Lecture 17

More on value representation, basic computer organization

Admin

- Project A
- Documentation
  - I've posted the slides from previous lectures.
  - I've also posted a list of links to web sites which cover the material I've covered over the last few days.

Bitwise operators

- C++ supplies ways to manipulate groups of bits.
  - "|" is bitwise or, while "&" is bitwise and.
- What do the following lines of code do?
  - A=5|7;
  - B=5&7;
  - C=3|16;
  - D=3|17;
  - E=D&17;
- "^" is xor

Shifting

- >> is bit shift right, << is bit shift left

```
main()
 {
    unsigned int Value=4;          /*  4 = 0000 0100 */
    unsigned int Shift=2;

    Value = Value << Shift;        /* 16 = 0001 0000 */
    Value = Value >> 1;            /* 8 = 0000 1000 */
}
```

Hex

- The reason we use base 16 is that it can be used to represent binary in an easier to read format.

Dec.	Binary	Hex
0	0000	0
1	0001	1
9	1000	8
129	1000 0001	81
255	1111 1111	FF

Floating point

```
Value   HEX            Binary
=====   ====           s exponent mantissa
                       = ======== =======================
0       00000000       0 00000000 00000000000000000000000
1       3F800000       0 01111111 00000000000000000000000
2       40000000       0 10000000 00000000000000000000000
4       40800000       0 10000001 00000000000000000000000
5.75    40B80000       0 10000001 01110000000000000000000
-.1     BDCCCCCD       1 01111011 10011001100110011001101
```

Ideas:

- Hidden bit
- Exponent in excess 127
- Doubles are the same, just more bits

Why *you* care

- Consider using a notation like
  - (S, E, M) in base 10.
    - S is sign
    - E is exponent
    - M is mantisa
  - Magnitude is $M * 10^E$
    - Say that M has 2 digits, E has 1 digit and a sign
      - What is the largest number you can have?
      - The smallest?
      - What if you add the biggest to the smallest?

Gap

- The good thing is we can add small numbers to small numbers and big numbers to big numbers.
  - But as numbers get larger, the *gap* between them grows.
    - For our decimal scheme, what is the gap between 100 and the next largest number?
    - How about between 1/100 and the next largest number?

Memory

- The basic idea is that memory is really just a big array of bytes.
  - Really big array. 1GB is around 1 billion bytes!
  - All of our data, doubles, ints, chars, whatever; are storied in these bytes.
    - If I ask for an int at location 1000 I really get the 4 bytes from 1000 to 1003.
    - Remember bits are bits are bits.

Computer organization

- A computer executes instructions out of memory.
  - As before these are just bits.
  - Usually an instruction is 32 bits (4 bytes).
  - The instruction will specify an operation (say add, subtract etc.)
  - Once the instruction finishes, it gets the next one in memory.
  - Some instructions (called branches) change the flow of the program.
    - They tell the computer to use some other address in memory for the next instruction.