

计算机应用基础

主 编 宋荷庆
副主编 朱夏君



浙江工商大学出版社
ZHEJIANG GONGSHANG UNIVERSITY PRESS

计算机应用基础

主 编：宋荷庆

副主编：朱夏君



浙江工商大学出版社
ZHEJIANG GONGSHANG UNIVERSITY PRESS

图书在版编目(CIP)数据

计算机应用基础 / 宋荷庆主编. — 杭州: 浙江工商大学出版社, 2012. 6

ISBN 978-7-81140-581-1

I. ①计… II. ①宋… III. ①电子计算机—教材
IV. ①TP3

中国版本图书馆 CIP 数据核字(2012)第 193630 号

计算机应用基础

主编 宋荷庆 副主编 朱夏君

责任编辑 孙一凡 赵 丹

责任校对 周敏燕

封面设计 包建辉

责任印制 汪 俊

出版发行 浙江工商大学出版社

(杭州市教工路 198 号 邮政编码 310012)

(E-mail: zjgsupress@163.com)

(网址: <http://www.zjgsupress.com>)

电话: 0571-88904980, 88831806(传真)

排 版 杭州朝曦图文设计有限公司

印 刷 杭州杭新印务有限公司

开 本 787mm×1092mm 1/16

印 张 13.25

字 数 323 千

版 印 次 2012 年 6 月第 1 版 2012 年 6 月第 1 次印刷

书 号 ISBN 978-7-81140-581-1

定 价 29.00 元

版权所有 翻印必究 印装差错 负责调换

浙江工商大学出版社营销部邮购电话 0571-88804227

前 言

“计算机应用基础”是高等学校非计算机专业学生的公共基础课,学习计算机应用的基础知识、掌握计算机应用的基本技能是高等学校培养新型人才的一个重要环节。随着计算机应用的普及,对学生掌握计算机应用技术的要求也逐渐提高。为了适应普通高校计算机基础教学的需要,根据“浙江省高校非计算机专业学生计算机基础知识和应用能力等级考试大纲”的要求,结合浙江省计算机等级考试的实际情况编写了本教材。

本教材充分考虑了当前计算机技术的发展,学生应用计算机水平的现状和非计算机专业的学生对计算机知识和应用能力的要求,合理安排了理论与应用、深度与广度方面的内容,使之能满足现阶段对学生计算机知识的要求。

本教材以 Windows XP 操作系统与 Office 2003 为重点内容。作为学习计算机应用基础的教材,不讲解过多的理论知识,不追求面面俱到。本书根据教学大纲的要求,按照非计算机专业学生的特点选择知识内容,内容简洁,结构清晰,叙述准确。在写法上遵照循序渐进的学习过程,突出简便易学的操作步骤,便于学生快速领会和掌握。通过本教材的学习,可以使学生了解计算机应用的基础知识、掌握计算机应用的基本技能,并为学生利用计算机学习其他课程打下良好的基础。

全书主要包括:

第一章主要介绍了计算机的基础知识,包括数据和信息、各种数据在计算机中的表示形式——信息数字化、计算机中数据的存储与编码、计算机系统的结构、计算机系统的软硬件组成等,使学生对计算机组成、结构和原理有一个全面认识。

第二章以目前使用最广泛的 Windows XP 操作系统为主要讲述对象,讲述 Windows XP 的各种基本操作、桌面设置、文件与文件夹的管理、控制面板以及基本附件的使用等。通过本章的学习,使学生掌握 Windows XP 最常用的基本操作。

第三、四、五章,分别讲述 Microsoft Office 2003 中的 3 个常用组件文字处理软件 Word 2003、电子表格软件 Excel 2003 和演示文稿软件 PowerPoint 2003 的基本知识、常用功能、使用方法及操作步骤。

第六章以 Access 为例介绍了关系型数据库的基本概念、Access 数据库的基本对象(表、查询、窗体和报表)使用方法和操作步骤。

第七章介绍了 FrontPage 2003 网页制作工具的基础知识、使用方法和操作步骤。

第八章介绍了计算机网络与 Internet 应用基础,包括网络体系结构的基本概念、计算机网络和 Internet 的基础知识、网络通信协议的概念、TCP/IP 协议、接入 Internet 的方法、WWW 服务及 IE 浏览器、Outlook Express 电子邮件服务及其使用。

