

第3章 数据表示与数据处理

信息包括文字、数字、图片、图表、图像、音频、视频等。信息的表示有两种形态：一种是人类能够识别和理解的信息形态；另一种是计算机能够识别和理解的信息形态。由于计算机硬件是由电子元器件组成的，而电子元器件大多都有两种稳定的工作状态，可以很方便地用“0”和“1”来表示，因此，在计算机内部普遍采用二进制编码“0”和“1”来表示信息，也就是说，各种信息都必须经过数字化编码后才能被传送、存储和处理，这就使得通过输入设备输入到计算机中的任何信息，都必须转换成“0”和“1”编码的表示形式，才能被计算机所识别。

计算机需要处理的信息分为数值信息和非数值信息。本章主要介绍数制的基本概念和数值信息及非数值信息的表示与处理。



(视频资料)

3.1 数制的相关概念

生活中数制是人们利用符号来计数的科学方法，又称为计数制。数制有很多种，例如，最常用的是十进制，钟表是六十进制，年是十二进制等。无论是哪种数制，都包含基数和位权两个基本要素。

3.1.1 数制的基本要素

1. 基数

在一个计数制中，表示每个数位上可用字符的个数称为该计数制的基数。例如，十进制数，每一位可使用的数字为0, 1, ..., 9共10个，则十进制的基数为10，即逢十进一；二进制中用0和1来计数，则二进制的基数为2，即逢二进一。一般来说如果数制只采用 R 个基本符号，则称为 R 数制， R 称为数制的“基数”。

2. 位权

数制中每一个固定位置对应的单位值称为“权”，一个数码处在不同位置所代表的值不同，例如，十进制中数字5在十位数位置上表示“50”，在百位数上表示“500”，而在小数点后第1位则表示“0.5”，可见每个数码所代表的真正数值等于该数码乘以一个与数码所在位置相关的常数，这个常数就叫位权。位权的大小是以基数为底幂的形式，数码所在位置的序号为指数的整数次幂，其中位置序号的排列规则在小数点左边，从右向左分别为0, 1, ...；在小数点右边，从左向右分别为-1, -2, ...

以十进制为例，十进制的个位数位置的位权为 10^0 ，十位数位置的位权为 10^1 ，小数点后第1位的位权为 10^{-1} 。十进制数12345.678的值等于 $1 \times 10^4 + 2 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 6 \times 10^{-1} + 7 \times 10^{-2} + 8 \times 10^{-3}$ 。十进制的基数 R 为10，十进制数“权”的一般形式为 10^n ($n = \dots, 1, 0, -1, -2, \dots$)。

3.1.2 计算机内部采用二进制的原因

(1) 技术实现简单：计算机是由逻辑电路组成的，逻辑电路通常只有两种状态，使用0和1进行计数。对于物理元器件而言，一般也都具有两种稳定状态，例如，开关的接通与断开，二极管的导通与截止，电平的高与低等，这些都可以用0和1两个数码来表示，假如采用十进制数，制造具有10种稳定状态的电子元器件是非常困难的。

(2) 简化运算规则：两个二进制数和、积运算组合各有三种，运算规则简单，有利于简化计算机内部结构，提高运算速度。

(3) 适合逻辑运算：逻辑代数是逻辑运算的理论依据，二进制只有两个数码 1 和 0，正好与逻辑代数中的“真”和“假”相吻合，可以很自然地地进行逻辑运算。

(4) 易于进行转换：二进制数与十进制数之间易于互相转换。

(5) 用二进制表示数据具有抗干扰能力强、可靠性高等优点。因为每位数据只有高、低两种状态，当受到一定程度的干扰时，仍能可靠地分辨出它是高还是低。

3.1.3 计算机中的常用数制

人们在日常生活中常使用的时间、消费的钱币等是存在进制关系的。计算机中有十进制系统 (Decimal System)，即有 10 个数 0、1、2、3、4、5、6、7、8、9，也存在二进制、八进制和十六进制。在计算机内部均用二进制数来表示各种信息，但计算机与外部的交互仍采用人们熟悉和便于阅读的形式，它们的转换则由计算机系统的软、硬件来实现。

1. 二进制

在现代电子计算机中，无论是什么类型的信息（数字、文本、图形、图像、音频、视频等），在计算机内部都采用二进制数形式表示，即采用 0 和 1 表示的二进制进行计数，基数为 2，如二进制数 1010 可以表示为 $(1010)_2$ 。

2. 八进制和十六进制

计算机使用二进制数进行各种算术运算和逻辑运算虽然有计算速度快、简单等优点，但也存在一些不足。在一般情况下，使用二进制数表示信息需要占用更多的位数，如十进制数 11，对应的二进制数为 1011，占 4 位。因此，为了方便读写，人们又发明了八进制和十六进制。

八进制基数为 8，使用数字 0, 1, ..., 7 共 8 个数字来表示，运算时逢八进一。

十六进制基数为 16，使用数字 0, 1, ..., 9, A, B, ..., F 共 16 个数字和字母来表示，运算时逢十六进一。

为了区别这几种数制表示方法，通常会在数字后面加一个缩写的大写字母，或者将要表示的数用圆括号括起来，然后用进制下标来标识，如表 3-1 所示。

表 3-1 数制表示方法

类别	基数	使用基本符号	字母标识	书写格式	英文单词
二进制数	2	0, 1	B	$(1001)_2$ 或 1001B	Binary
八进制数	8	0, 1, ..., 7	O	$(1001)_8$ 或 1001O	Octal
十进制数	10	0, 1, ..., 9	D	$(1001)_{10}$ 或 1001D	Decimal
十六进制数	16	0, 1, ..., 9, A, B, ..., F	H	$(1001)_{16}$ 或 1001H	Hexadecimal

