CS 3410: Computer System Organization and Programming

Anne Bracy
Computer Science
Cornell University

The slides are the product of many rounds of teaching CS 3410 by Professors Weatherspoon, Bala, Bracy, and Sirer.

How class is organized

• Before you take this class...
  • What to take, what to buy
• Lecture
• Lab Sections
• Office Hours
• Online Tools
• Grading
• Who’s Who

Pre-requisites and scheduling

CS 2110 is required (Object-Oriented Programming and Data Structures)
• Must have satisfactorily completed CS 2110
• Cannot take CS 2110 concurrently with CS 3410

CS 3420 (ECE 3140) (Embedded Systems)
• Take either CS 3410 or CS 3420
  – both satisfy CS and ECE requirements
• However, Need ENGRD 2300 to take CS 3420

CS 3110 (Data Structures and Functional Programming)
• Not advised to take CS 3110 and 3410 together

Who am I? (Part 1)

Anne Bracy:

Career Path
• Undergrad @ Stanford
• Grad School @ UPenn (computer architecture)
• Intel Labs
• Lecturer @ Washington University in St. Louis
Cornell
• Just moved here
• Sr. Lecturer
• 3410, 4410 this year...

Career Path
• Undergrad @ Stanford
• Grad School @ UPenn (computer architecture)
• Intel Labs
• Lecturer @ Washington University in St. Louis
Cornell
• Just moved here
• Sr. Lecturer
• 3410, 4410 this year...
Pre-requisites and scheduling

CS 2043 (UNIX Tools and Scripting)
- 2-credit course will greatly help with CS 3410.
- Spring only 😊

CS 2022 (Introduction to C) and CS 2024 (C++)
- 1 to 2-credit course will greatly help with CS 3410
- Unfortunately, 2022 rarely offered
- Instead, we will offer a primer to C during lab sections and include some C questions in homeworks

Required Textbooks

This:

+ 1 of these:

http://aop.cs.cornell.edu
download Ch. 1-4 + UNIX appendix, see if you like it

Active Learning

L. Deslauriers et al. Science 2011;332:862-864
Fig. 1 Histogram of 270 physic student scores for the two sections: Experiment w/ quizzes and active learning, Control without.
Active Learning

Demo: What year are you in school?
  a) Freshman
  b) Sophomore
  c) Junior
  d) Senior
  e) Other

Also, activity handouts will be available before class
  In front of doors before you walk in

Active Learning

How class is organized

• Before you take this class...
  • What to take, what to buy

• Lecture
• Lab Sections
• Office Hours
• Online Tools
• Grading
• Who’s Who

Lab Sections

<table>
<thead>
<tr>
<th>Time</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:40-12:55</td>
<td>A</td>
<td>D</td>
<td>F</td>
<td>H</td>
</tr>
<tr>
<td>1:25-2:40</td>
<td>B</td>
<td>E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2:55-4:10</td>
<td>C</td>
<td></td>
<td>G</td>
<td>I</td>
</tr>
</tbody>
</table>

• all in Carpenter Hall 104 (Blue Room)
• Start this week
• separate from lecture and homework
• Bring laptop to Labs
• This week: “hello world” lab: Intro to C, VMs
• Next week: Logisim, logic circuits, and building an adder
Office Hours

My Office Hours:
• Mine: MW 9-10
• Starts tomorrow!

TA Office Hours:
• See Google Calendar (linked on Piazza)
• Start next week

Course Staff

Lab/Homework TA’s:
Aditi Jain
Anthony Lin
Brian Wang
Charles Lai
Chirag Bharadwaj
Daniel Liu
Deniz Altinbuken
Dhruv Singhal
Hong Jeon
Kylar Henderson
Lucas Derraugh
Naman Agarwal
Oscar Pacheco
Rishab Gupta
Ryan Hall
Stephanie Guo

Find them on Piazza!
Administrative Assistant:
• Megan Gatch <mlg34@cornell.edu>

Online Tools: Course Website

http://www.cs.cornell.edu/courses/cs3410
• Office Hours / Consulting Hours
• Lecture slides, schedule, and Logisim
• CSUG lab access (esp. second half of course)
• will be up by the end of the week

Online Tools: Piazza

http://piazza.com/cornell/fall2015/cs3410
• Everything happens here
• Email risks getting overlooked/answered late

Do not email me or any TA class Questions
• Guaranteed response “Please post to piazza”
• Redundancy is bad
• Single point of failure is bad
• My inbox is abysmal

This class is relentless.
Stay on top of it!

While there: Answer someone else’s question!
Online Tools: CMS

http://cms.csuglab.cornell.edu
- Assignments submitted here
- Grades given back here

Lab Sections, Projects, and Homeworks

Labs Assignments
- Individual
- One week to finish (usually Monday to Monday)

Projects
- two-person teams
- Find partner in same section

Homeworks
- One before each prelim
- Will be released a few weeks ahead of time
- Finish question after covered in lecture

Grading

Lab (50% approx.)
- 5-6 Individual Labs
  - 2 out-of-class labs (5-10%)
  - 3-4 in-class labs (5-7.5%)
- 4 Group Projects (30-35%)
- Participation/Quizzes in lab (2.5%)

Lecture (50% approx.)
- 2 Prelims (35%)
  - Dates: September 29, December 1
- Homework (10%)
- Participation/Quizzes in lecture (5%)

Grading

Regrade policy
- Logistics to be announced on piazza
- In writing
- Within 1 week of the assignment (or exam)’s return

Late Policy
- Each person has a total of four “slip days”
- Max of two slip days for any individual assignment
- For projects, slip days are deducted from all partners
- 25% deducted per day late after slip days are exhausted
Who am I? (Part 2)

Nice and a vertebrate

- Piazza posts about course material very welcome!
- Correspondence about use of slip days, your alarm clock, your all-nighters, your alcohol intake, your car battery, etc. etc. waste your time and mine
- I do not grant exceptions
- Deadlines are firm

Academic Integrity

All submitted work must be your own
- OK to study together, but do not share soln’s
- Cite your sources

Project groups submit joint work
- Same rules apply to projects at the group level
- Cannot use of someone else’s soln

Closed-book exams, no calculators

- Stressed? Tempted? Lost?
  - Come see us before due date!

