Computer Organization and Architecture

Week One

Utah Tech University

Fall 2023

Introduction

Who am I?

- Dr Russ Ross
  - From Southern Utah, went to high school in St. George
  - Undergrad at Harvard (AB in Computer Science)
  - Worked in Boston area during the .com boom
  - Graduate school at Cambridge (PhD in Computer Science)
  - At UT since 2007
- Call me “Russ” or “Dr Ross” (how to choose?)
  - Never “Mr Ross”—we are not in high school

Who are you?

- CS? SE? Something else? Undecided?
  - Computer Science vs Software Engineering

Overview

What is this class about?

- The hardware-software interface
  - CPUs, ISAs, ABIs
  - Memory layout: segments, the stack, the heap
  - Assembly language and how it relates to high-level languages
- Data representation
  - Binary numbers, bytes, words
  - 2s-complement numbers
  - IEEE-754 floats
  - Arrays, structs, pointers
  - Strings
  - Instructions, functions
- Other low-level details programmers need to know about
  - Cache
  - Virtual memory
  - Memory allocation, garbage collection
  - User mode, kernel mode, system calls

The plan

1. Learn about CPU architecture
   - Our example will be RISC-V
   - We will learn about CPU design at a very high level
2. Write assembly language for that CPU
   - Learn how high-level language features actually work
   - Complete a substantial project (Sudoku solver)
3. Learn C (why C?)
   - *Not* C++
   - With a focus on how C relates to assembly language

Attendance, distractions, etc.

- Attendance is required. I will not take roll, but we will have in-class activities that are graded.
- You are responsible for what we talk about in class, and much of what we cover will *not* be available elsewhere
  - Assignment instructions, tips, etc.
  - If you miss class, you may not be able to complete the homework
- This is an in-person class. I will attempt to stream it via Zoom on request if there is a good reason, but the AV system is flaky and it will probably fail on some days
  - Do not depend on Zoom
- You are expected to take notes: bring pen and paper
- Laptops and mobile devices are not allowed in class unless specifically called for
  - Not even for notes or following along with demos
  - Exceptions need documentation

CodeGrinder and RISC-V assembly

First steps:

- Make sure you have a Unix shell to work in (Linux or Mac OS)
  - WSL works great for this
  - On Mac OS your best bet is to install a virtual machine
  - I recommend installing Debian 12 “Bookworm”

I will help you get started, but you should plan to spend time throughout the semester learning to better use your tools.

Unsigned binary numbers

- Bits: why?
- 2^n combinations for n bits
- Binary vs decimal vs hexadecimal vs octal
- Approximating how big a binary number is (1,000 vs 1024)
- Binary addition
- Binary fractions
- Bytes, half words, words, double words
  - Programs and data are all represented as sequences of bytes

CodeGrinder demo

- Logging in (once per semester per machine)
- Activating an assignment through Canvas
- `grind list`, `grind get`
- Recommendation: one `git` repo for the course directory
- Reading documentation for a problem
- “Hello, world!” demo