What’s CSE 240 All About?!

The “new” CSE240:
• Not at 9am!!!
• New approach (bottom up)
• New textbook
• New instructor
• Well integrated into curriculum
• Cooler assignments (demo coming later)
• No assumption that you know C programming
• Third time around

Big Picture
Introduction to computer architecture
• How is data represented?
• What are the pieces of a computer?
• How do computers work?

Programming
• How do I “talk” directly to the machine?
• How do I program in “C”?

Computer systems and computation
• How do simple HW/SW elements come together to realize complex computations?
Course Components

Part 1: Hardware
- Representing data, transistors, gates, digital logic structures, von Neumann machine model

Part 2: Assembly language
- Instructions, (structured) programming, input/output, relationship to hardware

Part 3: C programming
- Syntax, operators, control structures, functions, pointers, recursion, data structures, relationship to assembly language
- Assume already familiar with programming (but not C)

Today

Objectives
- Summarize course implementation
  - Background/Prerequisites
  - Lectures/Reading/Quizzes
  - Homework
  - Exams
  - Grades

Demo
- Sample homework/project!

Assembly Language Programming

What is an instruction?
- Basic unit of (SW) computation
- Very primitive
- E.g., in LC-3:
  > add, branch-on-condition, load-from-memory, store-to-memory

Focus: Where hardware meets software
- We will examine the hardware that executes instructions
- We will compose instructions to create software

Really little example, initially R1 is 10, R2 is 0
```
LOOP   ADD  R2, R1, R2
       ADD  R1, R1, #-1
BRp    LOOP
```

Why Take CSE240?

Foundational
- Intersects all aspects of computing

Preparatory/Complementary
- CSE 371: Digital Systems Organization and Design
- CSE 380: Operating Systems
- CSE 341: Compilers and Interpreters
- CSE 260: Mathematical Foundations of CS

Fun!!!
- Who wouldn't want to understand the magic?
Objectives
Understand role & relationship of hardware and software
Exposure to...
  • Machine organization (CSE 371 prep)
  • Assembly language programming (CSE 341 prep)
  • C programming (CSE 380 prep)

Understand how to build entire (slow) computing system
  • Hardware and software
  • You’ll get a chance in CSE 371/2 and CSE 380

Be distinguished from mere programmers

Why Learn Assembly Programming?
Helps understand capabilities of machine
  • E.g., stack smashing vulnerabilities
Many system components written in assembly
  • E.g., microcontrollers, device drivers, media kernels, digital signal processing (DSP) code

Why Study Hardware?
Important
  • Floaters can’t build effective systems!
  • Still drives industry
Timely
  • Multicore, hyper-threading, SSE, security, ...

Opens doors
  • Yet another option!

Why Learn C Programming?
What is C?
  • High-level language (than assembly anyway)
  • Invented in 1970s to write the Unix operating system
  • “Portable assembly language”
  • In between assembly and Java/C#

Very common
  • Operating systems and even general applications
    ➢ Still the right tool for many tasks
  • Foundation for C++/C#/Java
  • Assembly-to-C migration for embedded applications
CSE 240 Implementation

Staff

Background/Prerequisites
Requirement: Strong background in programming
- CSE 120
- CSE 121 (strongly recommended)

Lectures/Reading/Quizzes

Exams

Homework

Academic Integrity

Grades

Communication

Web site
- http://www.seas.upenn.edu/~cse240/
- Assignments, lecture notes, etc.

Office hours
- Fact to face help
- See web page for times

Discussion groups (via blackboard)
- Read by me, TAs, fellow students

E-mail
- Announcements via class e-mail list
- To reach us: cse240@seas.upenn.edu

Staff

Instructor
- Prof. Milo Martin (milom@cis.upenn.edu)

TAs
- Nick Monfort (grad TA)
- Netta Doron
- Geoffrey Hayes
- Andrew Lagatta
- Amanda Leicht

Background/Prerequisites

Requirement: Strong background in programming
- CSE 120
- CSE 121 (strongly recommended)

Why?
- Fast pace
- Assume you can program/debug
**Lecture**

**Expectation**
- Read appropriate sections in textbook before lecture
- See class schedule for reading assignments

**Quizzes**
- Complete easy online quiz *before* each class
- Can work ahead (do a week at once)
- Experimental (appears to be effective)

**Lectures**
- Will not simply “cover” the material
- Will focus on the “hard stuff”
- Will not stand alone, instead build on reading
- Will be interactive

---

**Homework**

**Paper and pencil assignments**
- Problem solving
- Great exam preparation

**Programming projects**
- Simple exercises
- Challenging projects (Snake game!)

**Discussion**
- Encouraged! (TAs, discussion group, etc.)
- Work must be completed alone
- Okay: discuss meaning of problem, discuss approaches
- Not okay: comparing answers, solving questions together

---

**Exams**

**Midterm**
- Covers digital logic and some assembly
- *Tentatively* scheduled: Wednesday, October 18th (in class)
- Open book

**Final**
- Comprehensive: covers assembly and C *and* digital logic
- Scheduled: Wednesday, December 20th (9am)
- Open book

**Good news: both exams are open book…**
- **Bad news:** it won’t actually help you

---

**Academic Integrity**

**The rule is simple**
- Claiming another’s work as your own will *ruin your life*
- See syllabus for details and examples

**Who will know?**
- We will (inspection, similarity detectors, exams)
- Your friends will… your parents will…
- You will

**Analogies**
- Cheating is like going 150 MPH over speed limit while drunk
- Analogous consequences (legal -> educational)

**Remember**
- If you need to cheat now, you’ve got much bigger problems
Grades
Midterm: 25%
Final: 30%
Homework: 35%
    • Three extensions allowed (see syllabus)
Quizzes: 5%
Participation, attendance, etc.: 5%

Assignments
Part I: Hardware (Digital Logic)
    • Paper and pencil assignments

Part II: Assembly Programming
    • Substantial programming project in assembly

Part III: C Programming
    • Two low-level programming assignments in C

See schedule for tentative due dates

Demo
Homeworks 6 - 7
    • Build “snake” game in assembly language!

Schedule
See web page
    • [www.seas.upenn.edu/~cse240/schedule.html](http://www.seas.upenn.edu/~cse240/schedule.html)
    • Subject to change

Notable items
    • Midterm Wed. before fall break
    • Wed before Thanksgiving: security & stack smashing
        ➢ Worth sticking around for
Next Time

Lecture
  • Chapter 1: Introduction to computer systems
  • Chapter 2 - 2.2: Integer data types

Reading
  • Chapter 1

Quiz
  • As always, online; due before start of class

Upcoming
  • Homework 1 due Friday, September 15th