ME 333: Winter Break Homework

Marshall Johnson

January 4, 2022

A Crash Course in C — Exercises

1) **Install C, create the HelloWorld.c program, and compile and run it.**

   See hello_world_demo.mp4

2) **Explain what a pointer variable is, and how it is different from a non-pointer variable.**

   A pointer variable stores the address of another variable. Pointer variables are allocated platform-dependent memory needed to store an address. Conversely, non-pointer variables are allocated memory based on their specific data type.

3) **Explain the difference between interpreted and compiled code.**

   The primary difference between interpreted and compiled code is when the program is turned into machine-code. Commands from interpreted programs are converted to machine-code and executed while the program is running. On the other hand, commands from compiled programs are converted to machine-code in advance. Due to this distinction, compiled code typically runs faster.

4) **Write the following hexadecimal (base-16) numbers in eight-digit binary (base-2) and three-digit decimal (base-10). Also, for each of the eight-digit binary representations, give the value of the most signifcant bit.**

   a. **0x1E**
      i. Eight-digit Binary (base-2): 00011110
         - Most significant bit: 0
      ii. Three-digit Decimal (base-10): 30

  b. **0x32**

   i. Eight-digit Binary (base-2): 00110010
    • Most significant bit: 0
   ii. Three-digit Decimal (base-10): 50

  c. **0xFE**

   i. Eight-digit Binary (base-2): 11111110
    • Most significant bit: 1
   ii. Three-digit Decimal (base-10): 254

  d. **0xC4**

   i. Eight-digit Binary (base-2): 11000100
    • Most significant bit: 1
   ii. Three-digit Decimal (base-10): 196

6) **Assume that each byte of memory can be addressed by a 16-bit address, and every 16-bit address has a corresponding byte in memory. How many total bits of memory do you have?**

16-bit address — 65,536 possible bytes addressed, so **524,288 total bits of memory**.

7) **Let ch be of type char.**

  a. The assignment ch = 'k' can be written equivalently using a number on the right side. What is that number?
   107

  b. The number for '5'?
   53

  c. For '='?
   61

  d. For '?'?
   63

8) **What is the range of values for an unsigned char, short, and double data type?**

  • unsigned char: 0 to 255
  • short: -32,768 to 32,767
  • double: -2^{2048} to 2^{2048}

10) **Explain the difference between unsigned and signed integers.**

An unsigned integer does not include negative values in its range. Signed integers include negative values. The range for an unsigned integer is 0 to 2^{32} - 1, while the range for a signed integer is -(2^{31}) to 2^{31} - 1.

11) a. **For integer math, give the pros and cons of using chars vs. ints.**

Using chars rather than ints uses less memory and can result in faster computation time. However, with chars, the range of integer values is more limited and can lead to integer overflow.

  b. **For floating point math, give the pros and cons of using floats vs. doubles.**

Using float rather than double uses less memory and can result in faster computation time. However, doubles will give greater precision in representation.

  c. **For integer math, give the pros and cons of using chars vs. floats.**

Using chars rather than floats uses less memory and can result in faster computation time. However, chars are limited to smaller numbers and can lead to integer overflow. Using floats will allow for integer math with larger numbers. That said, floats will lead to longer computation time and can lead to decreased precision.

16) **Technically the data type of a pointer to a double is "pointer to type double." Of the common integer and foating point data types discussed in this chapter, which is the most similar to this pointer type? Assume pointers occupy eight bytes.**

unsigned long int — this data type occupies 8 bytes and represents a non-negative integer, similar to the data type "pointer to type double"

17) **For each of the comments (a)-(g) above, give the contents (in hexadecimal) at the address ranges 0xB0..0xB3 (the unsigned int i), 0xB4..0xB7 (the unsigned int j), 0xB8 (the pointer kp), and 0xB9 (the pointer np), at that point in the program, after executing the line containing the comment.**

  a. unknown for all memory locations

  b.
  - 0xB0..0xB3: unknown
  - 0xB4..0xB7: unknown
  - 0xB8: 0xB0
  - 0xB9: unknown

  c.
  - 0xB0..0xB3: unknown
  - 0xB4..0xB7: unknown
  - 0xB8: 0xB0
  - 0xB9: unknown

  d.
  - 0xB0..0xB3: 0xAE
  - 0xB4..0xB7: unknown
  - 0xB8: 0xB0
  - 0xB9: unknown

  e.
  - 0xB0..0xB3: 0xAE
  - 0xB4..0xB7: unknown
  - 0xB8: 0xB0
  - 0xB9: 0xB0

  f.
  - 0xB0..0xB3: 0x12
  - 0xB4..0xB7: unknown
  - 0xB8: 0xB0
  - 0xB9: 0xB0

  g.
  - 0xB0..0xB3: 0x12
  - 0xB4..0xB7: 0x12
  - 0xB8: 0xB0
  - 0xB9: 0xB0