3.2 进制转换

3.2.1 R 进制数转换成十进制数

任意 R 进制数可以按其位权方式进行展开。若 L 有 n 位整数，m 位小数，则其各位数为： $(K_{n-1}K_{n-2} \cdots K_0.K_{-1}K_{-2} \cdots K_{-m})$ ，L 可以表示为：

$$L = \sum_{i=-m}^{n-1} K_i R^i = K_{n-1} R^{n-1} + K_{n-2} R^{n-2} + \cdots + K_0 R^0 + K_{-1} R^{-1} + K_{-2} R^{-2} + \cdots + K_{-m} R^{-m}$$

当一个 R 进制数按位权展开后,也就得到了该数值所对应的十进制数。所以,当 R 进制数转换为十进制数时,采用按位权展开各项相加的法则。

【例 3-1】将二进制数 10110.11B 转换成对应的十进制数。

$$(10110.11)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = (22.75)_{10}$$

【例 3-2】将八进制数 45.7O 转换成对应的十进制数。

$$(45.7)_8 = 4 \times 8^1 + 5 \times 8^0 + 7 \times 8^{-1} = (37.86)_{10}$$

【例 3-3】将十六进制数 9B.4H 转换成对应的十进制数。

$$(9B.4)_{16} = 9 \times 16^1 + 11 \times 16^0 + 4 \times 16^{-1} = (155.25)_{10}$$

3.2.2 十进制数转换成 R 进制数

十进制数转换成 R 进制数,应该把十进制数分为整数部分和小数部分分别转换。

整数部分的转换法则是除以 R 取余法;小数部分的转换法则是乘以 R 取整法。

对于整数 L ,我们可以表示为:

$$L = K_{n-1} R^{n-1} + K_{n-2} R^{n-2} + \cdots + K_0 R^0$$

其中 K_i 表示除以 R 得到的各位余数。

对于小数 L ,我们可以表示为:

$$L = K_{-1} R^{-1} + K_{-2} R^{-2} + \cdots + K_{-m} R^{-m}$$

其中 K_i 表示乘以 R 得到的各位整数。

【例 3-4】将十进制数 35.625D 转换为二进制数。

整数部分转换: 35 除以 2 取各位余数。

	除以 R	取余数	对应二进制位数
2 35	35 ÷ 2 = 17	1	K_0 最低位
2 17	17 ÷ 2 = 8	1	K_1
2 8	8 ÷ 2 = 4	0	K_2
2 4	4 ÷ 2 = 2	0	K_3
2 2	2 ÷ 2 = 1	0	K_4
2 1	1 ÷ 2 = 0	1	K_5 最高位
0			

所以 35D=100011B。

注意: 当除以 R 的商为 0 时,应停止取余操作。先得到的余数作为低位,后得到的余数作为高位。

小数部分转换: 0.625 乘以 2 取各位上的整数。

	乘以 R	取整数	对应二进制位数
0.625 × 2 — 1.250	0.625 × 2 = 1.250	1	K_{-1} 小数点后最高位
0.250 × 2 — 0.500	0.25 × 2 = 0.5	0	K_{-2}
0.500 × 2 — 1.000	0.5 × 2 = 1.0	1	K_{-3} 小数点后最低位

所以 0.625D=0.101B。

注意: 在转换小数部分时,当乘以 R 后小数部分为 0 时,或满足某些精度要求时,应停止取

整操作。先得到的整数作为高位，后得到的整数作为低位。另外，取走的整数部分不再参与下次乘法运算。

最后将整数部分和小数部分的转换结果相加，得到 $35.625D=100011.101B$ 。

【例 3-5】 将十进制数 $42.425D$ 转换为八进制数，精确到小数点后两位。

整数部分转换：

8	42		
	8	5	
		0	
		商	

2	↑ 低
5	↑ 高
余数	

所以 $42D=52O$ 。

小数部分转换：

0.425			
× 8			
3.400			
0.400			
× 8			
3.200			
0.200			
× 8			
1.600			
积			

3	↑ 高
3	↑ 高
1	↓ 低
整数	

所以 $0.425D=0.33O$ 。

注意：精确到小数点后两位，要求出小数点后的第三位，然后按照舍入规则进行取舍（八进制 3 舍 4 入，二进制 0 舍 1 入，十六进制 7 舍 8 入）。

所以 $42.425D=52.33O$ 。

【例 3-6】 将十进制数 $246.325D$ 转换为十六进制数，精确到小数点后两位。

整数部分转换：

16	246		
	16	15	
		0	
		商	

6	↑ 低
F	↑ 高
余数	

所以 $246D=F6H$ 。

小数部分转换：

所以 $0.325D=0.53H$ 。

所以 $246.325D=F6.53H$ 。

3.2.3 二进制数、八进制数和十六进制数的相互转换

1. 二进制数与八进制数的转换

由于 $2^3=8$ ，三位二进制数正好可以用一位八进制数表示，所以，只要把每三位二进制数码转换成相应的八进制数码即可。基本法则是整数部分以小数点为界从右向左，每三位为一组进行转换，小数部分从小数点开始，从左向右，每三位为一组进行转换。整数部分不足三位一组者，左边补 0，小数部分不足三位一组者，右边补 0。

若是八进制数转换成二进制数，则只要把八进制数的每一位数码用相应的三位二进制数码表

示出来,排列在一起就是这个八进制数的二进制表示。

【例 3-7】二进制数 10101101.101B 转换成八进制数。

$$10101101.101B = \underline{010} \ \underline{101} \ \underline{101} . \underline{101} B = 255.5O$$

