

第3章

计算机中数据的表示与编码

自然界的的信息是丰富多彩的,数值、字符、图形、图像、音频、视频等都是计算机处理的对象。但是本质上,计算机只能处理“0”、“1”组成的二进制代码。因此,无论何种数据或信息,最终均要转换成二进制形式进行处理。其表现主要体现在两方面:一是输入计算机的各种数据信息需要从其自然形式转换为二进制代码形式,供计算机处理;另一方面是,计算机中输出的数据需要进行逆向转换,转换成大众所接受的自然形式。本章主要介绍常用的数制及其转换方式以及各类数据信息在计算机中的表示方法。

3.1 计算机中的数制及转换

计算机系统内部广泛使用的是二进制数,即运算数只有“0”和“1”两个,与人们熟识的十进制相比,无论是运算数还是运算法则均大大减少。从而在计算机的物理设计上,二进制数运算的电路设计上远远比十进制简单。同时,二进制数据的“0”和“1”的两种状态,描述电子元器件的电压高低、晶体管导通截止、电容充放等十分吻合,也便于表示逻辑的“真”、“假”,方便逻辑运算。虽然十进制仍在人们生活中占据十分重要的地位,但在计算机中,IBM 公司推出的 Power 6 处理器中,首次实现了在 CPU 内部采用硬件进行十进制计算,目前仍难以看到这种设计的用途和优势。

3.1.1 数制

数制,也称计数制,是指用一组固定的符号和统一的规则来表示数值的方法。数制为非进位计数制和进位计数制两种,前者数值的大小与它在数中位置无关,而后者相关。本书讨论的都是进位计数制。例如,日常生活中用到最多的是十进制,其原则是逢十进一,借一当十;一周为七天,是七进制数,逢七进一;时间上的秒和分钟为六十进制,逢六十进一。进位计数制的基本要素是基数和位权。

(1) 基数。基数即该数制中所有的 R 个基本整数符号 $0 \sim R-1$ 。十进制数的基数为 $0 \sim 9$,十六进制数的基数为 $0 \sim 15$,二进制为 $0 \sim 1$ 。

(2) 位权。顾名思义,每位上的权值,十进制数个位权值为 10^0 ,十位上位权值为 10^1 ,百位上位权值为 10^2 等。可见,每个基数符号在固定位置上的计数单位为位权,它是以小

数点为分界线,每一位上的幂次级分别为 $\cdots 3, 2, 1, 0, -1, -2, -3 \cdots$ 。对于 R 进制数,整数部分第 i 位上的位权值 $R^{(i-1)}$, 小数部分第 i 位位权值为 R^{-i} 。

任何一种进位计数制类型的数均可表示为按其权展开的多项式之和,即“各位上的基数与其位权值乘积的连加形式”。

例如,十进制数中 123.45 可表示为:

$$(123.45)_{10} = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$$

3.1.2 数制转换

计算机内部采用二进制数,然而当表示一个大数时,位数太多,不便人们快速获取信息。因此,方便起见经常采用八进制或十六进制作为二进制的缩写形式。八进制的基数为 $0 \sim 7$,十六进制的基数为 $0 \sim 15$,其中 $10 \sim 15$ 无法在一位上显示所以用英文大写字母 A~F 代替。

1. R 进制数与十进制数的互化

(1) 将 R 进制数转换为十进制数。将 R 进制数转换为十进制数,只要把它们以按权展开的幂级数多项式和形式展开并进行计算,所得结果就是十进制数。

【例 3.1】 把二进制数 $(1101.01)_2$ 转换为十进制数。

$$\begin{aligned}(1101.01)_2 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= 8 + 4 + 0 + 1 + 0 + 0.25 \\ &= (13.25)_{10}\end{aligned}$$

【例 3.2】 把八进制数 $(3671.65)_8$ 转换为十进制数。

$$\begin{aligned}(3671.65)_8 &= 3 \times 8^3 + 6 \times 8^2 + 7 \times 8^1 + 1 \times 8^0 + 6 \times 8^{-1} + 5 \times 8^{-2} \\ &= 1536 + 384 + 56 + 1 + 0.75 + 0.078125 \\ &= (1977.828125)_{10}\end{aligned}$$

注意: 小数部分 8^{-1} 为 0.75 而非 0.8, 8^{-2} 为 0.078125 而非 0.08。

【例 3.3】 把十六进制数 $(3DA.6C)_{16}$ 转换为十进制数。

$$\begin{aligned}(3DA.6C)_{16} &= 3 \times 16^2 + 13 \times 16^1 + 10 \times 16^0 + 6 \times 16^{-1} + 12 \times 16^{-2} \\ &= 768 + 208 + 10 + 0.3075 + 0.046875 \\ &= (986.421875)_{10}\end{aligned}$$

(2) 十进制数转换成 R 进制数。十进制数转换为 R 进制数时,对整数部分和小数部分分别处理。整数部分“除 R 取余逆序排”,小数部分“乘 R 取整,正序取”。整数部分除至商为 0 结束取余过程,小数部分取整取至小数部分为 0 或达到精度要求为止。对于 R 进制数,其基数为 $0 \sim R-1$,整数部分除以基数后所得的余数范围,不大于 R ,即 $0 \sim R-1$;而小数部分均为纯小数,乘以基数 R 后取整数的范围,也不大于 R ,即 $0 \sim R-1$ 。

【例 3.4】 把十进制数 $(986.422)_{10}$ 转换成十六进制数(小数部分保留两位有效数字)。

整数部分：

$$\begin{array}{r|l} 16 & 986 \\ \hline 16 & 61 \quad \cdots 10 \\ 16 & 3 \quad \cdots 13 \\ & 0 \quad \cdots 3 \end{array} \quad \uparrow$$

小数部分：

$$\begin{array}{r|l} & 0.422 \\ \times & 16 \\ \hline & 6.752 \rightarrow 6 \\ & 0.752 \\ \times & 16 \\ \hline & 12.032 \rightarrow 12 \end{array} \quad \downarrow$$

结果为 $(986.422)_{10} \approx (3DA.6C)_{16}$

2. 二、八、十六进制数之间的互化

八进制和十六进制能成为二进制的缩写形式，是因为 $8=2^3$ ，八进制的基数为 $0\sim7$ 可由三位二进制数表示。根据位权的原理，三位二进制数上的位权分别为 2^2 、 2^1 、 2^0 ，即 4、2、1。每位上只能为二进制的“0”或“1”，利用排列组合的关系可知其能表示 $0\sim7$ 共 8 个基数，也即八进制的所有基数。同理， $16=2^4$ ，四位二进制数可表示成一位十六进制数。对应关系如图 3-1 所示。