本书的主要教学内容符合浙江省计算机等级考试大纲(一级)的基本要求,可作为普通高校的“计算机应用基础”课程的教科书和教学参考用书,也可以作为相应计算机技术培训班的教材或自学计算机应用技术人员的学习指导书。

本教材由浙江工商大学杭州商学院“计算机应用基础”编写委员会编写,宋荷庆主编,朱夏君副主编。《计算机应用基础》各章编写分工情况如下:第一章宋荷庆,第二章宋荷庆、焦华,第三章黄瑶、王珊珊,第四章朱夏君、蔡俊,第五章朱夏君,第六章郑济杰、宋荷庆,第七章郑济杰、王冰,第八章崔莅凯、杨桂君。教材由宋荷庆统稿,凌云审核。在本教材的编写过程中,得到了学院领导的关心和支持,在此表示衷心的感谢!

在教材的定位、选材和编写过程中,编者参阅了许多计算机基础知识书籍和相关论著,从中得到了不少启发,在此谨向这些论著的作者深表谢意!

由于编者水平有限,对于书中存在的不足之处,敬请各位读者批评指正。

编 者

2012 年 6 月

目 录

第一章 计算机基础知识	1
1.1 计算机处理的对象——信息	1
1.2 信息在计算机中的表示——信息数字化	1
1.3 计算机系统.....	11
第二章 Windows 操作系统	16
2.1 操作系统的概念.....	16
2.2 Windows XP 的基本操作	18
2.3 Windows XP 的文件系统及操作	25
2.4 控制面板.....	34
第三章 Word 2003 文档处理	38
3.1 Word 2003 的基本操作.....	38
3.2 文档的编辑与排版.....	45
3.3 表格制作.....	65
3.4 图文排版.....	68
3.5 页面设置与打印.....	74
3.6 案例.....	77
第四章 Excel 2003 数据统计与分析	81
4.1 Excel 2003 简介	81
4.2 Excel 2003 的基本操作.....	81
4.3 格式操作.....	93
4.4 数据管理.....	99
4.5 案例	110
第五章 PowerPoint 2003 演示文稿制作	115
5.1 PowerPoint 2003 简介	115
5.2 PowerPoint 2003 的基本操作	115

5.3 演示文稿的格式化	125
5.4 演示文稿的演示技术	127
5.5 案例	133
第六章 Access 2003 数据库基础	137
6.1 Access 2003 概述	137
6.2 Access 2003 表	140
6.3 查询	147
6.4 窗体和报表	152
第七章 FrontPage 2003 网页设计	159
7.1 FrontPage 2003 简介	159
7.2 网页制作	161
7.3 网站的发布	177
第八章 计算机网络基础及应用	178
8.1 计算机网络概述	178
8.2 Internet 介绍	186
8.3 IE 浏览器	191
8.4 Outlook Express 电子邮件	197
参考文献	203

第一章 计算机基础知识

1.1 计算机处理的对象——信息

1.1.1 信息的概念

信息一词对我们来说并不陌生,在日常生活和工作中,每个人随时随地都在与信息打交道,都在不断地接收信息、加工信息、利用信息。计算机就是一种高效的信息处理工具,所以有时候也把计算机叫作信息处理工具。那么到底什么是信息呢?

(1)信息的定义。一方面,信息是一个内容丰富、运用普遍、含义又相当模糊的概念,要对信息一词作出确切的定义是很困难的。另一方面,信息的概念广泛地渗透到各门学科之中,人们可以根据各学科自身的特点为信息作出各种各样的定义。“信息是经过解释和处理后对接受者进行决策时有意义的的数据”。这是人们普遍接受的一个定义,本书中我们引用这个定义。

(2)信息和数据。在日常生活中,人们容易把信息与数据混同。实际上,从信息定义中也可以看到,信息和数据是既有联系又有区别的。数据是对某种情况的记录,包括数值数据(各种统计资料数据)以及非数值数据(各种图形图像、表格、文字符号、声音、视频等)两种。只有对接受者有用的数据才是信息。信息和数据的关系如图 1-1 所示,示例如图 1-2 所示。