【例 3-8】将八进制数 255.6O 转换成二进制数。

$$\underline{2} \ \underline{5} \ \underline{5} . \underline{6} O = \underline{010} \ \underline{101} \ \underline{101} . \underline{110} B = 10101101.11B$$

2. 二进制数与十六进制数的转换

与八进制和二进制之间的转换类似,由于 $2^4=16$,四位二进制数正好可以用一位十六进制数表示,所以,只要把每四位二进制数码转换成相应的十六进制数码即可。基本法则是,整数部分以小数点为界从右向左,每四位为一组进行转换,小数部分从小数点开始,从左向右,每四位为一组进行转换,整数部分不足四位一组者,左边补 0,小数部分不足四位一组者,右边补 0。

若是十六进制数转换成二进制数,则只要把十六进制数的每一位数码用相应的四位二进制数码表示出来,排列在一起就是这个十六进制数的二进制表示。

【例 3-9】将二进制数 11011010.011B 转换成十六进制数。

$$11011010.011B = \underline{1101} \ \underline{1010} . \underline{0110} B = DA.6H$$

【例 3-10】将十六进制数 B9C.AH 转换成二进制数。

$$\underline{B} \ \underline{9} \ \underline{C} . \underline{A} H = \underline{1011} \ \underline{1001} \ \underline{1100} . \underline{1010} B = 101110011100.101B$$

常用计数制对照表如表 3-2 所示

表 3-2 常用计数制对照表

十进制数	二进制数	八进制数	十六进制数	十进制数	二进制数	八进制数	十六进制数
0	0	0	0	8	1000	10	8
1	1	1	1	9	1001	11	9
2	10	2	2	10	1010	12	A
3	11	3	3	11	1011	13	B
4	100	4	4	12	1100	14	C
5	101	5	5	13	1101	15	D
6	110	6	6	14	1110	16	E
7	111	7	7	15	1111	17	F

3.2.4 二进制数运算

计算机内二进制数可以做两种基本运算:算术运算和逻辑运算。

1. 算术运算

算术运算包括加、减、乘、除,运算规则类似于十进制数运算。

(1) 加法规则: $0+0=0$ 、 $0+1=1$ 、 $1+0=1$ 、 $1+1=10$ (向高位进位)。

【例 3-11】计算二进制数 $(1101)_2 + (1011)_2 = (11000)_2$ 。

$$\begin{array}{r} 1101 \\ + \quad 1011 \\ \hline 11000 \end{array} \quad (\text{向高位进位})$$

(2) 减法规则: $0-0=0$ 、 $0-1=1$ (向高位借位)、 $1-0=1$ 、 $1-1=0$ 。

【例 3-12】计算二进制数 $(1101)_2 - (1011)_2 = (0010)_2$ 。

$$\begin{array}{r}
 1101 \\
 - 1011 \\
 \hline
 0010 \quad (\text{向高位借位})
 \end{array}$$

若被减数小于减数, 则将被减数与减数交换位置, 按上述方法计算后, 在两数的差前面加一个负号。

【例 3-13】计算二进制数 $(1011)_2 - (1101)_2 = -(1101)_2 - (1011)_2 = (-0010)_2$ 。

(3) 乘法规则: $0 \times 0 = 0$ 、 $0 \times 1 = 0$ 、 $1 \times 0 = 0$ 、 $1 \times 1 = 1$ 。

【例 3-14】计算二进制数 $(1101)_2 \times (1011)_2 = (10001111)_2$ 。

$$\begin{array}{r}
 1101 \\
 \times 1011 \\
 \hline
 1101 \\
 1101 \\
 0000 \\
 1101 \\
 \hline
 10001111
 \end{array}$$

(4) 除法规则: $0 \div 1 = 0$ 、 $1 \div 1 = 1$ 。

【例 3-15】计算二进制数 $(110111)_2 \div (101)_2 = (1011)_2$ 。

$$\begin{array}{r}
 101 \overline{) 110111} \\
 \underline{101} \\
 111 \\
 \underline{101} \\
 101 \\
 \underline{101} \\
 0
 \end{array}$$

2. 逻辑运算

逻辑运算包括与、或、非, 是在对应的两个二进制数位之间进行的, 不存在算术运算中的进位或借位情况。

(1) 逻辑与规则: $0 \cap 0 = 0$ 、 $0 \cap 1 = 0$ 、 $1 \cap 0 = 0$ 、 $1 \cap 1 = 1$ 。

【例 3-16】计算二进制数 $(1101)_2 \cap (1011)_2 = (1001)_2$ 。

$$\begin{array}{r}
 1101 \\
 \cap 1011 \\
 \hline
 1001
 \end{array}$$

(2) 逻辑或规则: $0 \cup 0 = 0$ 、 $0 \cup 1 = 1$ 、 $1 \cup 0 = 1$ 、 $1 \cup 1 = 1$ 。

【例 3-17】计算二进制数 $(1101)_2 \cup (1011)_2 = (1111)_2$ 。

$$\begin{array}{r}
 1101 \\
 \cup 1011 \\
 \hline
 1111
 \end{array}$$

(3) 逻辑非规则: $\bar{0} = 1$ 、 $\bar{1} = 0$ 。

3.3 计算机信息编码

3.3.1 计算机中的存储单位

计算机内部均用二进制数来表示各种信息,计算机与外部的交互仍采用人们熟悉和便于阅读的形式,由计算机系统的软、硬件来实现转换。

信息存储的单位有以下3种。

1. 位 (b)

