Introduction

Thursday, September 26, 2019 2:09 PM

CS 33: Demystifying the world of computer science

Computers Pre-1964

The boundary between software/hardware was pretty fuzzy

ENIAC: First machine with distinctions between software/hardware

ISA: The interface that determines the representation of programs to hardware
What goes in an ISA?
Ingredients:

- Memory! A place to put values, keep track of state, variables, etc.
- Instructions! A way to manipulate memory, move from one state to the next over time
- Program! A set of instructions (let's put it in memory)
- Execution model! When we execute each instruction

Von Neumann model

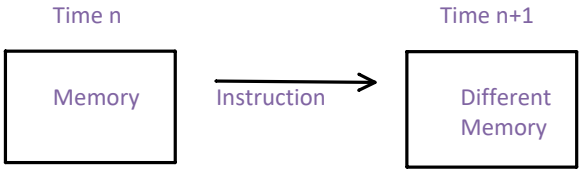
Software's implementation of the ISA: Translates our program to a reasonable sequence of instructions

Hardware's use of the ISA: Executes the sequence of instructions

Fetch
Decode
Execute
Writeback

Computer Breakdown

SOFTWARE
\/
algorithm
application
programming language
compiler
operating system - manages what programs are running when
--- Architecture (ISA) ---
hardware organization
component design
Circuit design
devices (transistors)
physics/manufacturing
/\
HARDWARE



The Limits of Abstraction

Int's are not Integers
Float's are not Reals

Is $x^2 \geq 0$?

Float's: Yes!

Int's: 40,000 * 40,000 is positive
But 50,000 * 50,000 is negative! Too many bits

Is $(x + y) + z = x + (y + z)$

Int's: yes!

Float's: augh nope

There's more to performance than asymptotic complexity.
Constant factors matter too! Must optimize at multiple levels, works best when programs are written with a specific OS/machine in mind

Something as simple as swapping the order of operations of 2 for loops could result in a program running 10 times faster or slower