Lecture 8: Arithmetic Operations & Condition Codes

Wednesday, April 15, 2015

- Arithmetic Operations: src, dest
  - add/subtract/multiply (e.g. add src, dest => dest+=src)
  - shift arithmetic/logical right/left
  - xor, and, or
  - increment, decrement, negate, complement

- Reading x86
  - instructions may not be one-to-one (many lines of assembly for one in C, and vise versa)
  - compilation of two different C programs may yield the same assembly program

- Condition codes: indicate something about the state of the last operation
  - carry bit set if computing the result overflowed when interpreted as unsigned
  - sign bit set if the result is less than zero/ a > b when comparing
  - overflow bit set if computing the result overflowed when interpreted as signed
  - zero bit set if the result is zero / a==b when comparing
  - Set by: math operations, test operations, and compare operations
  - Used by: jump operations, set operations

- Conditional/Unconditional Jumps (Branches)
  - jmp always makes the jump to a location
  - all others are conditional (e.g. je jump if equal to zero)
    - looks at condition codes to decide if conditional is true (e.g. zero bit is set)
    - if true, make jump to target location
    - otherwise, falls through and continues executing
      - creates separate (labeled) sections of code for if/else statements
  - hardware makes a guess as to which direction it's going to go (often guesses wrong!)

- Conditional move operations
  - e.g. cmovle src, dest move value from src to dest if condition holds
  - but requires computing the results for both sides of the branch ahead of time