Plagiarism in any form will not be tolerated

“Sometimes it is the people that no one imagines anything of who do the things that no one can imagine”

-- quote from the movie The Imitation Game

“Can machines think?”

-- Alan Turing, 1950
Computing Machinery and Intelligence
**Course Objective**

Understand the HW / SW interface software
- How a processor works
- How a computer is organized

How to write good programs. What is good?
- Correct
- Fast
- Secure

Understand technology (past, present, future)

**Who are you?**

Demographics
Introduce yourself to the people next to you

“Sometimes it is the people that no one imagines anything of who do the things that no one can imagine.”
Where did it begin?

Electrical Switch
- On/Off
- Binary

Transistor

The first transistor on a workbench at AT&T Bell Labs in 1947

Moore’s Law

1965
- # of transistors integrated on a die doubles every 18-24 months (i.e., grows exponentially with time)

Amazingly visionary
- 2300 transistors, 1 MHz clock (Intel 4004) - 1971
- 16 Million transistors (Ultra Sparc III)
- 42 Million transistors, 2 GHz clock (Intel Xeon) – 2001
- 55 Million transistors, 3 GHz, 130nm technology, 250mm² die (Intel Pentium 4) – 2004
- 290+ Million transistors, 3 GHz (Intel Core 2 Duo) – 2007
- 721 Million transistors, 2 GHz (Nehalem) - 2009
- 1.4 Billion transistors, 3.4 GHz Intel Haswell (Quad core) – 2013

Processor Performance Increase
Parallelism

CPU: Central Processing Unit

Then and Now

- The first transistor
  - One workbench at AT&T Bell Labs
  - 1947
  - Bardeen, Brattain, and Shockley

- An Intel Haswell
  - 1.4 billion transistors
  - 177 square millimeters
  - Four processing cores

Parallelism

CPU: Central Processing Unit

GPU: Graphics Processing Unit

Then and Now

- The first transistor
  - One workbench at AT&T Bell Labs
  - 1947
  - Bardeen, Brattain, and Shockley

- Galaxy Note 3
  - 8 processing cores
Supercomputers

- Petaflops \( (10^{15}) \)
  - GPUs/multicore/100s-1000s cores

Course Objective

Bridge the gap between hardware and software
- How a processor works
- How a computer is organized

Establish a foundation for building higher-level applications
- How to understand program performance
- How to understand where the world is going
To be better Computer Scientists and Engineers

- Abstraction: simplifying complexity
- How is a computer system organized? How do I build it?
- How do I program it? How do I change it?
- How does its design/organization effect performance?
Computer System Organization

Computer System = ?
Input +
Output +
Memory +
Datapath +
Control

CPU

Registers

Keyboard

Mouse

Video

Network

USB

bus

bus

Serial

Memory

Disk

Audio

Compilers & Assemblers

C

int x = 10;
int y = 2 * x + 15;
int z = r0 = 0

MIPS

assembly
language

addi r5, r0, 10
mul r5, r5, 2
addi r5, r5, 15

r5 = r0 + 10
r5 = r5 * 2
r5 = r15 + 15

MIPS

machine
language

00100000000001010000000000001010
0000000000000101010101010100010000
00100001100101000100000000001111

op = addi r0 r5 10
bp = addi r5 r5 15

Instruction Set Architecture

ISA

• abstract interface between hardware and the lowest level software

• user portion of the instruction set plus the operating system interfaces used by application programmers

Basic Computer System

A processor executes instructions

• Processor has some internal state in storage elements (registers)

A memory holds instructions and data

• von Neumann architecture: combined inst and data

A bus connects the two

regs

processor

bus

addr, data, r/w

memory

01010000
10010100
...
How to Program the Processor:  
**MIPS R3000 ISA**

### Instruction Categories
- Load/Store
- Computational
- Jump and Branch
- Floating Point
  - coprocessor
- Memory Management

### Registers

<table>
<thead>
<tr>
<th>OP</th>
<th>rs</th>
<th>rt</th>
<th>rd</th>
<th>sa</th>
<th>funct</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>jump target</td>
</tr>
</tbody>
</table>

---

**Overview**

- Application
- Operating System
- Compiler
- Firmware
- Instruction Set Architecture
  - Memory system
  - I/O system
  - Datapath & Control
  - Digital Design
  - Circuit Design
Applications

Everything these days!
  • Phones, cars, televisions, games, computers,…

Covered in this course

Why take this course?
  • Basic knowledge needed for all other areas of CS: operating systems, compilers, ...
  • Levels are not independent
    hardware design ↔ software design ↔ performance
  • Crossing boundaries is hard but important
    device drivers
  • Good design techniques
    abstraction, layering, pipelining, parallel vs. serial, ...
  • Understand where the world is going
    The Mysteries of Computing will be revealed!