八进制	位权值 4 2 1			十六进制	位权值 8 4 2 1				十六进制	位权值 8 4 2 1			
0	0	0	0	0	0	0	0	0	8	1	0	0	0
1	0	0	1	1	0	0	0	1	9	1	0	0	1
2	0	1	0	2	0	0	1	0	A	1	0	1	0
3	0	1	1	3	0	0	1	1	B	1	0	1	1
4	1	0	0	4	0	1	0	0	C	1	1	0	0
5	1	0	1	5	0	1	0	1	D	1	1	0	1
6	1	1	0	6	0	1	1	0	E	1	1	1	0
7	1	1	1	7	0	1	1	1	F	1	1	1	1

图 3-1 八进制、十六进制对应二进制转换图

(1) 将二进制数转换为八、十六进制数。根据以上原则，二进制数转换八、十六进制数以小数点为中心分别向左右两边分组，八进制三位一组，十六进制四位一组，两端位数不足一组时以零补齐，简称“多化一”。

【例 3.5】 把二进制数 $(1011101.00101)_2$ 转换为八进制数和十六进制数。

$$\begin{array}{ccccccc} \underline{001} & \underline{011} & \underline{101} & \underline{001} & \underline{010} & & \\ 1 & 3 & 5 & . & 1 & 2 & \end{array} \quad (135.12)_8$$

$$\begin{array}{ccccccc} \underline{0101} & \underline{1101} & \underline{0010} & \underline{1000} & & & \\ 5 & D & . & 2 & 8 & & \end{array} \quad (5D.28)_{16}$$

(2) 将八、十六进制数转换为二进制数。为上述运算的逆运算，即“一化多”，八进制数一位化为三位，十六进制数一位化为四位，若两端有多余零省略即可。

【例 3.6】 将例 3.5 的八进制、十六进制数转换回二进制数。

$$\begin{array}{ccccccc} (1 & 3 & 5 & . & 1 & 2)_8 & = (1011101.00101)_2 \\ & & & & & & 001\ 011\ 101.001\ 010 \end{array}$$

$$(5D.28)_{16} = (1011101.00101)_2$$

$$0101\ 1101\ .\ 0010\ 1000$$

八进制数和十六进制数的转换,一般利用二进制数作为中介进行。

【例 3.7】 将 $(5D.28)_{16}$ 转换为八进制数。

$$(5D.28)_{16} = (1011101.00101)_2$$

$$0101\ 1101\ .\ 0010\ 1000$$

$$(\underline{001}\ \underline{011}\ \underline{101}.\underline{001}\ \underline{010})_2 = (135.12)_8$$

$$1\ 3\ 5\ .\ 1\ 2$$

$$(5D.28)_{16} = (135.12)_8$$

3.1.3 二进制数的运算

二进制数可以进行十进制数一样的加减乘除算数运算,其原则是“逢二进一,借一当二”,另外,二进制数据还可以直接进行逻辑运算。

1. 算数运算

加法运算法则如下:

$$0+0=0\quad 0+1=1\quad 1+0=1\quad 1+1=0(\text{进位})$$

减法运算法则如下:

$$1-0=1\quad 1-1=0\quad 0-0=0\quad 0-1=1(\text{借位})$$

乘法运算法则如下:

$$0\times 0=0\quad 0\times 1=0\quad 1\times 0=0\quad 1\times 1=1$$

乘法和除法的运算方式和十进制数据算数运算的方法完全一致,不再赘叙,以下以例题形式介绍。

【例 3.8】 请计算 8 位二进制数 10101100 和 00111010 的和与差。

$$\begin{array}{r} 10101100 \\ +00111010 \\ \hline 11100110 \end{array}$$

$$10101100+00111010=11100110$$

$$\begin{array}{r} 10101100 \\ -00111010 \\ \hline 01110010 \end{array}$$

$$10101100-00111010=01110010$$

【例 3.9】 请计算 1101 与 1011 的乘积。

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

$$1101\times 1011=10001111$$

【例 3.10】 请计算 100011 除以 101 的结果。

$$\begin{array}{r}
 111 \\
 101 \overline{) 100011} \\
 \underline{101} \\
 111 \\
 \underline{101} \\
 101 \\
 \underline{101} \\
 0
 \end{array}$$

$$100011 \div 101 = 111$$

2. 逻辑运算

逻辑代数是一种二级代数,只有“0”和“1”两个量,用来表示两个对立的逻辑状态。逻辑变量之间的运算称为逻辑运算。二进制数“1”和“0”在逻辑上可以代表“真”与“假”、“是”与“否”、“有”与“无”。它是按位进行的,位与位之间不像加减运算那样有进位或借位的联系。其包括“或”、“与”、“非”、“异或”四种运算。

“或”运算又称逻辑加法。通常用符号“+”或“ $\vee$ ”或“ $\cup$ ”来表示,给定的两个逻辑变量中,只要有 1 其结果便为 1。具体规则如下:

$$0+0=0; \quad 0+1=1; \quad 1+0=1; \quad 1+1=1$$

“与”运算又称逻辑乘法。通常用符号“ $\times$ ”或“ $\wedge$ ”或“ $\cdot$ ”来表示。给定的两个逻辑变量中,只要有 0 其结果便为 0。具体规则如下:

$$0 \times 0=0; \quad 0 \times 1=0; \quad 1 \times 0=0; \quad 1 \times 1=1$$

“非”运算又称逻辑否。通常用符号“-”或“ $\neg$ ”来表示。它表示逻辑变量的相反值:

$$\bar{0}=1; \quad \bar{1}=0$$

“异或”运算又叫半加运算。通常用符号“ $\oplus$ ”表示,两个逻辑变量相异,输出才为 1。其运算规则为:

$$0 \oplus 0=0; \quad 0 \oplus 1=1; \quad 1 \oplus 0=1; \quad 1 \oplus 1=0$$

注意: 本节所讲的数制转换、算术运算、逻辑运算等均可在计算机操作系统所附带的计算器中实现。选择“开始”|“附件”|“计算器”命令可以打开“计算器”窗口。二进制相关运算可选择“查看”|“科学型”命令实现,如图 3-2 所示。



图 3-2 科学型计算器

3.2 计算机中的数据表示

二进制形式适用于对各类数据的编码,各类数据所传递的信息也不尽相同,在处理计算机内部的数据之前,有必要先了解数据与信息概念。信息是事物发出的消息、指令、数据、符号等所包含的内容。信息既是对各种事物变化和特征的反映,又是事物之间相互作用和联系的表现。信息必须依附于载体,具有可传递性,共享性和可处理性等特点。数据是信息的载体,是用以载荷信息的物理符号。数字、字符、声音、图像等都是不同形式的数据,均需以适当的方法转换为计算机可识别的二进制码输入处理;输出方面,逆向转换为人们方便识别的多样化数据输出。

3.2.1 数据单位

位(bit,b)是计算机中最小的数据单位,对位的操作是计算机中最直接最基本的操作。每位的状态只能是“0”或“1”,两位二进制数通过排列组合,可以表示 00、01、10、11 这 4 种状态(可参照理解八进制数,是 3 位二进制数,共 8 种状态,分别表示 0~7)。