位 (bit) 是计算机内部存储信息的最小单位,1个二进制位只能表示0或1,要想表示更大的数,就得把更多的位组合起来作为一个整体,每增加1位所能表示的信息量就增加1。

2. 字节 (B)

字节 (Byte) 是计算机内部存储信息的基本单位,1个字节由8个二进制位组成,即 $1\text{B}=8\text{b}$ 。

在计算机中,常用的信息存储单位还有千字节 (KB)、兆字节 (MB)、吉字节 (GB) 和太字节 (TB) 等,其中:

$$1\text{KB}=2^{10}\text{B}=1024\text{B}$$

$$1\text{MB}=2^{10}\text{KB}=1024\text{KB}$$

$$1\text{GB}=2^{10}\text{MB}=1024\text{MB}$$

$$1\text{TB}=2^{10}\text{GB}=1024\text{GB}$$

$$1\text{PB}=2^{10}\text{TB}=1024\text{TB}$$

$$1\text{EB}=2^{10}\text{PB}=1024\text{PB}$$

$$1\text{ZB}=2^{10}\text{EB}=1024\text{EB}$$

$$1\text{YB}=2^{10}\text{ZB}=1024\text{ZB}$$

$$1\text{BB}=2^{10}\text{YB}=1024\text{YB}$$

$$1\text{NB}=2^{10}\text{BB}=1024\text{BB}$$

$$1\text{DB}=2^{10}\text{NB}=1024\text{NB}$$

$$1\text{CB}=2^{10}\text{DB}=1024\text{DB}$$

3. 字 (Word)

1个字通常由1个字节或若干个字节组成,是计算机进行信息处理时1次存取、加工和传送的数据长度。字长是衡量计算机性能的重要指标,字长越长,计算机1次所能处理信息的实际位数就越多,运算精度就越高,最终表现为计算机的处理速度越快,常用的字长有8位、16位、32位和64位等。

3.3.2 数值型数据编码

数值在计算机中采用“二进制”方式存储。数值有正、负和大、小之分,为了解决数据的正、负问题,引入数据的原码、反码、补码表示。为了解决数据的表示范围问题,引入数据的定点表示和浮点表示。

1. 真值数与机器数

1) 真值数

在机器外用“+”“-”表示有符号数的正、负,如-6。

2) 机器数

机器数可分为无符号数和带符号数两种,无符号数是指计算机字长的所有二进制位均表示数值;带符号数是指机器数分为符号和数值两部分,且均用二进制数表示,在机器内用“0”表示“正号”,“1”表示“负号”。如真值数+6和-6用8位带符号机器数分别表示为00000110和10000110。

3) 机器数的特点

① 机器字长是有限的,因此由字长决定数的表示范围。机器字长是指以多少个二进制位表示一个数。

② 符号数值化,参与运算。

③ 小数点按约定方式标出,而不是由专门元器件表示。

2. 数的定点和浮点表示

1) 定点数

所有数值数据的小数点隐含在某一个固定位置上,称为定点表示法,简称定点数,通常分为定点小数和定点整数。

① 定点小数。

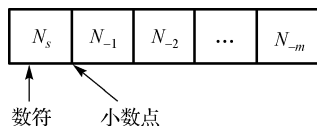


图 3-1 定点小数

指小数点固定在符号位之后,机器中的所有数均为纯小数。任何一个数都可以写成: $N=N_sN_{-1}N_{-2}\cdots N_{-m}$, N_s 表示符号位(数符),如图 3-1 所示。注意,在这种表示数的方法中,小数点紧接在符号位之后,不用明确表示出来,即不占用二进制的位。对 $m+1$ 个二进制位表示的小数,其值的范围为: $|N|\leq 1-2^{-m}$ 。

【例 3-18】 $\pm 0.625D$ 的机器数表示。

数的真值 $\pm 0.625D = \pm 0.101B$

机器数	+0.625	0	1	0	1	0	0	0	0
	-0.625	1	1	0	1	0	0	0	0

② 定点整数。

指小数点固定在最低位之后,机器中的所有数均为整数。整数分为带符号和不带符号两类。带符号整数,符号位仍然在最高位。可以写成: $N=N_sN_NN_{N-1}\cdots N_0$, N_s 表示符号位,如图 3-2 所示。

对 $N+1$ 个二进制位表示的整数,其值的范围为: $|N|\leq 2^N-1$ 。

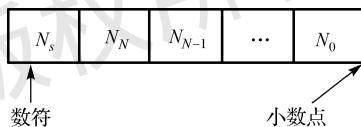


图 3-2 定点整数

对于不带符号的整数,所有的 $N+1$ 个二进制位均看成数值,此时数值表示范围为: $0\leq N\leq 2^{N+1}-1$ 。

由于实际参与运算的数往往既有整数部分又有小数部分,因此必须选取合适的比例因子,把原始の数缩小成纯小数或扩大成纯整数后再进行处理,所得到的运算结果还需要根据比例因子还原成实际的数值,这是很烦琐的。所以定点表示法仅适用于计算较简单且数的范围变化不太大的场合。

2) 浮点数

指小数点的位置不固定的数。与科学计数法相似,任意一个 J 进制数 N ,总可以写成 $N=J^E \times M$,式中 M 称为数 N 的尾数,是一个纯小数; E 为数 N 的阶码,是一个整数,其符号位称为阶符, J 称为比例因子 J^E 的底数。这种表示方法相当于数的小数点位置随比例因子的不同而在一定范围内自由浮动,所以称为浮点表示法。

