Chapter 2: Numbers

Binary

- When converting from decimal to binary we can divide by two and get each successive bit.
- Example 1005
  - 1005 / 2 = 502 r 1
  - 502 / 2 = 251 r 0
  - 251 / 2 = 125 r 1
  - 125 / 2 = 62 r 1
  - 62 / 2 = 31 r 0
  - 31 / 2 = 15 r 1
  - 15 / 2 = 7 r 1
  - 7 / 2 = 3 r 1
  - 3 / 2 = 1 r 1
  - 1 / 2 = 0 r 1
  - The binary number = 1111101101

Hexadecimal

- Base 16 numbers and we can use the same process as above to convert to hex.
- example 1005
  - 1005 / 16 = 62 r 13
  - 62 / 16 = 3 r 14
  - 3 / 16 = 0 r 3
  - The hex number = 0x3ED

Integers

- Integers on x86_64 can be 1 byte, 2 bytes, 4 bytes, or 8 bytes.
    - can be unsigned or signed.
- Signed integers are stored in two's complement.
    - the first bit of the integer is a sign bit.
        - 0 for positive
        - 1 for negative.
    - invert the bits and add 1.

Binary Arithmetic

- Addition: Works how it should with carrying etc.
- Multiplication: Works the same way as well.

Floating Point Numbers

- x86_64 supports three different varieties of floating point numbers.
    - 32-bit, 64-bit, and 80-bit.
    - 32-bit is float.
    - 64-bit is double
    - 80-bit is long double.
- Each order has the highest order bit as the sign bit.
    - 1 for negative 0 for positive.
- The float type.
    - Exponent for a float number is 8.