字节(Byte,B)定义 8 位二进制数为一个字节。字节是计算机中用来表示存储空间大小的最基本容量单位,除字节,还有千字节(KB)、兆字节(MB)和吉字节(GB)等表示。其换算关系如下:

$$1\text{B}=8\text{b}$$

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

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

字(Word)由若干字节构成,一般为字节的整倍数,如 16 位双字,32 为四字,64 位八字等。它是计算机数据处理和运算的单位。字长是计算机性能的重要标志,一次性处理数据的字长越长,计算机性能越高。

3.2.2 数值数据

1. 机器数、真值和原码、反码、补码的表示

由于计算机只能直接识别和处理用“0”或“1”两种状态表示的二进制形式的数据,所以在计算机中无法按人们日常的书写习惯用正负号加绝对值来表示数值,而与数字一样,需要用二进制代码“1”或“0”在左端最高位上来表示正、负号。这种采用二进制表示形式的连同数符一起代码化了的数据,在计算机中统称为机器数或机器码。而与机器数对应的用正、负符号加绝对值来表示的实际数值称为真值。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部分,且均用二进制代码表示。

【例 3.11】 以一个字节表示 +11 和 -11 的真值和机器数。

+11 的真值为 +0001011,机器数为 00001011(数符“+”以左端最高位的 0 表示)

-11 的真值为 -0001011,机器数为 10001011(数符“-”以左端最高位的 1 表示)

注意: 机器数表示范围受字长和数据类型限制。例如,字长为 8 位的一个整数,则最

大正数为 01111111,最高位为符号位,即表示十进制的 127。超出范围,就要“溢出”。

【例 3.12】 以机器数形式计算十进制 $(-10)+4$ 的结果,假设以一个字节表示。

$$\begin{array}{rcl} 10001010 & \cdots & -10 \text{ 的机器数} \\ +00000100 & \cdots & 4 \text{ 的机器数} \\ \hline 10001110 & \cdots & \text{运算结果为 } -13 \end{array}$$

可见,用机器数并不能很好实现带符号的二进制数的算数运算,以下引入原码、反码、补码的概念来弥补机器数无法进行算数运算的不足。

原码就是机器数。反码的正数形式与原码相同,负数符号相同,余位取逻辑反值。补码是在反码基础上加 1。

【例 3.13】 以一字节表示 ± 127 和 ± 0 的原码、反码、补码形式。

$$\begin{array}{ll} [+127]_{\text{原}} = 01111111 & [-127]_{\text{原}} = 11111111 \\ [+0]_{\text{原}} = 00000000 & [-0]_{\text{原}} = 10000000 \\ [+127]_{\text{反}} = 01111111 & [-127]_{\text{反}} = 10000000 \\ [+0]_{\text{反}} = 00000000 & [-0]_{\text{反}} = 11111111 \\ [+127]_{\text{补}} = 01111111 & [-127]_{\text{补}} = 10000001 \\ [+0]_{\text{补}} = 00000000 & [-0]_{\text{补}} = 00000000 \end{array}$$

注意: $[-0]_{\text{补}} = \boxed{1}00000000$,最高位的进位 1 已经超出 8 位,被丢弃。多出来的这个补码 100000000 可以扩展补码的表示范围,将最小值 -127 扩展至 -128 ,此处的 1 即表示符号位又表示数值位。

可见,只有补码的 ± 0 值相同,所以,利用补码可以方便地实现正、负数的加法运算,规则简单,在数的有效范围内,符号位如同数符一样参与运算,也允许溢出的最高位进位或被丢弃,所以补码使用广泛。

【例 3.14】 利用补码再计算例 3.10 中 $(-10)+4$ 的结果。

$$\begin{array}{ll} [-10]_{\text{原}} = 10001010 & \\ [-10]_{\text{反}} = 11110101 & \\ [-10]_{\text{补}} = \boxed{11110110} & \\ [+4]_{\text{原}} = [+4]_{\text{反}} = [+4]_{\text{补}} = \boxed{00000100} & \\ \begin{array}{rcl} 11110110 & \text{此时结果是补码形式,} & [11110110]_{\text{反}} = 10000101 \\ +00000100 & \text{需再进行一次求补} & [11110110]_{\text{补}} = 10000110 \\ \hline 11111010 & \text{运算转换成原码} & (10000110)_2 = (-6)_{10} \end{array} & \end{array}$$

可见,利用补码可以顺利地得出正确结果,所以在计算机内部,大多算术运算都是以补码形式进行的。

2. 定点数与浮点数

(1) 定点数。定点数,顾名思义,小数点固定的数。此类数在十进制中有两类:整数,小数点固定在最右端;纯小数,小数点固定在首位 0 后。所以,对于定点数的表示,默认小数点的预定位即可,整数为最右端,小数为最左端(整数部分 0 省略无须表示)。如用

一个字节处理 $(+120)_{10}$ 可表示为 01111000; -0.125 可表示为 10010000。

(2) 浮点数。对于普通格式的小数,其小数点位置不再固定,但可以借助科学(指数)基数法的格式将其转换成定点小数与定点整数的组合。如 -1101.1011 可表示为:

$$-1101.1011 = -1.1011011 \times 2^3$$

注意: 指数部分 3 本应用二进制表示,但习惯上写成十进制。

其中 -1.1011011 叫尾数,是定点小数形式,在浮点数规格化规定中,按 IEEE 754 规定,尾数部分必须化成“1.”形式,但其并不在数据中存储,以节省存储空间。

3 是阶码,是定点整数形式,规格化中规定指数需加上 127 存储,目的是消除负指数,可以省略指数部分的阶码符号,节省存储空间。从而浮点数可表示成如下格式(以 4 个字节 32 位的单精度数为例):

1 位	8 位	23 位
数符	阶码	尾数

【例 3.15】 $(-12.75)_{10}$ 作为单精度浮点数在计算机的表示。

$$(-12.75)_{10} = (-1100.11)_2 = -1.10011 \times 2^3 \text{ B}$$

阶码转换: $3 + 127 = 130 = 10000010 \text{ B}$

则 $(-12.75)_{10}$ 的存储格式为

1	10000010	100110000000000000000000
---	----------	--------------------------

双精度为 8 字节(64 位),其中数符、阶码和尾数分别占 1,11 和 52 位,阶码是指数加上 1023 所得。

3.2.3 字符数据

1. 西文字符

所有不可做算术运算的数据均可称为字符,西文字符编码最常用的是 ASCII 编码(American Standard Code for Information Interchange,美国信息交换标准代码)。其用 7 个二进制数表示一个字符,可以表示 2^7 即 128 个字符。一个字节中多余的 1 位,即最高位用 0 表示。ASCII 码表如表 3-1 所示,其中 H 表示高四位,L 表示低四位。

