University of Waterloo

CS241 - Winter 2024 - Course Notes

Author: Brandon Zhou

Course Code: CS241

Course Name: Foundations of Sequential Programs

Instructor:

Room:

Days & Times:

Section:

Date Created: January 09, 2024

Final Exam Date: TBA

Disclamer: These course notes are intended to supplement primary instructional materials and facilitate learning. It's worth mentioning that some sections of these notes might have been influenced by ChatGPT, an OpenAI product. Segments sourced or influenced by ChatGPT, where present, will be clearly indicated for reference.

While I have made diligent efforts to ensure the accuracy of the content, there is a potential for errors, outdated information, or inaccuracies, especially in sections sourced from ChatGPT. I make no warranties regarding the completeness, reliability or accuracy of the notes contained in this notebook. It's crucial to view these notes as a supplementary reference and not a primary source.

Should any uncertainties or ambiguities arise from the material, I strongly advise consulting with your course instructors or the relevant course staff for comprehensive understanding. I apologize for any potential discrepancies or oversights.

Any alterations or modifications made to this notebook after its initial creation are neither endorsed nor recognized by me. For any doubts, always cross-reference with trusted academic resources.

Table of Contents

Lecture 1

- Definition: A bit is a binary digit. That is, a 0 or 1 (off or on)
- Definition: A nibble is 4 bits.
    - Example: 1001
- Definition: A byte is 8 bits.
    - Example: 10011101
- in C/C++:
    - Char: 8 bits
    - Unsigned char: 8 bits
    - Short: 2 bytes/16 bits
    - int: 4 bytes
    - longlong: 16 bytes
- Definition (Hexadecimal Notation): The base-16 representation system is called the hexadecimal system. It consists of the numbers from 0 to 9 and the letters a, b, c, d, e, f(which convert to the numbers from 10 to 15 in decimal notation)
- Notation:
    - Binary: 0, 1
    - Decimal(Base 10): 0, ..., 9
    - Hexadecimal: 0, ..., 9, A(10), B(11), C(12), D(13), E(14), F(15)
- Examples:
    - 0000(base 2) -> 0x0(base 16)
    - 1111(base 2) -> 0xf(base 16)
- What do bytes represent?
    - Numbers
    - Characters
    - Garbage in memory
    - Instructions(Words, or 4 bytes, will correspond to a compute instruction in our computer system)
- Bytes as Binary Numbers
    - Unsigned (non-negative integers)
        - $b_7...b_0$(base 2) = $b_7 \times 2^7 + ... + b_0 \times 2^0$(base 10)
        - Example: $01010101 = 0 \times 2^7 + ... + 1 \times 2^0$
        - Converting to Binary:
            - Take the largest power of 2 less than the unsigned integer, subtract and repeat
            - Another way is to constantly divide by 2, get the remainer for each division, reading *from the bottom to up* at the end, and that will be the binary representation of this unsigned integer
    - Signed integers
        - Attempt 1: make the first bit a signed bit. This is called the "sign-magnitude" representation
            - Problems:
                - Two representations of 0(wasteful and awkward)
                - Arithmetic is tricky. Is the sum of a positive number and a negative number positive or negative.
        - Attempt 2: Two's complement form
            - Similar to "sign-magnitude" representation in spirit, first bit is 0 if negative, 1 if positive
            - Negate a value ... (INCOM)
            - Decimal to Two's Compliment:
                - A trick to the the same thing:
                    - Take the complement of all bits
                    - Add 1
                - A slightly faster trick is to locate the rightmost 1 bit and flip all the bits to the left of it
                    - Example: 11011010 Negating: 00100101 + 1 = 00100110
                    - Example: compute -38_{10} using this notation in one byte of space:
                        - Step 1: $38_{10} = 00100110_2$
                        - Step 2: take the complment of all the bits: 11011001_2
                        - Step 3: plus 1 11011010_2
            - Two's Compliment to Decimal

Lecture 2

Lecture 3

Lecture 4

Lecture 5

Lecture 6

Lecture 7

Lecture 8

Lecture 9

Lecture 10

Lecture 11

Lecture 12

Lecture 13

Lecture 14

Lecture 15

Lecture 16

Lecture 17

Lecture 18

Lecture 19

Lecture 20

Lecture 21

Lecture 22

Lecture 23

Lecture 24

Final Review