C: A Reference Manual*
    - Classic reference on the language.
    - Optional.
    - Copies will be left around labs.
- Plauger: *The Standard C Library*
    - Classic reference on the library.
    - Optional.
    - Copies will be left around labs.

More Administrivia

- I won't discuss a lot of the syllabus because
    - students rarely remember syllabus detail
    - you'll look at it in the introductory survey
    - it's all on the Web anyay.
- Homework is usually due on Wednesday evenings at 11:00 p.m.

    - Guess why.
- Three take-home exams, one optional final.
- Ask questions on readings (nightly)
- Participate in class!

Lab

- Attempt the lab.
- Be prepared to reflect.