底数是事先约定好的(常取 2),在计算机中不出现。当在机器中表示一个浮点数时,一是要给出尾数,用定点小数形式表示,尾数部分给出有效数字的位数,从而决定了浮点数的表示精度;二是要给出阶码,用整数形式表示,阶码指明小数点在数据中的位置,从而决定了浮点数的表示范围,浮点数也要有符号位,称为数符。如果用 16 位二进制来表示一个浮点数,则 16 位二进制的分配方式如图 3-3 所示。

15	14-12	11	10-----0
阶符	阶码	数符	尾数

图 3-3 浮点数

【例 3-19】 $N=-35.625=-100011.101B=-0.100011101\times 2^{110}$ 的浮点表示,如图 3-4 所示。

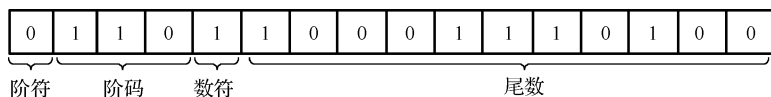


图 3-4 浮点表示

阶码是一个带符号的整数，它用来指示尾数中的小数点应当向左或向右移动的位数。尾数表示数值的有效数字，其本身的小数点约定在数符和尾数之间。

3. 原码、反码和补码

常见的机器数有原码、反码、补码三种不同形式。

原码是指在表示数的时候最高位为符号位，其余各位为数值本身的绝对值。

反码要分两种情况考虑，正数的反码与原码相同；负数的反码符号位为 1，其余位对原码取反。

补码也分两种情况考虑，正数的原码、反码、补码相同；负数的补码最高位为 1，其余位为原码取反，再对整个数加 1。

【例 3-20】求 -7 的原码、反码、补码。

-7D=10000111B

原码：10000111

反码：11111000 在原码的基础上符号位不变，其余各位取反。

补码：11111001 在反码的基础上加 1，得到补码。

注意：对于 0 这个数字来讲，也分为正数 0 和负数 0 两种，它们的原码、反码是不同的。若以 $N+1$ 位二进制位表示一个数，若按照原码的形式表示数，则该数所表示的范围为： $-2^N+1 \sim 2^N-1$ ；若按照补码的形式表示数，则该数所表示的范围为： $-2^N \sim 2^N-1$ 。

特别规定：-128 的补码为 10000000，所以有符号字节的补码表示范围为： $-128 \sim 127$ ，-128 不在表示范围之内，所以没有反码。

(1) 原码：把真值数的符号位用“0”表示正号，“1”表示负号。

【例 3-21】+6 的原码为：00000110

-6 的原码为：10000110

(2) 反码：正数的反码与原码相同，负数的反码即除符号位之外其他位按位取反而成。

【例 3-22】+6 的反码为：00000110

-6 的反码为：11111001

(3) 补码：正数的补码与原码相同，负数的补码为将它的反码加 1。

【例 3-23】+6 的补码为：00000110

-6 的补码为：11111010

3.3.3 非数值信息编码

所谓非数值信息，通常是指字符、图像、音频、视频等信息，在计算机中用得最多的非数字信息是文本字符，字符又可以分为汉字字符和非汉字字符。非数值信息通常不用来表示数值的大小，在计算机内部都采用了某种编码标准，通过编码标准可以把其转换成 0、1 代码串进行处理，计算机将这些信息处理完毕再转换成可视的信息显示出来。

1. 西文字符编码

字符是计算机中使用最多的信息形式之一，是人与计算机进行通信、交互的重要媒介。字符的集合称为“字符集”。西文字符集由字母、数字、标点符号和一些特殊符号组成。在计算机中，要为每个字符指定一个确定的编码，作为识别与使用这些字符的依据。各种字母和符号也必须使

用规定的二进制码表示，计算机才能处理。

在西文领域，目前普遍采用的是 ASCII 码(American Standard Code for Information Interchange, 美国标准信息交换码)，ASCII 码虽然是美国国家标准，但它已被国际标准化组织（ISO）定为国际标准，并在全世界范围内通用，作为国际通用的信息交换标准代码，对应的国际标准是 ISO 646。ASCII 码有 7 位 ASCII 码和 8 位 ASCII 码两种。

标准的 ASCII 码是 7 位码，用一个字节表示最高位是 0，可以表示 128 (2⁷) 个字符。前 32 个码和最后一个码通常是计算机系统专用的，代表一个不可见的控制字符。数字字符 0~9 的 ASCII 码是连续的，为 30H~39H (H 表示十六进制数)；大写英文字母 A~Z 和小写英文字母 a~z 的 ASCII 码也是连续的，分别为 41H~5AH 和 61H~7AH。因此知道一个字母或数字的 ASCII 码，就很容易推算出其他字母和数字的 ASCII 码，如表 3-3 所示。

表 3-3 ASCII 码表

低 4 位		高 3 位							
		0	1	2	3	4	5	6	7
		000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	.	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	\$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	ST	US	/	?	O	↓	o	DEL

8 位 ASCII 码称为扩展的 ASCII 码字符集，由于 7 位 ASCII 码只有 128 个字符，在很多应用中无法满足要求，为此国际标准化组织又制定了 ISO 2002 标准，它规定了在保证与 ISO 646 兼容的前提下，将 ASCII 码字符扩充为 8 位编码的统一方法，8 位 ASCII 码可以表示 256 个字符。

ASCII 码的每个字符用 7 位二进制表示，其排列次序为 $d_6d_5d_4d_3d_2d_1d_0$ ， d_6 为高位。但是一个字符在计算机内部实际上是用 8 位表示的。所以，最高一位 d_7 设置为“0”。如果需要奇偶校验，这一位可用于存放奇偶校验的值，此时称这一位为校验位。