2. 汉字字符

英文由字母组合而成,其字符不超过 128 个,一个字节 2^8 即可满足编码要求。但汉字是象形文字,难以组合,每个汉字都需要一个二进制代码,所以汉字编码方案有 2~4B 不等。汉字输入时要经过输入码、国标码、内码、字形码等一系列的编码和转换。

(1) 输入码。输入码也叫外码,是人们接触最多的一种编码。属于音码类的智能 ABC,搜狗输入法和属于形码类的五笔字型等都是常用的输入码。编码主要考虑以下规则:编码短,重码少,好学好记。

表 3-1 标准 ASCII 码表

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

(2) 国标码。1981 年我国制定的《GB 2312—1980 中华人民共和国国家标准信息交换汉字编码 基本集》简称国标码。在这个集中,收进汉字 6763 个图形符号 682 个。其中一级汉字 3755 个,二级汉字 3008 个。一级汉字为常用字,按拼音顺序排列,二级汉字为次常用字,按部首排列。所有国标汉字组成 94×94 的矩阵,即 94 个区和 94 个位,区号和位号共同构成区位码。如 45 区 85 位的汉字“王”,区位码为 4585,十六进制表示为 2D55H。区位码加 2020H 即为国标码,所以“王”的国标码为 4D75H。

(3) 机内码。国标码占两个字节,每个字节的最高位为 0,而一个字节的 ASCII 码的最高位也是 0,如此计算机将无法区分是一个汉字的编码还是两个西文字符的编码。为此将国标码加 8080H,将每个字节的最高位由 0 化为 1 以区分中西文字符。所以,机内码 = 国标码 + 8080H。

(4) 字形码。字形码又称汉字字模,用于显示或输出。有点阵和矢量两种表示方式。

生活中常见的 LED 广告显示屏是最直观的点阵式输出方式。常用的汉字点阵有 16×16 , 24×24 , 36×36 , 48×48 等,点阵数越多,表示的字形信息越完整,显示的汉字越

精细,图 3-3 所示为 24×24 点阵表示的“跑”字,完全依照字形设计。

矢量字体中每一个字形是通过数学曲线来描述的,它包含了字形边界上的关键点,连线的导数信息等,字体的渲染引擎通过读取这些数学矢量,然后进行一定的数学运算来进行渲染。这类字体的优点是字体实际尺寸可以任意缩放而不变形、变色。Windows 中使用的 TrueType 基数就是汉字矢量表示方式。

3.2.4 多媒体数据

图形、图像、音频、视频等多媒体信息,也需转化成 0、1 代码在计算机内进行表示,这些都需要对各种多媒体信息进行不同的编码。

3.2.5 程序设计概述

1. 程序

程序是完成一定功能的指令序列。程序的功能一般指其处理数据的能力,包括对数据的描述和对操作的描述两方面的内容。对数据的描述是在程序中要指定处理数据的类型和组织形式,即数据结构;对操作的描述也叫算法,数据是操作的对象,操作的目的是对数据进行加工处理以得到期望的结果。著名的计算机科学家 Nikiklaus Wirth 提出的“程序=数据结构+算法”体现了数据结构和算法的重要性。

(1) 数据结构。数据是对客观事物的名称、数量、特征、性质的描述形式,是计算机所能处理的一切符号的总称。一般对于孤立的数据并没有意义,研究主要着眼在众多数据元素组成的数据集合。研究集合中元素之间存在怎样的联系,需要对数据和数据集合进行哪些运算,如何提高运算效率等这就引出了数据结构。

数据结构包括一批数据,是数据的集合。集合中每一数据个体称为数据元素,是基本单位。一个元素又叫数据结点,简称结点。数据结构是带有结构特性的数据元素的结合,它研究的是数据的逻辑结构和存储结构以及它们之间的相互关系,并对这种结构定义相适应的运算,设计出相应的算法。

(2) 算法。算法是程序的核心,是对特定问题求解步骤的详细描述,需要满足有穷性、确定性、可行性、输入性、输出性五大原则。算法有很多种表示方法,常用的有自然语言、流程图、N-S 图、伪代码等。

自然语言是用人们日常使用的语言来描述算法通俗易懂,但因易产生歧义,语句繁琐、冗长等不常使用。

流程图是用图 3-4 所示的一些图框、线条以及文字说明来描述算法。流程图表示的算法形象、直观,便于交流,因此被广泛使用。

N-S 图是一种简化的流程图,去掉了流程图中的流程线,全部算法写在一个矩形框内。N-S 图 3 种基本结构——顺序结构、选择结构、循环结构的符号如图 3-5 所示。N-S

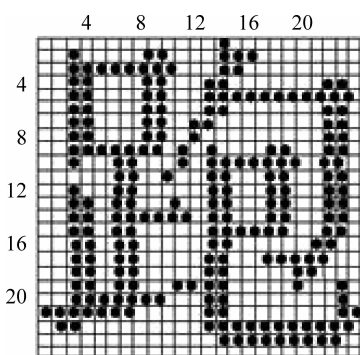


图 3-3 点阵字形码

图表示算法直观、形象,且比流程图紧凑易画,经常应用在实际中。

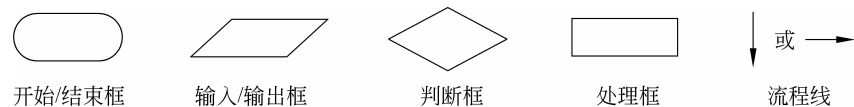


图 3-4 流程图的常用符号

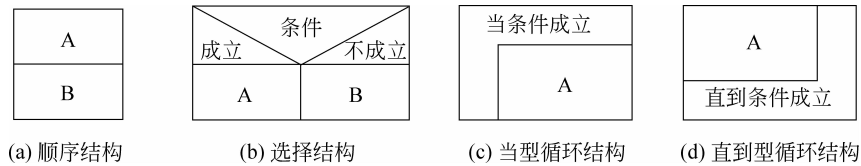


图 3-5 N-S 图的三种基本结构

为了设计算法时方便,也常使用一种称为伪代码的工具。所谓“伪代码”就是用介于自然语言和计算机语言之间的文字和符号来描述算法。伪意味着假,因此用伪代码写的算法是一种假代码——不能被计算机所理解,但便于转换成某种语言编写的计算机程序。用伪代码写算法并无固定的、严格的语法规则,只要意思表达清楚,书写格式清晰易读即可。

2. 程序设计

程序设计是设计、编制、调试程序的方法和过程。其基本过程一般由分析问题,设计算法,选择程序语言,编写程序调试直至正确几个阶段组成。当需要编写一个计算机程序时,首先要搞清楚为什么要编写这个程序,这个程序要干什么用,需要完成什么功能。这就是所谓的分析问题,是编程的第一步。问题分析清楚之后,就要考虑如何用计算机解决这些问题,也就是要设计算法。算法设计完成后,编写程序之前需要选择程序设计语言。一般来说,不同的语言可以完成同样的任务,因此选择什么语言都行。但是,在实际工作中,人们往往会综合各种要求选择最适合的语言。编写好的程序一般称为源程序,不能直接执行,必须要经过编译生成计算机可识别的二进制文件才能执行。最后通过调试修改无误后方完成整个设计过程。