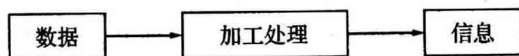


图 1-1 信息和数据的关系

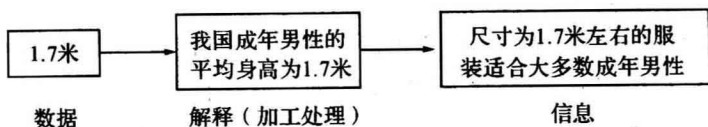


图 1-2 信息和数据的关系示例

1.2 信息在计算机中的表示——信息数字化

1.2.1 信息数字化的定义

根据表示信息的数据类型,信息可分为数值型信息、文本型信息、声音信息、图形信息、图像信息、视频信息。计算机要处理和存储这些信息,首先要把这些信息以计算机能接受的形式表示出来,这就是信息的数字化。根据香农采样定理,数字、文字、图像、语音等各种信息,实际上都可以用 0 和 1 这两个基本元素的编码来表示,这种编码方式也叫作二进制

编码。

通俗地说,把多种类型的信息转变为计算机能处理的数据的技术,就是数字化技术。确切地说,数字化技术指的是运用二进制编码,通过电子计算机、光缆、通信卫星等设备,来表达、传输和处理所有信息的技术,一般包括数字编码、数字压缩、数字传输、数字调制与解调等技术。数字化技术是计算机的基础,也是信息技术的核心。可以这样说,如果没有数字化技术,就没有当今的计算机,因为数字计算机的一切运算和功能都是用数字来完成的。

为什么要选择二进制编码?早期电子计算机使用的是电子管,由于电子管只有开和关两种基本的状态,正好表示为二进制中的0和1。用二进制编码的数据计算也简单,这就决定了以电子管为基础的电子计算机采用二进制来表示数字和数据。不管是何种类型的信息,在计算机内部总要转变成二进制数据。

1.2.2 二进制计数制和其他几种计数制

要表示一个数,日常生活中大家采用的一般是十进制计数方法。而现在我们知道,机器内部所有要处理的信息都是以二进制数据存在,十进制数和二进制数之间怎么进行转换呢?因为二进制数表示一个相对大的数时书写起来比较麻烦,同时也为了十进制数和二进制数转换上的方便,有时候我们还要用到八进制数和十六进制数。这一节主要介绍这几种进位计数制和它们之间相互转换的方法。

(1)进位计数制的基本概念。将数字符号按序排列成数位,并遵照某种由低位到高位进位的原则进行计数,这种表示数值的方式,称作进位计数制。如十进位计数制,就是按照“逢十进一”的原则进行计数的。

进位计数制的表示主要包含三个基本要素:数码、基数和位权。数码指的是一种进位制中用来表示数值的所有数字,如十进位计数制中的数码有0、1、2、3、4、5、6、7、8、9十个数码;基数是指在某种进位计数制中,每个数位上所能使用的数码的个数,如十进位计数制中,每个数位上可以使用的数码为0、1、2、3、...、9十个数码,即其基数为10;逢基进一是进位计数制中进位的原则,如前面提到的十进位计数制,基数为十,所以进位原则就是逢十进一。位权是指一个固定值,一般为当这位的数码为“1”时所表征的数值的大小。不同位的位权是不同的。每个数位上的数码所代表的数值的大小,等于在这个数位上的数码乘上位权。同一个数码所处的位置不同,代表数的大小也不同。例如,在十进位计数制中,有数值132.4,小数点左边第一位位权为值10的0次方,即权值为1,这个位置上的2所代表的数值大小就是 2×1 ;左边第二位位权值为10(10的1次方),所以这里的数码3所代表的数值大小就是 $3 \times 10 = 30$;左边第三位位权值为10的2次方,对应数码1所代表的数值大小就是 $1 \times 100 = 100$;小数点右边第一位位权值为10的-1次方,对应数码4所代表的数值大小就是 $4 \times 0.1 = 0.4$,以此类推。

①十进制。十进位计数制简称十进制,有十个不同的数码符号:0、1、2、3、4、5、6、7、8、9,基为10,逢十进一,位权值是 10^{n-1} , n 为要计算的数码在数中的位置次序号(本节中以下的 n 同此定义)。如小数点前第一个数码, n 为1,小数点前第二个数码,对应的 n 为2,以此类推。每个数码根据它在这个数中所处的位置(数位),按位权来决定其实际数值。

例如: $(215.48)_{10} = 2 \times 10^2 + 1 \times 10^1 + 5 \times 10^0 + 4 \times 10^{-1} + 8 \times 10^{-2}$

②二进制。二进位计数制简称二进制,有两个数码符号:0、1,基为2,逢二进一,位权

为 2^{n-1} 。

例如: $(11001.01)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = (25.25)_{10}$