00H~1FH 段的 32 个代码是对控制符的编码。一个控制符代表一种操作。例如，“CR”代表“回车”操作，在键盘上按下回车键，将代码 0DH 送入主机。

20H 是对“空格”的编码，“空格”是字符，而且在文字之间是可见的字符。

2. 中文字符编码

由于汉字是象形文字，具有字形结构复杂、重音字和多音字多等特殊性，因此汉字的输入、

存储、处理及输出过程中所使用的汉字编码是不相同的,其中包括用于汉字输入的输入码,用于机内存储和处理的机内码和用于输出显示和打印的字模点阵码(或称字形码)。

1) 汉字的输入码

汉字的输入码是为了利用现有的计算机键盘,将形态各异的汉字输入计算机而编制的代码。目前在我国推出的汉字输入编码方案很多,其表示形式大多使用字母、数字或符号。编码方案大致可以分为,以汉字发音进行编码的音码,如全拼码、简拼码、双拼码等;以汉字书写的形式进行编码的形码,如五笔字型码等。

2) 汉字的机内码

汉字的机内码是供计算机系统内部进行存储、加工处理、传输等统一使用的代码,又称为汉字内部码或汉字内码。不同的系统使用的汉字机内码有所不同。使用最广泛的是一种 2B(2 个字节)的机内码,俗称变形的国标码。其最大优点是表示简单,且与交换码之间有明显的对应关系,同时也解决了中西文机内码存在二义性的问题。

3) 汉字的字形码

汉字的字形码是汉字字库中存储的汉字字形的数字化信息,用于汉字的显示和打印。汉字字形的产生方式大多是数字式,即以点阵方式形成汉字。因此,汉字字形码主要是指汉字字形点阵的代码。汉字字形点阵有 16×16 点阵、24×24 点阵、32×32 点阵、64×64 点阵等。如“春”字的 24×24 点阵表示形式,如图 3-5 所示。一个汉字方块中,行数、列数分得越多,描绘的汉字也就越精确,但占用的存储空间也就越大。

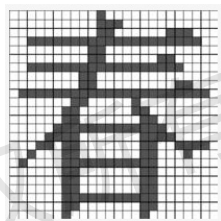


图 3-5 “春”字的 24×24 点阵表示形式

4) 国标码

国标码就是国家标准汉字编码 GB 2312—80 所规定的机器内部编码,代表中文简化字,在我国广泛使用。其用于汉字信息处理系统之间或者通信系统之间交换信息,因此,又称为汉字交换码。国标码采用 ASCII 码表中的可显示字符的代码 21H~7EH 作为汉字的区码和位码,构成 94×94 的矩阵,对收录的 6763 个汉字、682 个西文字符和图符进行编码。矩阵的每行称为“区”,每列称为“位”。国标码规定每个汉字用 2 个字节二进制编码表示,每个字节的最高位为 0,其余 7 位用于表示汉字信息。如汉字“啊”的国标码为 00110000 00100001。

5) 区位码

国标码是一个 4 位十六进制数,区位码是一个 4 位的十进制数,每个国标码或区位码都对应着一个唯一的汉字或符号,但因为十六进制数很少用到,所以常用的是区位码,它的前两位叫区码,后两位叫位码。

在 Windows 中常用 Ctrl+空格组合快捷键和 Ctrl+Shift 组合快捷键调出区位码。如“2901”代表“健”字,“4582”代表“万”字,“8150”代表“楮”字,这些都是汉字,用区位码还可以很轻松地输入特殊符号,如“0189”代表“※”(特殊符号),“0528”代表“ゼ”(日文),“0711”代表“Й”(俄文),“0949”代表“十”(制表符)。

6) UCS 编码

为了统一表示世界各国的文字,1993 年我国国家标准采用国际标准化组织公布的“通用多八位编码字符集”的国际标准 ISO/IEC 10646,简称 UCS(Universal Code Set)。UCS 包含了中、日、韩等国的文字,这一标准为包括汉字在内的各种正在使用的文字规定了统一的编码方案。该标准是用 4 个字节来表示每个字符,并相应地指定组、平面、行和字位。

整个编码字符集包含 128(1 个字节的低 7 位即 $2^7=128$) 个组,其中每个组表示 256($2^8=256$)

个平面, 每个平面包含 256 行, 每行有 256 个字位。4 个字节共 32 位, 足以包容世界上所有的字符, 同时也符合现代处理系统的体系结构要求。

第 1 个平面 (00 组中的 00 平面) 称为基本多文种平面, 它包含字母文字、音节文字及表意文字等, 其分成 4 个区。

① A 区: 代码位置 0000H~4DFFH (19903 个字位), 用于字母文字、音节文字及各种符号。

② I 区: 代码位置 4E00H~9FFFH (20992 个字位), 用于中、日、韩统一的表意文字。

③ O 区: 代码位置 A000H~DFFFH (16384 个字位), 留作未来标准化用。

④ R 区: 代码位置 E000H~FFFDH (8190 个字位), 作为基本多文种平面的限制使用区, 它包括专用字符、兼容字符等各种符号。

如汉字“大”的国标码为 3473H, UCS 编码为 00005927H, 即在 00 组, 00 面, 59H 行, 第 27H 字位上。

3. Unicode 编码

随着因特网的迅速发展, 人们进行信息交换的需求越来越大, 不同的编码成为信息交换的障碍, 于是 Unicode 编码应运而生。Unicode 编码是由国际标准化组织于 20 世纪 90 年代初制定的一种字符编码标准, 它用多个字节表示 1 个字符, 几乎所有的书面语言都能够用单一的 Unicode 编码表示。前 128 个 Unicode 字符是标准 ASCII 码字符, 接下来是 128 个扩展的 ASCII 码字符, 其余的字符供不同的语言使用。在 Unicode 中, ASCII 码字符也用多个字节表示, 这样 ASCII 码字符与其他字符的处理就统一起来了, 大大简化了处理的过程。