结构化程序设计的概念最早由荷兰科学家 E. W. Dijkstra 提出的。早在 1966 年他就指出,任何程序都基于顺序、选择、循环 3 种基本的控制结构,如图 3-6 所示。并且程序具有模块化特征,每个程序模块具有唯一的入口和出口。这为结构化程序设计的技术奠定了理论基础。

结构化程序设计主要包括两个方面。

(1) 在软件设计和实现过程中,提倡采用自顶向下、逐步细化的模块化程序设计原则;

(2) 在代码编写时,强调采用单入口单出口的 3 种基本控制结构,避免使用 GOTO 语句。

采用结构化程序设计方法设计的程序结构简单清晰,可读性强,模块化强,描述方式

符合人们解决复杂问题的普遍规律,在软件重用性、软件的可维护性方面有所进步,可以显著提高软件开发的效率。因此,结构化程序设计方法在应用软件的开发中发挥了重要的作用。

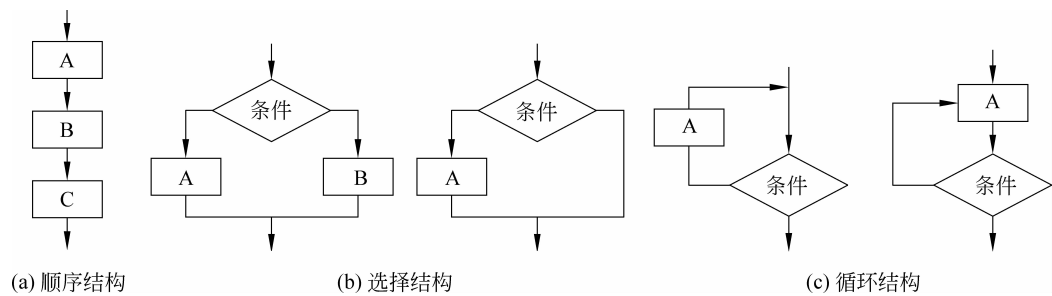


图 3-6 3 种基本的控制结构

阅读材料 3

周易八卦和二进制

众所周知二进制数学是 16 世纪初德国科学家莱布尼兹发明的。但二进制真正的起源来自中国三千年前的著作《周易》。其对二进制数的使用和二、十进制数的转换编码的运用上更简单、更先进、更科学。

图 3-7 所示为《周易》中的“先天八卦次序”，它由“两仪”、“四象”、“八卦”3 行黑白矩形组组成。“两仪”中有两个矩形，“四象”中有 4 个矩形，“八卦”中有 8 个矩形。矩形上面是八卦的卦符。

那么“先天八卦次序”又表示了什么呢，八卦的卦符又是根据什么画出来的？在“先天八卦次序”中，白矩形表示阳，可以用阳爻表示，黑矩形表示阴，可以用阴爻表示。如果沿八卦各卦的垂直方向看“两仪”、“四象”、“八卦”中矩形的颜色，用阳爻表示白矩形，阴爻表示黑矩形，就可以画出八卦各卦的卦符。

由此可见，八卦的卦符表示了八卦各卦的生成过程，而不是江湖术士和易学专家所说的“卦符是古人用蓍草算卦得出来的”。

根据二进制数的规定：用“1”表示有，用“0”表示无。人们可以得出八卦各卦阳爻和阴爻的二进制数。下面人们写出八卦各卦阳爻的二进制数(即有阳爻为“1”，无阳爻为“0”)，八卦各卦阴爻的二进制数(即有阴爻为“1”，无阴爻为“0”)：

八卦各卦阳爻的二进制数

- 坤：黑黑黑，卦符阴阴阴，二进制数为 000
- 艮：黑黑白，卦符阴阴阳，二进制数为 001
- 坎：黑白黑，卦符阴阳阴，二进制数为 010
- 巽：黑白白，卦符阴阳阳，二进制数为 011



图 3-7

震：白黑黑，卦符阳阴阴，二进制数为 100
离：白黑白，卦符阳阴阳，二进制数为 101
兑：白白黑，卦符阳阳阴，二进制数为 110
乾：白白白，卦符阳阳阳，二进制数为 111

八卦各卦阴爻的二进制数

坤：黑黑黑，卦符阴阴阴，二进制数为 111
艮：黑黑白，卦符阴阴阳，二进制数为 110
坎：黑白黑，卦符阴阳阴，二进制数为 101
巽：黑白白，卦符阴阳阳，二进制数为 100
震：白黑黑，卦符阳阴阴，二进制数为 011
离：白黑白，卦符阳阴阳，二进制数为 010
兑：白白黑，卦符阳阳阴，二进制数为 001
乾：白白白，卦符阳阳阳，二进制数为 000

可见“先天八卦次序”中，八卦的二进制数排列是有规律的。

在“先天八卦次序”中，“两仪”、“四象”、“八卦”中矩形面积比是 4：2：1 如果定义“八卦”中矩形面积为 1，那么“四象”、“两仪”中的矩形面积分别为 2 和 4。人们可以沿八卦各卦的垂直方向，依次将“两仪”、“四象”、“八卦”中同色矩形的面积相加就可以得出八卦各卦的十进制数。八卦各卦阳爻的二进制数转换成十进制数即为 0~7(例如坤： $[000]_2 = 0 \times 4 + 0 \times 2 + 0 \times 1 = 0$)。

由此可见，用“先天八卦次序”表示的方法将二进制数转换成十进制数与莱布尼兹的方法完全相同。无须再计算每一位的二的幂级的位权值 2^x ，而是直接以位权结果代替，从而更简单、更先进、更科学。应当指出的是，“先天八卦次序”不仅表示了二进制数的符号及其转换编码，还表明二进制数不仅是数，而且还是数理逻辑符号。

《易之道》中用现代自然科学的方法证明了，《周易》是中国古代贤哲依据生命现象创造的一个用严密的数理逻辑语言表达宇宙基本结构和普遍规律的科学体系。或者说，《周易》是一个关于生命、关于宇宙的二进制数学模型。而“先天八卦次序”是生物体中 DNA 双螺旋结构自我复制的二进制数学模型。这是现代生物学家和数学家加起来也做不到的，甚至连想都想不到的。

现代的一些世界著名科学家和哲学家指出，生物学和生命现象是告诉人们如何解决疑难问题的百科全书。从这个意义上讲，《周易》不仅是中国传统文化的源泉，也是现代世界文化的源泉。