③八进制。八进位计数制简称八进制,有八个数码符号:0、1、2、3、4、5、6、7,基为8,逢八进一,位权为 8^{n-1} 。

例如: $(162.4)_8 = 1 \times 8^2 + 6 \times 8^1 + 2 \times 8^0 + 4 \times 8^{-1} = (114.5)_{10}$

④十六进制。十六进位计数制简称十六进制,有十六个数码符号:0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F,基为16,逢十六进一,位权为 16^{n-1} 。

例如: $(2BC.48)_{16} = 2 \times 16^2 + B \times 16^1 + C \times 16^0 + 4 \times 16^{-1} + 8 \times 16^{-2} = (700.28125)_{10}$

⑤各种进位制的表示方法。一般情况下,用(数值)下标表示不同进制的数。例如:二进制数101用 $(101)_2$ 表示,十进制数19用 $(19)_{10}$ 表示,八进制数37用 $(37)_8$ 表示。在计算机中,一般在数字的后面用特定字母表示该数的进制,B(英文单词 binary 的第一个字母的大写形式)表示二进制,如101B表示101是个二进制数;D(英文单词 decimal 的第一个字母的大写形式)表示十进制,如59D表示59这个数是十进制数;O(英文单词 octavo 的第一个字母的大写形式)表示八进制,为了避免和中文数字的0相混淆,有时也用Q来代替;H(英文单词 hexadecimal 第一个字母的大写形式)表示十六进制数。

总结以上四种进位计数制,可以将它们的特点概括为每一种计数制都有一个固定的基数;每一个数位可取基数中的不同数值;每一种计数制都有自己的位权,并且遵循“逢基数进一”的原则。如表1-1所示。

表 1-1 常用进位制的总结

常用进制	英文表示符号	数码符号	进位规律	进位基数
二进制	B	0,1	逢二进一	2
八进制	O	0,1,2,3,4,5,6,7	逢八进一	8
十进制	D	0,1,2,3,4,5,6,7,8,9	逢十进一	10
十六进制	H	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F	逢十六进一	16

(2)进位计数制之间的转换。不同进位计数制之间的转换原则是:如果两个有理数相等,则这两个数的整数部分和小数部分一定分别相等。因此,数制之间进行转换时,通常对整数部分和小数部分分别进行转换。

①非十进制数转换为十进制数。当把非十进制数转换为十进制数时,将这个数按权展开,然后求和,所得到的和值即为十进制数。

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

$(125.24)_8 = 1 \times 8^2 + 2 \times 8^1 + 5 \times 8^0 + 2 \times 8^{-1} + 4 \times 8^{-2} = (85.3125)_{10}$

$(3A8.48)_{16} = 3 \times 16^2 + A \times 16^1 + 8 \times 16^0 + 4 \times 16^{-1} + 8 \times 16^{-2} = (936.28125)_{10}$

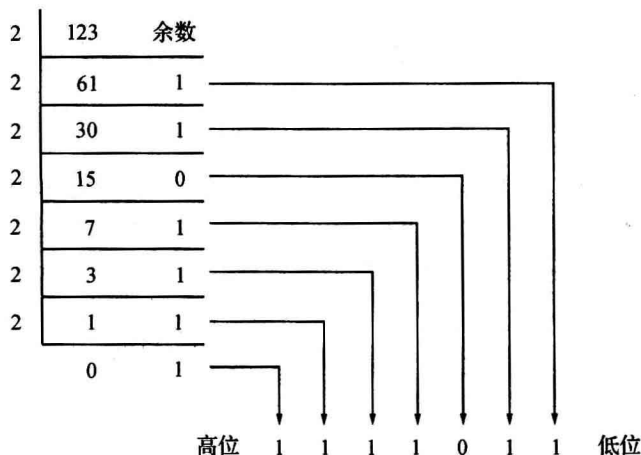
②十进制数转换为非十进制数。把一个十进制数转换为非十进制数时,将十进制数的整数部分逐次除以相应非十进制数的基数,取其各次余数由低位向高位排列,即得到转换后的数的整数部分,这种方法也叫“除基取余法”;将十进制数的小数部分逐次乘以相应进制数的基数,取其各次积的整数,由高位向低位排列后所得到的数,即为转换后的数的小数部分,