4. 图形和图像的表达

图形是由计算机绘图工具绘制的图形, 图像是由数码相机或扫描仪等输入设备捕捉的实际场景画面, 通常可以将图形和图像统称为图像。在计算机中图像常采用位图图像或矢量图像两种表示方法。



图 3-6 位图图像表示

1) 位图图像

计算机屏幕图像是由一个个像素点组成的, 将这些像素点的信息有序地储存到计算机中, 用来保存整幅图的信息, 这种图像文件类型叫位图图像, 如图 3-6 所示。

对于黑白图像只有黑白两种颜色, 计算机只要用 1 位 (1bit) 数据即可记录 1 个像素的颜色, 用 0 表示黑色, 1 表示白色。若增加表示像素的二进制数的位数, 则能够增加计算机表示的灰度度。例如, 计算机用 1 个字节 (8 位) 数据记录 1 个像素的颜色, 则从 00000000 (纯黑) 到 11111111 (纯白) 可以表示 256 色灰度图像。

对于彩色图像, 每个像素的颜色用红 (R)、绿 (G)、蓝 (B) 三原色的强度表示, 若每个颜色的强度用 1 个字节来表示, 则每种颜色包括 256 个强度级别, 强度从 00000000 到 11111111。因此, 描述每个像素需要 3 个字节, 该像素的颜色是三种颜色的复合结果。例如, 11111111 (R)、00000000 (G)、00000000 (B) 为红色, 11111111 (R)、11111111 (G)、00000000 (B) 为黄色, 11111111 (R)、11111111 (G)、11111111 (B) 为白色。

常见的位图图像文件类型有 .bmp、.pcx、.gif、.jpg、.tif、.psd 和 .cpt 等, 同样的图形以不同类型的文件保存时, 文件大小也会有所差别。

位图图像能够制作出颜色和色调变化丰富的图像, 可以逼真地表现出自然界的景观, 广泛应用于照片和绘图图像中, 而且很容易在不同软件之间交换文件。其缺点是无法制作真正的三维图像, 并且图像在缩放、旋转和放大时会产生失真现象, 同时文件较大, 对内存和硬盘空间容量的

需求也较高。

2) 矢量图像

矢量图像是用一组指令集合来描述图像的内容,包括描述构成该图像的所有直线、圆、圆弧、矩形、曲线等图元的位置、维数和形状。

矢量图像所占的存储容量较小,可以很容易地进行放大、缩小和旋转等操作,并且不会失真,适合用于表示线框型的图画、工程制图、美术字和三维建模等方面。但是矢量图像不易制作色调丰富或色彩变化太多的图像。

常见的矢量图图像文件类型有.ai、.eps、.svg、.dwg、.dxf、.wmf和.emf等。

5. 音频的表示

音频用于表示声音和音乐,音频本身是模拟信号,是连续的,不适合在计算机中存储,需要对其离散化。首先需要对其采样,采样就是以相等的间隔来测量信号的值;然后再量化采样值,就是给采样值分配值,例如,如果一采样值为34.2,而值集为0到63的整数值,那么将该采样值量化为值34;最后将量化值转换为二进制并存入计算机。

常见的音频格式有.wav、.midi、.mp3、.au、.wma等。

6. 视频的表示

视频实际上是由一系列的静态图像组成的动态图像,其中每幅静态图像称为帧。若组成动态图像的每帧图像是由人工或计算机加工而成的,则称为动画。若组成动态图像的每帧图像是通过实时摄取自然景象或活动对象而成的,则称为视频。

视频文件是将静态图像运用位图的形式有序储存,但这样数据量太大,因此现在的视频文件大多采用了视频压缩技术。总体而言,空间冗余性可以借由“只记录单帧画面的一部分与另一部分的差异性”来降低,这种技巧称为帧内压缩(Intraframe Compression)。而时间冗余性则可借由“只记录两帧不同画面间的差异性”来降低,这种技巧被称为帧间压缩(Interframe Compression),包括运动补偿及其他技术。

根据所采用的压缩编码技术的不同,常见的视频格式有MPEG/MPG/DAT、AVI、RA/RM/RAM、MOV、WMV、RMVB、FLV等。

7. 动画的表示

动画是通过把人物的表情、动作、变化等分解后画成许多动作瞬间的画幅,再用摄影机连续拍摄成一系列画面,给人们的视觉带来连续变化的图画感。它的基本原理与电影、电视一样,都是视觉暂留原理。

按工艺技术可将动画分为:平面手绘动画、立体拍摄动画、虚拟生成动画、真人结合动画;

按传播媒介可将动画分为:影院动画、电视动画、广告动画、科教动画;

按动画性质可将动画分为:商业动画、实验动画。

动画由于应用领域不同,也存在着不同的存储格式,常见的有.gif、.swf、.mov、.fli、.flc、.mov等格式。

动画和视频经常被认为是多媒体技术中重要的媒体形式,主要是缘于它们都属于“动态图像”的范畴。动态图像是连续渐变的静态图像或者图形序列,沿时间轴顺序更换显示,从而产生运动视觉感受的媒体形式。