下面将“先天八卦次序”表示的二进制数及其二、十进制数编码用图 3-8 表示。图 3-8 左上方是作的“先天八卦次序”的十进制数的图像。既然“先天八卦次序”中存在二进制数及二、十进制数的编码，那么《周易》中就肯定存在与二进制图像对应的十进制图像。“太极图”就是“先天六十四卦方位”的十进制图像。下面论证这个问题。

首先，仿照图 3-8 画出图 3-9“先天六十四卦次序”的十进制数图像。图 3-9 下方是“先天六十四卦次序”，上方是其十进制数图像。“先天六十四卦方位”及其六十四卦的排列规律，如图 3-10 和图 3-11 所示。

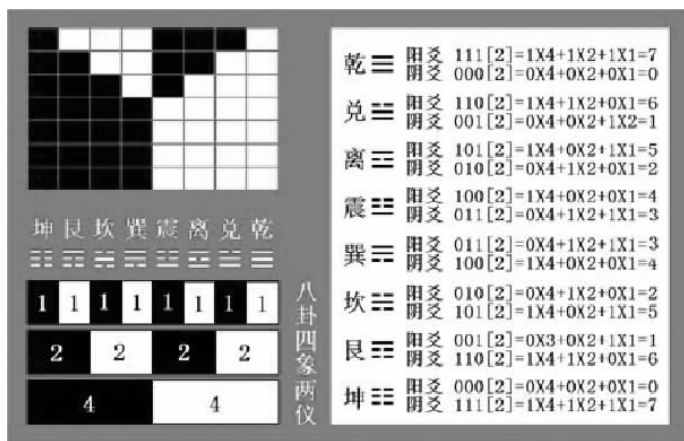


图 3-8

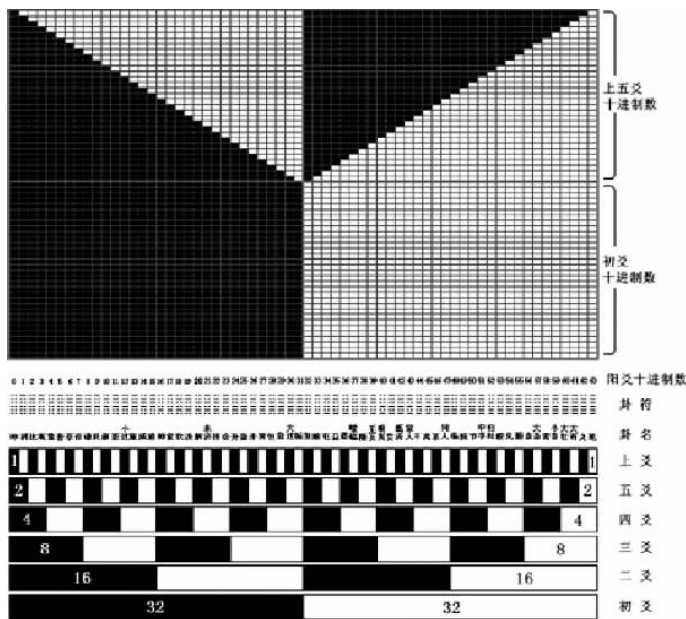


图 3-9

根据图 3-10～图 3-11 可画出“先天六十四卦方位”的十进制数图像,如图 3-12 所示。

图 3-12 左半圆由一个小半圆和一个扇形构成,它们分别用十进制数表示了“先天六十四卦方位”初爻是阳爻的 32 卦的上五爻和初爻;右半圆由一个小半圆和一个扇形构成,它们分别用十进制数表示了“先天六十四卦方位”初爻是阴爻的 32 卦的上五爻和初爻。初爻的十进制数都是 64,所以可以用一个小白圆替代左边的扇形,并将其移入左边的小半圆中的黑色区中。同样,可以用一个小黑圆替代右边的扇形,并将其移入右边的小半圆中的白色区中。这样图 3-12 就变成图 3-13。显然图 3-13 就是“太极图”。这样就证明了“太极图”就是“先天六十四卦方位”的十进制数图像。所以说,在《周易》中存在一个完整