这种方法叫“乘基取整法”。

a. 十进制数转换为二进制数。十进制数的整数部分采取“除 2 取余法”，小数部分采取“乘 2 取整法”。

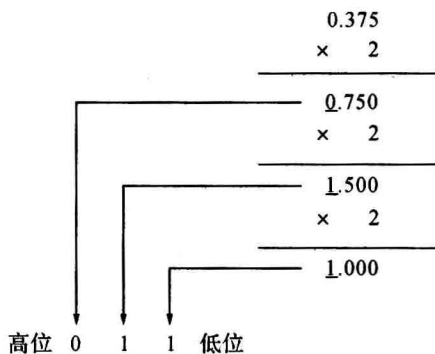
例 1.1: 将十进制数 $(123.375)_{10}$ 转换为二进制数。

步骤一：先将整数部分转换为二进制数，如下所示：



因此，整数部分 $(123)_{10} = (1111011)_2$

步骤二：对小数部分进行转换，如下所示：



因此，小数部分 $(0.375)_{10} = (0.011)_2$

所以， $(123.375)_{10} = (1111011.011)_2$

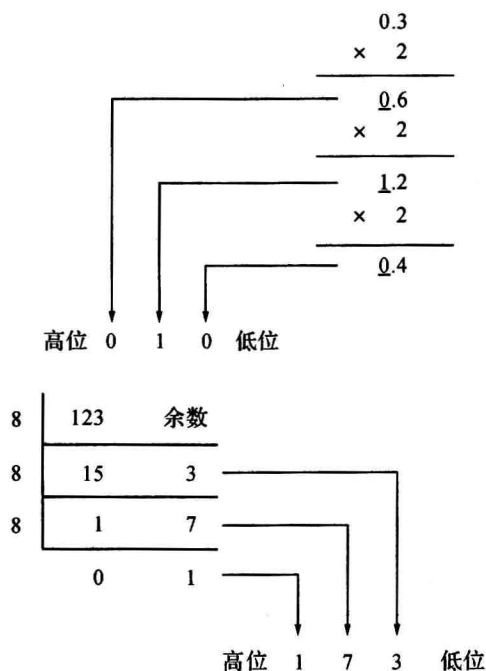
注意，把十进制数的小数部分转换为非十进制数时，最后的结果不一定能使小数部分全为 0，此时要根据对结果的精度要求确定什么时候终止计算。以十进制数 0.3 为例，如要把 0.3 转换为二进制数，经过有限次的计算后发现，不管乘多少次 2，最后的结果小数部分都不会全为 0。如果我们要精确到小数点后三位，经过三次乘 2 计算，可以得到有误差的结果：

即 $(0.3)_{10} = (0.010)_2$

b. 十进制数转换为八进制数。同理，当十进制数转换为八进制数时，整数部分的转换采取“除 8 取余法”，小数部分的转换采取“乘 8 取整法”。

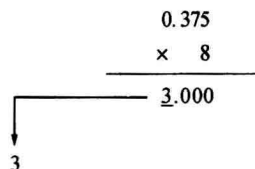
例 1.2: 将十进制数 $(123.375)_{10}$ 转换为八进制数。

步骤一：先将整数部分转换为八进制数，如下所示：



整数部分 $(123)_{10} = (173)_8$

步骤二：对小数部分进行转换，如下所示：



小数部分 $(0.375)_{10} = (0.3)_8$

因此， $(123.75)_{10} = (173.3)_8$

c. 十进制数转换为十六进制数。同样的，十进制数转换为十六进制数时，整数部分采取“除 16 取余法”，小数部分采取“乘 16 取整法”。仍以十进制的 123.75 为例，可以得到 $(123.75)_{10} = (7B.C)_{16}$ 。

(3) 非十进制数之间的相互转换。

① 八进制数与二进制数之间的转换。由于 1 位八进制数和 3 位二进制数存在完全的对对应关系，如表 1-2 所示。因此，要将八进制数转换成二进制数时，只需以小数点为界，向左或向右每 1 位八进制数用相应的 3 位二进制数取代即可。反之，二进制数转换成相应的八进制数，只要以小数点为界，向左或向右每 3 位二进制数用相应的 1 位八进制数取代，不足 3 位，用零补足即可。

表 1-2 八进制数和二进制数的对应关系

二进制	000	001	010	011	100	101	110	111
八进制	0	1	2	3	4	5	6	7

例 1.3:将八进制数 $(357.162)_8$ 转换成二进制数。

$$\begin{array}{ccccccc} 3 & 5 & 7 & . & 1 & 6 & 2 \\ \hline 011 & 101 & 111 & & 001 & 110 & 010 \end{array}$$

