汇编语言程序设计

Assembly Language Programming

第一章 基础知识

§1 数的表示

- 数制
- 数制之间的转换
- 计算机中数的表示
- BCD码
- 字符编码

1.1 数 制

- **十进制**：基数为10，逢十进一

  $12.34D = 1\times10^{1} + 2\times10^{0} + 3\times10^{-1} + 4\times10^{-2}$

- **二进制**：基数为2，逢二进一

  $1101B = 1\times2^{3} + 1\times2^{2} + 1\times2^{0} = 13_{10}$

- **十六进制**：基数为16，逢十六进一

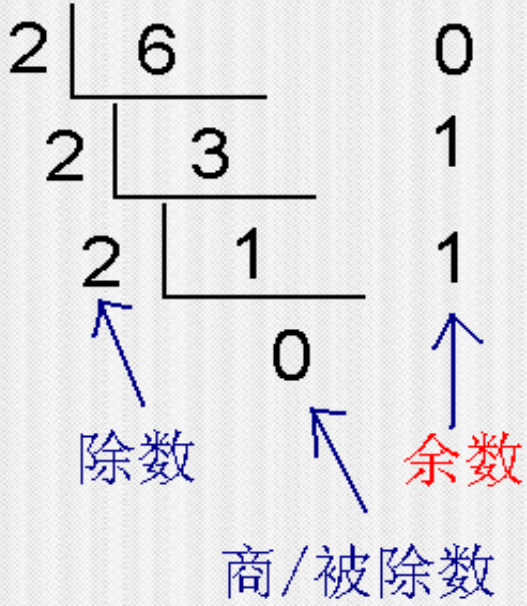
  9 1 8 7 H

  $= 9\times16^{3} + 1\times16^{2} + 8\times16^{1} + 7\times16^{0}$

- **八进制**：基数为8，逢八进一

1.2 数制之间的转换

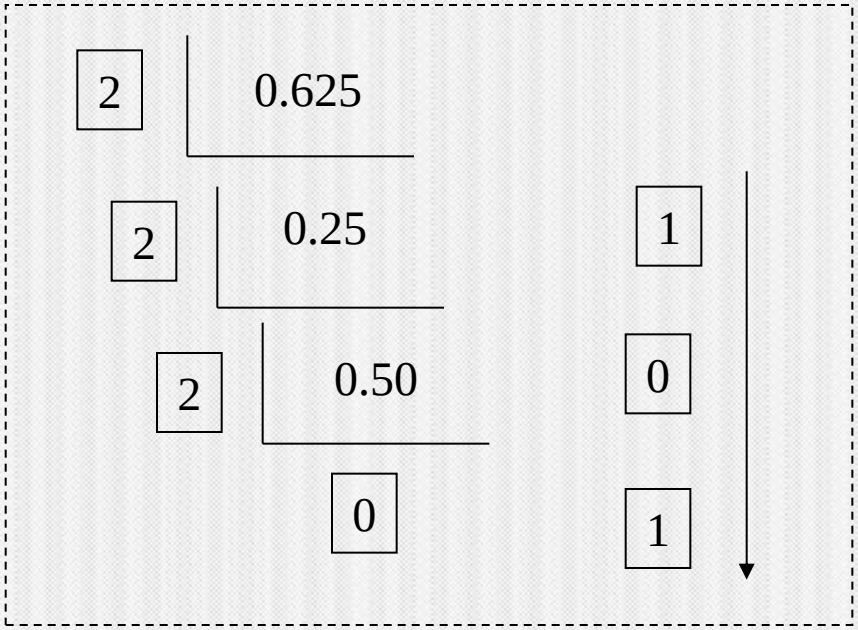
- 二进制 ⇄ 十进制
  - → 1011B = 11D
  - ← 除法：
    - 整数部分除2；小数部分乘2。