的二进制数学体系。

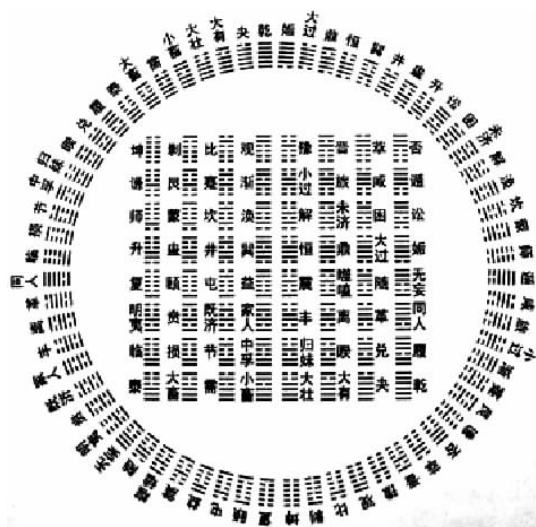


图 3-10

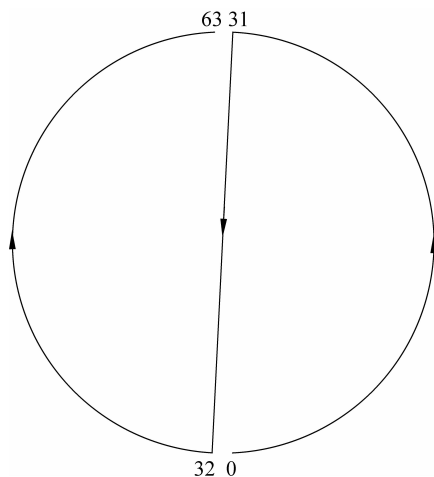


图 3-11

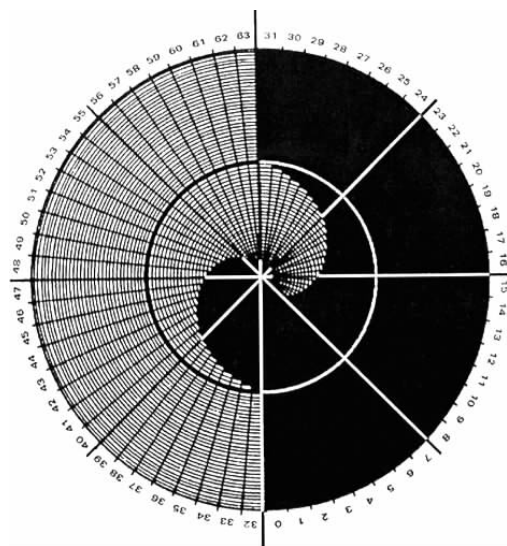


图 3-12

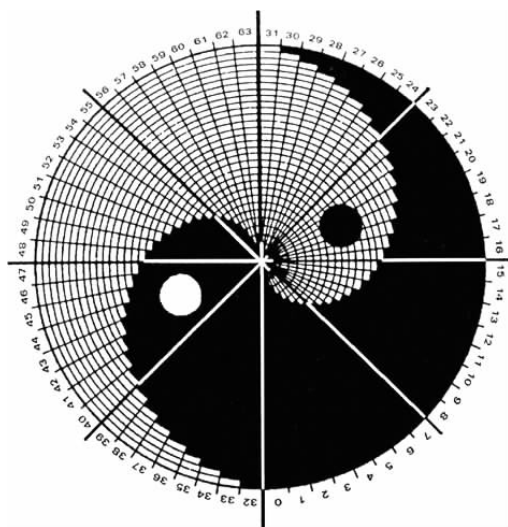


图 3-13

从目前已知的西方历史文献中,可以得知中国的易经图于17世纪二三十年代就已被世人称为二进制广为流传于欧洲。莱布尼兹(Gottfried Wilhelm von Leibniz)是德国著名的数学家和哲学家,他对法国人帕斯卡设计的第一台机械式数字计算机——加法机——很感兴趣,于是也开始了计算机的研究。1666—1667年,莱布尼兹在纽伦堡学习时已开始接触中国古典哲学中的易经图,如卫匡国在《中国上古史》中译著的伏羲六十四卦方位图、柏应理在《中国哲圣孔子》所译著的太极八卦次序图、八卦方位图和文王六十四卦图。特别是他所看到的与其有过密切交往的斯比塞尔(Gottfried Spizel),于1660

年编著出版的 *Dere litteraria Sinensium commentarius* (中文译为《中国文史评析》、《中国文学》、《论中国的宗教》等) 一书, 其中对此已有较详细完整的介绍。

此后, 大约是在 1672—1676 年, 莱布尼兹开始了“0”与“1”的二进制思考。1679 年 3 月 15 日, 他撰写了题为《二进算术》的论文, 对二进制进行了充分的讨论, 并建立了二进制的表示及运算。1701 年, 莱布尼兹将关于二进制的论文提交给法国科学院, 但要求暂不发表。1703 年, 他将修改后的论文再次送给法国科学院, 并要求公开发表。这是西方第一篇关于二进位制的文章, 是莱布尼兹在《皇家科学院纪录》上发表的, 标题为《二进制算术的解说》, 副标题为“它只用 0 和 1, 并论述其用途以及伏羲氏所使用的古代中国数字的意义”。自此, 二进制开始公之于众。1716 年, 他又发表了《论中国的哲学》一文, 专门讨论八卦与二进制, 指出二进制与八卦有其共同之处。

习题 3

一、选择题

1. 十进制数 39.625 的二进制数为()。
A. 100010.011 B. 110010.0101 C. 110110.101 D. 100111.101
2. 二进制数 11101.010 的十进制数为()。
A. 31.25 B. 29.75 C. 29.5 D. 29.25
3. 一个字节包含()位。
A. 8 B. 16 C. 32 D. 2
4. 二进制数 00111111 和 10101010 的逻辑与运算的结果为()。
A. 10111111 B. 00101010 C. 10010101 D. 00111111
5. 二进制数 00111111 和 10101010 的逻辑或运算的结果为()。
A. 10111111 B. 00101010 C. 10010101 D. 00111111
6. 二进制数 00111111 和 10101010 的逻辑异或运算的结果为()。
A. 10111111 B. 00101010 C. 10010101 D. 00111111
7. 二进制数 10110001 的逻辑非运算的结果为()。
A. 10110001 B. 01101010 C. 01001110 D. 00111111
8. 八进制数 245.27 转换为二进制数的结果为()。
A. 10100101.010111 B. 11101010.01011
C. 10001001.101101 D. 10100111.01001
9. 二进制数 10100111 转换为八进制的结果为()。
A. 237 B. 247 C. 165 D. 327
10. 设汉字点阵为 32×32 , 那么 100 个汉字的字形形状信息所占用的字节数是()。
A. 12800B B. 12800b C. $32 \times 32K$ D. 128K
11. 流程图符号中, 菱形代表的含义是()。
A. 开始/结束 B. 输入输出 C. 处理 D. 判断

12. ()不属于结构化程序设计的基本结构。
A. 顺序结构 B. 选择结构 C. 循环结构 D. 跳转结构
13. 用一字节表示的-42的反码为()。
A. 10101010 B. 01010101 C. 01010110 D. 11010101
14. 用一字节表示的-42的补码为()。
A. 11010110 B. 01010101 C. 01010110 D. 01010111
15. 大写字母 A 的 ASCII 为()。
A. 65 B. 97 C. 48 D. 32

二、简答题

1. 简述为何在计算机内使用二进制编码。
2. 浮点数在计算机中是如何表示的?
3. 简述机内码和国标码的转换关系。
4. 如果一个有符号数占有 n 位,那么它的最大值是多少?
5. 计算机对英文字符进行编码用了几个字节,最高位是什么? 对汉字进行编码采用了几个字节,最高位是什么?
6. 简述算法的几种表示方法。

第4章

操作系统基础

操作系统是计算机最基本、最核心的系统软件,是计算机中不可或缺的必要组成部分,是整个计算机系统的控制和管理中心。系统的全部硬件资源、软件资源及数据资源均由操作系统统一调配,它还负责控制程序运行,改善人机交互界面,为其他应用软件提供支持,使计算机系统所有资源最大限度地发挥作用,为用户提供方便的、有效的、友善的服务界面。

4.1 操作系统概述

没有安装操作系统的计算机被称为“裸机”,裸机无法进行任何工作,不能从键盘、鼠标接收信息,也不能在显示屏上显示信息,更不能运行各类程序。操作系统是配置在计算机硬件上的第一层软件,是对硬件的首次扩充,它在计算机系统中属于根基性软件,所有的计算机均需配置一种或几种操作系统。汇编语言、编译程序、数据库管理系统等其他系统软件及大量的应用软件也均需依赖操作系统的支持、服务才能顺利运行。计算机的软硬件层次的关系如图 4-1 所示。

操作系统在整个计算机系统中具有极其重要的特殊地位,它不仅是硬件与其他软件系统的接口,也是用户和计算机进行“交流”的界面。操作系统的作用是调度、分配和管理所有的硬件设备和软件系统统一协调地运行,以满足用户实际操作的需求。操作系统的作用主要体现在以下两方面。

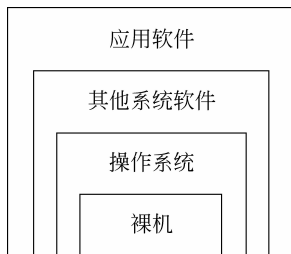


图 4-1 操作系统软硬件层级关系图

(1) 有效管理计算机资源。操作系统要合理地组织计算机的工作流程,确保软件和硬件之间、用户和计算机之间、系统软件和应用软件之间信息传输和处理流程的准确畅通;有效地管理和分配计算机系统的硬件和软件资源,使得有限的系统资源能够发挥更大的作用。