然而,动画和视频事实上是两个不同的概念。动画的每帧图像都是由人工或计算机产生的。根据人眼的特性,用15~20帧/秒的速度顺序地播放静止图像帧,就会产生运动的感觉。视频的每帧图像都是通过实时摄取自然景象或者活动对象获得的。视频信号可以通过摄像机、录像机等连续图像信号输入设备来产生。

3.4 自主实践

3.4.1 Windows 附带应用程序——写字板和记事本的使用

一、预习内容

- (1) 学习写字板和记事本的启动、退出。
- (2) 了解文档的创建、打开、保存及窗口的组成。

二、实践目的

- (1) 掌握写字板和记事本的启动和退出方法。
- (2) 熟悉写字板和记事本的功能界面。
- (3) 掌握写字板和记事本的基本操作。

三、实践内容

(1) “写字板”是一个使用简单，功能强大的文字处理程序，用户可以利用它在日常工作中进行文件的编辑。它不仅可以进行中英文文档的编辑，而且还可以进行图文混排，插入图片、声音、视频等多媒体资料。

实践内容如下：

单击“开始”按钮，选择“所有程序”→“附件”→“写字板”命令，即可打开默认文件名为“文档.rtf”的写字板文档，输入文字内容，完成效果。

(2) “记事本”用于纯文本文档的编辑，功能没有写字板强大，适合编写一些篇幅短小的文件。在记事本中用户可以使用不同的语言格式创建文档，而且可以用不同的格式打开或保存文件。

实践内容如下：

单击“开始”按钮，选择“所有程序”→“附件”→“记事本”命令，即可打开默认文件名为“无标题.txt”的记事本文档，粘贴复制的内容，自行测试相应功能。

3.4.2 Windows 附带应用程序——画图工具和媒体播放器的使用

一、预习内容

- (1) 学习画图工具和媒体播放器的启动、退出。
- (2) 了解画图工具和媒体播放器的基本操作。

二、实践目的

- (1) 掌握画图工具和媒体播放器的启动、退出方法。
- (2) 熟悉画图工具和媒体播放器的功能界面。
- (3) 掌握画图工具和媒体播放器的基本操作。

三、实践内容

1. 使用画图工具编辑文件

在 Windows 7 中，画图工具的界面和功能得到了进一步提升，可用于在空白绘图区域或在现有图片上创建绘图。下面简单介绍一下 Windows 7 系统中画图工具的功能。

- (1) 画图预先设置了可插入的图形的形状区。
- (2) 加入图片后还可以加入文字，而且字体、大小、颜色都可以调整。
- (3) 画图工具可以修改照片的尺寸。

(4) 画图工具最常用的功能是配合键盘上的截图抓屏键保存当时的屏幕显示，按 **PrtScn SysRq** 键保存整个屏幕，按 **Alt+ PrtScn SysRq** 组合快捷键抓取当前活动窗口的区域。

读者可自行选定内容完成一幅作品并加工美化。

2. 使用媒体播放器编辑文件

Windows 7 系统的强大功能和酷炫界面深受用户们的喜欢,其自带的一些实用功能更是给用户带来了方便。Windows 7 系统中自带多媒体播放器 Windows Media Player,它能够将计算机变身为媒体工具,实现本地媒体的播放。

单击“开始”按钮,选择“所有程序”→“Windows Media Player”命令,即可打开应用程序,读者可自行熟悉程序功能。

3.5 拓展实训

使用计算器进行各种进制的计算。单击“开始”按钮,选择“所有程序”→“附件”→“计算器”命令,系统默认启动的计算器为标准型计算器。单击“查看”菜单,里面提供了标准型、科学型、程序员、统计信息 4 种模式,下面还有基本、单位转换、日期计算、工作表 4 种功能。

科学型计算器的功能比标准型计算器的功能更加强大,它不仅可以完成标准型计算器的计算操作,还能进行进制、弧度、角度转换及三角函数等高级运算。

习题 3

1. 十进制算术表达式: $3 \times 512 + 7 \times 64 + 4 \times 8 + 5$ 的运算结果用二进制数表示为 ()。
A. 10111100101 B. 11111100101
C. 11110100101 D. 11111101101 11111100101
2. 与二进制数 101.01011 等值的十六进制数为 ()。
A. A.B B. 5.51 C. A.51 D. 5.58
3. 与十进制数 2004 等值的八进制数为 ()。
A. 3077 B. 3724 C. 2766 D. 4002
4. $(2004)_{10} + (32)_{16}$ 的结果是 ()。
A. $(2036)_{10}$ B. $(2054)_{16}$ C. $(4006)_{10}$ D. $(100000000110)_2$
5. 与十进制数 2006 等值的十六进制数为 ()。
A. 7D6 B. 6D7 C. 3726 D. 6273
6. 与十进制数 2003 等值的二进制数为 ()。
A. 11111010011 B. 10000011 C. 110000111 D. 1111010011
7. $(2008)_D - (3723)_O$ 的结果是 ()。
A. $(-1715)_D$ B. $(5)_D$ C. $(-5)_H$ D. $(111)_B$
8. 100 个 24×24 点阵的汉字字模信息所占用的字节数是 ()。
A. 2400 B. 7200 C. 57600 D. 73728
9. 已知英文大写字母 D 的 ASCII 码值是 44H,那么英文大写字母 F 的 ASCII 码值为十进制数 ()。
A. 46 B. 68 C. 70 D. 15
10. 一汉字的机内码是 B0A1H,那么它的国标码是 ()。
A. 3121H B. 3021H C. 2131H D. 2130H



(习题答案)