步骤演化：

2 | 6 0
3
① 6 ÷ 2，商2余0

2 | 6 0
2 | 3 1
1
② 3 ÷ 2，商1余1



0.625D = 0.101B

- 小数可能不能用二进制来表示完全，如0.6

- 二进制 ⇄ 十六进制

→ 0011 0101 1011 1111

↓ ↓ ↓ ↓

3 5 B F

← A 1 9 C

↓ ↓ ↓ ↓

1010 0001 1001 1100

∴ A19CH = 1010,0001,1001,1100B

- 4位二进制对应一位16进制

- 十六进制 ⇄ 十进制

⇄ $BF3CH = 11\times16^3 + 15\times16^2 + 3\times16^1 + 12\times16^0$

降幂法　除法

- 常用数

0—00H　128—80H　255—FFH　256—100H

32767—7FFFH　　65535—FFFFH

1.3 计算机中数的表示

- 肯定是用二进制
- 无符号数：直接用“二进制”来表示。
- 有符号数：原码、反码、补码。
- 浮点数：（尾数）规格化+（指数）移码减一

补 码

- 定义
  - 最高有效位为符号位：0—正；1—负
  - 正数的补码为它本身（二进制值）;负数的补码由$2^n-|x|$求得。n为机器的字长。
  - 注：$2^n-|x|$ 等价于 对|x|取反+1
  - $[46]_{补码}$ = 0010 1110
  - $[-46]_{补码} = 2^8 - |-46| = 1,0000,0000 - 0010\ 1110 = 11010010$
  - $[-46]_{补码}$：|-46| = 46 = 0010 1110 → 11010010

补码的特性

- 求补特性
  - 求补：取反加1
  - $[x]_{补}$  $[-x]_{补}$
  - 补码为11001100，十进制数为？
- 加减
  - $[x+y]_{补} = [x]_{补} + [y]_{补}$
  - $[x-y]_{补} = [x]_{补} - [y]_{补}$
  - 这个特性使得对于补码数或者一般的二进制数对加减规则一样。不可用于乘除运算。
  - $[x-y]_{补} = [x]_{补} + [-y]_{补}$：减法可用加法实现。

n位二进制补码的表数范围

十进制	二进制	十六进制	十进制	十六进制
	n=8		n=16	
+127	0111 1111	7F	+32767	7FFF
+126	0111 1110	7E	+32766	7FFE
…	…	…	…	…
+2	0000 0010	02	+2	0002
+1	0000 0001	01	+1	0001
0	0000 0000	00	0	0000
-1	1111 1111	FF	-1	FFFF
-2	1111 1110	FE	-2	FFFE
…	…	…	…	…
-126	1000 0010	82	-32766	8002
-127	1000 0001	81	-32767	8001
-128	1000 0000	80	-32768	8000

一个带符号数在不同位数下其二进制补码表示可能是不同的

数 的 范 围

- 无符号数：8位(0---255） 16位（0---65535）
- 有符号数：

  8位（-128---127） 16位（-32768---32767）
- n位补码表示数范围： $-2^{n-1} \leq N \leq 2^{n-1}-1$

补码:C语言

- 其实C语言中unsigned和signed两个关键字就是来区别无符号数和有符号数，缺省的是signed类型，也就是有符号数。
- 实例:
  - char i = -1;
  - short j = -1;
  - int k = -1

补码:JAVA语言

- 实例:
  - byte x=(byte)255 System.out.println(x) //?
  - byte x=64;

    byte y=64;

    byte z=(byte)(x+y);

    System.out.println(z);//?

    System.out.println(x+y);//?

//-1;-128;128

1.4 BCD（Binary-Coded Data）码

- Why? 十进制用起来很方便。
- 二进制编码的十进制：
  - Packed BCD：用4位二进制表示一个十进制数码
    - 0000-----0　　0001-----1
    - 0001,0010,0011,0100 = 1234
  - Unpacked BCD：用8位二进制表示一个十进制数码
    - ****0000----0　****0001----1
    - ****,0001,****,0010, ****,0011, ****,0100 = 1234

- 如何表示一个长度为100的十进制数?
  - 用长度为100的数组来表示
- 如何实现两个长度为100的数的加法?
  - 用数组加
- 如何实现两个长度为100的数的乘法?
- 如何实现长度为200的数除以长度为100的数?

公钥体制和数字签名的基本原理

- P和Q为长度为100左右的质数，P*Q=N较为容易
- 由N求P和Q特别难
- 公钥：N对外公开，用来加密，与PQ有关的数作为私钥，解密
- 数字签名：用私钥对要签名内容的散列值加密，用N解密验证。

1.5 字符编码

- ASCII：英文，7 bits
- 常用字符的ASCII码。
  - 数字'0'～'9'：30H～39H
  - 字母'A'～'Z'：41H～5AH
  - 字母'a'～'z'：61H～7AH
  - 空格：20H
  - **回车CR：0DH**
  - **换行LF：0AH**
  - 空字符：0
  - 注意回车与换行的差别： CR用来控制光标回到当前行的最左端；LF用来移动光标到下一行，而所在列不变。

1.5 字符编码

- UNICODE：16位二进制 = 65536

  汉字：20902个

- UTF-8: 用4字节表示，2^{32} = 42亿….