(2) 方便用户有效地使用计算机,使计算机成为人们工作、生活、休闲的工具。操作系统通过内部极其复杂的综合处理,为用户提供友好、敏捷的操作界面,以使用户无须了解计算机硬件或系统软件的有关细节就能方便地使用计算机。

4.1.1 操作系统的发展

操作系统在其发展的历史中,经历了发生、发展和成熟的过程。它从无到有,从小到大,从简单到复杂,从单一到多样,形成了计算机科学的一个重要分支。操作系统的形成与发展经历的几个阶段如表 4-1 所示。

表 4-1 操作系统发展阶段

手工操作阶段 (1946—1955 年)	早期批处理阶段 (1955—1965 年)	多道程序系统阶段 (1965—1980 年)	现代操作系统阶段 (1980 至今)
计算机元器件为电子管器件。无操作系统,人们通过各种操作按钮来控制计算机	计算机元器件为晶体管。有了计算机语言和相应程序,计算机可以集中处理一批用户作业,由批处理程序自动转换、依次处理运行	计算机元器件为中、小规模集成电路。引入多道程序设计技术,形成强大的监管程序,进而发展为传统操作系统,其基础为多道程序分时系统和多处理器操作系统	计算机元器件为大、超大规模集成电路。操作系统面向用户,视窗操作和视窗界面迅速发展,具有强大的网络互连和通信功能,能够高效处理多媒体信息

操作系统种类很多,DOS 曾在 20 世纪 80 年代占主流地位,目前主要有 Windows、UNIX、Linux、和 Mac OS,如表 4-2 所示。

表 4-2 常用操作系统

操作系统	主 设 计 人	出现时间	最 新 版 本	系 统 特 点
DOS	Tim Paterson	1981 年	终极版为 DOS 7.0(1995 年)	命令行界面
Windows	Microsoft 公司	1985 年	Windows XP、Windows 7、Windows 8	图形用户界面
UNIX	贝尔实验室	1969 年	版本众多	分时系统
Linux	Linus Torvalds	1991 年	版本众多	免费、源代码开放
Mac OS	比尔·阿特金森 杰夫·拉斯金 安迪·赫茨菲德尔	1984 年	Mac OS X Leopard	运行在 Macintosh 计算机上

UNIX 系统是 1969 年在贝尔实验室诞生,最初是在中、小型计算机上运用。最早移植到 80286 微型计算机上的 UNIX 系统,称为 Xenix。Xenix 系统的特点是短小精悍,系统开销小,运行速度快。UNIX 为用户提供了一个分时的系统以控制计算机的活动和资源,并且提供一个交互,灵活的操作界面。UNIX 被设计成能够同时运行多进程,支持用户之间共享数据。

1976 年,美国 Digital Research 公司研制了 8 位的 CP/M 操作系统,该系统允许用户通过键盘对系统进行控制和管理,能实现文件或其他设备文件的自动存取。之后出现了一些类似的磁盘操作系统,如 C-DOS、M-DOS、TRS-DOS、S-DOS 等。1981 年,IBM 成功的开发了个人计算机,所配操作系统即为微软公司的 MS-DOS。从 1981 年开始,DOS 经

历了 7 次版本升级,从 1.0 版到 7.0 版,但其单用户、单任务、字符界面和 16 位的大格局并没有变化,因此,它对于基本内存管理也局限在 1MB 范围内。

1981 年,Xerox(施乐)公司推出了世界上第一个商用图形用户界面(GUI)的操作系统,用于 Star 8010 工作站。但由于种种原因,技术上的先进性并没有带来商业上的成功。当时,Apple Computer(苹果)公司创始人之一 Steve Jobs,在参观了施乐公司之后,认识到了图形用户接口的重要性以及广阔的市场前景,开始着手进行自己的 GUI 系统的研究与开发,并于 1983 年成功研制了 GUI 系统——Apple Lisa。随后不久又推出第二个系统 Apple Macintosh,这是一个成功的商用 GUI 系统。之后的 Mac OS 操作系统也仅能使用于 Macintosh 计算机。在当时的 PC 还只是 DOS 枯燥的字符界面的时候,Mac OS 率先采用了一些至今仍为人称道的技术,比如 GUI 图形用户界面、多媒体应用、鼠标等,Macintosh 计算机在出版、印刷、影视制作和教育等领域有着广泛的应用,Microsoft Windows 至今在很多方面还有 Mac OS 的影子。但由于只开发了自己微型计算机上的 GUI 系统,不能与已占领市场大部分份额的 Intel x86 微处理器芯片计算机兼容,这样,给 Microsoft 公司开发 Windows 提供了良好的机会。

Microsoft 公司于 1983 年宣布开始研究基于 Intel x86 微处理器芯片 Windows 操作系统。Windows 1. x 是其在 1985 年 11 月发布的第一代窗口式多任务系统,它使 PC 开始进入了图形用户界面时代。1987 年底,Microsoft 公司又推出了 MS-Windows 2. x,它具有窗口重叠功能,窗口大小也可调整,从而提高了整台计算机的性能。1990 年,Microsoft 公司推出 Windows 3. 0,具有强大的内存管理功能,并提供了相当多的 Windows 应用软件,因此成为 386、486 系列微型计算机新的操作系统版本。随后,发布 Windows 3. 1 版本,且推出了相应的中文版。1995 年,Microsoft 公司推出了 Windows 95。在此之前的 Windows 都是由 DOS 引导的,也就是说它们还不是一个完全独立的系统,而 Windows 95 是一个独立的系统,并在很多方面作了进一步的改进,还集成了网络功能和即插即用功能,是一个全新的 32 位操作系统。1998 年推出的 Windows 98,最大特点就是把微软公司的浏览器技术整合到了操作系统里,使得访问 Internet 资源非常方便。近几年来,Microsoft 公司由陆续推出了 Windows ME、Windows 2000、Windows XP、Windows 2003、Windows Vista、Windows 7 和 Windows 8。

Linux 是开放源代码的操作系统,其本身是一个功能可与 UNIX 和 Windows 相媲美的操作系统,具有完备的网络功能。由于源代码开放,激起了全球计算机爱好者的热情,许多人下载该源程序并按自己的意愿完善某一方面的功能,再发回到 Internet 上,Linux 也因此被雕琢成为一个稳定的、有发展前景的操作系统。它的用法与 UNIX 非常相似,因此许多用户不再购买昂贵的 UNIX,转而投入 Linux 等免费系统的怀抱。

4. 1. 2 操作系统的分类

对于操作系统可以有多种分类方式。

(1) 按机型分:大型计算机操作系统、中型计算机操作系统、小型计算机操作系统和微型计算机操作系统。

(2) 按用户数目分:单用户操作系统和多用户操作系统。