Today

- Course Overview
- LMC intro

Assignments

- HW1 (out, due Tues)
Course Overview

Course communication

- Website: [http://www.cs.gonzaga.edu/bowers/courses/cpsc223](http://www.cs.gonzaga.edu/bowers/courses/cpsc223)
- Piazza (you’ll receive an invite)

Please be sure to **carefully** read the syllabus

- **Office hours**: TR 11–1, W 1–2, by appointment
- **Grading**:
  - 40% homework
  - 10% quizzes
  - 30% three mid-semester exams
  - 20% final exam
Exam, Quiz, and Homework basics

Exams and Quizzes are closed book, closed note, your own work

- you are responsible for understanding how to do the assignments
- you will be tested on material from assignments
- you must turn in homework individually

(Some) collaboration is encouraged ...

- discuss and talk about assignments
- work together (as long as you are doing your own work)
- plagiarism not allowed (e.g., do not split up work, share code)
Class Sessions

- Mix of lecture, exercises, and short quizzes
- No electronic devices allowed during lecture and quizzes
  - Please see me for accommodation
  - Okay to use during exercises as needed
- Quizzes mostly at beginning of lectures (no makeup quizzes)
- Highly encourage questions during lecture
- Use piazza for questions outside of class
- Lecture notes posted after each class

Homework

- Mix of written and programming assignments
- Programming assignments more like “exercises” (versus “real” applications)
Course Topics

Main goals:

- Understanding of computer architecture and low-level programming
- Interaction between higher (C/C++) and lower-level (assembly) languages
- Background needed for Operating Systems, Security, Networking, Architecture
- “Power programming” (see book)

Main topics:

- Basic intro to Computer Architecture (about 4 weeks)
- Data / number representations (about 2 weeks)
- Machine and Assembly Programming (about 5 weeks)
- Memory hierarchy (about 3 weeks)
- Basic C programming (integrated)

Warning: lots of details and complexity ...

- This class is a jumping off point / the basics
- The book goes into much more detail than we can cover in class
Quizzes

At least one quiz a week
- usually at beginning of class
- usually take about 10 minutes
- no make-up quizzes!

Why are there so many quizzes?
- Give you feel for kind of questions I ask on exams
- Give you feedback on your understanding of material
- Give me feedback on your understanding
- Short quizzes shown to help improve final course grades / retention

⇒ Quiz 0
Basic Building Blocks of Computer Systems

- Hardware supports machine instructions
  - called the “Instruction Set Architecture” (ISA)
  - ARM, x86, x86-64 are examples
- Operating system provides abstractions over hardware (e.g., files, processes)
- And support for writing/running programs (e.g., libraries, execution)
- Programs in high-level languages converted to machine executable formats
Getting Started: Simple Model of a Computer

The LMC (“Little Man Computer”)

- Simple “computer” model for learning architecture and machine-programming

The little man repeats the following steps

1. Read 2-digit number from instruction counter (e.g., 00)
2. Fetch slip of paper in mailbox with that number (e.g., in box 00)
3. Remember 3-digit number (e.g., 901) on paper, return paper to mailbox
4. Push “next” button (increment) on the instruction counter
5. Perform operation (e.g., 901) on slip of paper

Called the “Fetch-Decode-Execute” cycle:

- fetch instruction from memory, decode it (type, operands), and execute it
More details:

There is an LMC operator (different from the little man) that:
  • clears and places slips of paper (instructions) in the mailboxes
  • resets the instruction counter to 00
  • wakes up the little man to start program execution

The inbox and outbox hold paper slips for interacting with users
  • each paper slip contains a three-digit number
  • little man adds new slips to the outbox tray
  • user adds new slips to the inbox tray
  • little man always takes slips from the bottom of the inbox (slip there the longest)
  • slips taken from the inbox tray are never put back by the little man

The calculator
  • stores 3 digit numbers (the “accumulator”)
  • can add and subtract from the stored 3-digit number (result stored)
  • if result of subtracting is negative, screen turns red (“negative flag”)
  • if value is zero, screen turns green (“zero flag”)

Instructions
  • First digit of an instruction is the “operation code” (or “opcode” for short)
  • Operations to add, subtract, store, load, input/output, etc.
  • Rest of instruction is an “operand”, usually a mailbox number (“address”)