CSCI-1190: Beginning C Programming for Engineers

Lecture 6: Number Systems

Number Systems

- Different number systems use different *bases*
  - Decimal (Base 10):
    $3472=3*10^3+4*10^2+7*10^1+2*10^0=3472$
  - Octal (Base 8):
    $3472=3*8^3+4*8^2+7*8^1+2*8^0=1850$
  - Hexadecimal (Base 16)
    $3472=3*16^3+4*16^2+7*16^1+2*16^0=13426$

Binary Numbers

- Binary is a base 2 number system with only two digits available, 0 and 1
  $1011 = 1*2^3+0*2^2+1*2^1+1*2^0 = 11$
- Binary digits are called *bits*
- A bit can be represented electronically with an on/off switch
- A byte consists of 8 bits

Bit Operators

- The bits operators operate bitwise on numbers as bit patterns
- For each bit, x and y:

x	y	AND	OR	XOR
		x&y	x|y	x^y
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

x	Complement
	~x
0	1
1	0

Bit Operations: Examples

x	0	0	1	0	0	1	0	1	37
y	1	1	1	1	0	1	1	0	246
x&y									36
x|y									247
x^y									211
~x									218

Bit Operators versus Logical Operators

- Logical operations:
  `37 && 246` → `1`
  `37 || 246` → `1`
  `! 37` → `0`
- Bit operations:
  `37 & 246` → `36`
  `37 | 246` → `247`
  `~ 37` → `218`

Masking and Setting

x	b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
28	
x&28	b_4 b_3 b_2
227	
x|227	b_4 b_3 b_2

Shift Operations

- Shift operators move bit patterns either left or right
  - 1 bit left: multiply by 2
  - 1 bit right: divide by 2

x	0	0	1	0	0	1	0	1	37
x<<1									74
x<<2									148
x<<3									40
x>>1									18
x>>2									9
x>>3									4

From Binary to Decimal

- Input as a string: 0100101
- From characters to digits:
  '0' → 0, '1' → 1
- Calculate:
  $0*2^6+1*2^5+0*2^4+0*2^3+1*2^2+0*2^1+1*2^0$
  = 0+32+0+0+4+0+1=37

From Decimal to Binary

- Input as an integer: 37
- Repeat:
  - x&2^7 → `0` → `0`
  - x&2^6 → `0` → `0`
  - x&2^5 → `32` → `1`
  - x&2^4 → `0` → `0`
  - x&2^3 → `0` → `0`
  - x&2^2 → `4` → `1`
  - x&2^1 → `0` → `0`
  - x&2^0 → `1` → `1`

In-Class Exercise 6-1

- Write a program that reads an integer from the keyboard, then prints out its binary representation.
  - Assume the input integer is always smaller than 2^{16}=65536
  - Use the shift operator to generate the sequence of 2^{15},2^{14},2^{13},…,2^1,2^0

Octal Numbers

- Base: 8
- Digits: {0,1,2,3,4,5,6,7}
- Examples:
  127 = $1*8^2+2*8^1+7*8^0$ = 87
- Octal integer constants are numbers with a leading zero

```
017 /* 1*8+7=15 */
0234 /* 2*64+3*8+4=156 */
019 /* error */
```

Hexadecimal Numbers

- Base: 16
- Digits: {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}
- Examples:
  - A2F = $A*16^2+2*16^1+F*16^0$
    $= 10*16^2+2*16^1+15*16^0 = 2607$
- Hex integer constants are numbers with a leading *0x*

```
0x11 /* 1*16+1=17 */
0xA0 /* 10*16+0=160 */
0xAB /* 10*16+11=171 */
```

Bit Pattern Conversions

Octal Numbers		Hexadecimal Numbers			
0	000	0	0000	8	1000
1	001	1	0001	9	1001
2	010	2	0010	A	1010
3	011	3	0011	B	1011
4	100	4	0100	C	1100
5	101	5	0101	D	1101
6	110	6	0110	E	1110
7	111	7	0111	F	1111

- No binary constants in C
- Octal and hexadecimal constants are convenient
  - 101 1001 1011 1101 1111 = 0x59BDF

From Hex to Dec

- Input as a string: 59BDF
- From characters to digits:

```
/* a is the character */
a is '0'-'9': n=a-'0';
a is 'A'-'F': n=a-'A'+10;
```

- From digits to the number

```
59BDF
=5*16^4+9*16^3+11*16^2+14*16^1+15*16^0
```

From Dec to Hex

- Input as an integer: 44444
- Repeat:
  - $44444/16^3=10$ remainder 3484 10 → 'A'
  - $3484/16^2=13$ remainder 156 13 → 'D'
  - $156/16^1=9$ remainder 12: 9→ '9'
  - $12/16^0=12$ remainder 0: 12 → 'C'
- This algorithm also works for octal numbers and binary numbers

In-Class Exercise 6-2

- Write a program that reads an integer from the keyboard, then prints out its hexadecimal representation.
  - Assume the input integer is smaller than 16^4=65536, so we start from 16^3=4096