即 $(357.162)_8 = (11101111.0011101)_2$

例 1.4:将二进制数 $(11011110.10110001)_2$ 转换成八进制数。

$$\begin{array}{ccccccc} 011 & 011 & 110 & . & 101 & 100 & 010 \\ \hline 3 & 3 & 6 & & 5 & 4 & 2 \end{array}$$

即 $(11011110.10110001)_2 = (336.542)_8$

②十六进制数与二进制数之间的转换。由于1位十六进制数和4位二进制数存在完全的对对应关系,如表1-3所示。因此,要将十六进制数转换成二进制数时,只需以小数点为界,向左或向右每1位十六进制数用相应的4位二进制数取代即可。反之,二进制数转换成相应的十六进制数,只要以小数点为界,向左或向右每4位二进制数用相应的1位十六进制数取代,不足4位,用零补足即可。

表 1-3 十六进制数和二进制数的对应关系

二进制	0000	0001	0010	0011	0100	0101	0110	0111
十六进制	0	1	2	3	4	5	6	7
二进制	1000	1001	1010	1011	1100	1101	1110	1111
十六进制	8	9	A	B	C	D	E	F

例 1.5:将十六进制数 $(5AB.8CE)_{16}$ 转换成二进制数。

$$\begin{array}{ccccccc} 5 & A & B & . & 8 & C & E \\ \hline 0101 & 1010 & 1011 & & 1000 & 1100 & 1110 \end{array}$$

即 $(5AB.8CE)_{16} = (10110101011.10001100111)_2$

例 1.6:将二进制数 $(1100101001011.001100101)_2$ 转换成十六进制数。

$$\begin{array}{ccccccc} 0001 & 1001 & 0100 & 1011 & . & 0011 & 0010 & 1000 \\ \hline 1 & 9 & 4 & B & . & 3 & 2 & 8 \end{array}$$

即 $(1100101001011.001100101)_2 = (194B.328)_{16}$

1.2.3 各类信息的数字化

(1)数值信息的表示。所有数值型信息在机器中的表示形式统称为机器数。机器数把数值数据的符号数字化。实用的数据有正数和负数,计算机分别用0、1来表示数据的正、负号,通常放在二进制数的最高位,称符号位。因为有符号位占据一位,所以机器数的形式值就不等于真正的数值。带符号位的机器数对应的真正数值称为该机器数的真值。

机器数有三种表示形式——原码、反码、补码。由于计算机中运算器的位数是固定的,在下面的介绍中,我们以8位二进制数举例。

①原码。正数的符号位为0,负数的符号位为1,其他为数值位,存放该数绝对值的二进制形式。用这种方法得到的数就是该数的原码。8位二进制原码的表示范围: $-127 \sim +127$ 。

例如: 符号位 数值位

$$[+7]_{\text{原}} = 0 \quad 0000111 \quad \text{B}$$

$$[-7]_{\text{原}} = 1 \quad 0000111 \quad \text{B}$$

注意:数 0 的原码有两种形式:

$$[+0]_{\text{原}} = 00000000\text{B}, [-0]_{\text{原}} = 10000000\text{B}$$

②反码。正数的反码与其原码相同;负数的反码是对其原码除符号位外,数值部分逐位取反。8 位二进制反码的表示范围: $-127 \sim +127$ 。

例如: 符号位 数值位

$$[+7]_{\text{反}} = 0 \quad 0000111 \quad \text{B}$$

$$[-7]_{\text{反}} = 1 \quad 1111000 \quad \text{B}$$

注意:数 0 的反码也有两种形式:

$$[+0]_{\text{反}} = 00000000\text{B}, [-0]_{\text{反}} = 11111111\text{B}$$

③补码。正数的补码与其原码相同;负数的补码是在其反码的末位加 1。8 位二进制数补码的表示范围为 $-128 \sim +127$ 。

例如: 符号位 数值位

$$[+7]_{\text{补}} = 0 \quad 0000111 \quad \text{B}$$

$$[-7]_{\text{补}} = 1 \quad 1111001 \quad \text{B}$$

注意:与原码、反码不同,数值 0 的补码只有一个,即 $[0]_{\text{补}} = 00000000\text{B}$ 。

补码在计算机中是一种重要的编码方式,采用补码后,可以方便地将减法运算化成加法运算,运算过程得到简化。正数的补码即是它所表示的数的真值,而负数的补码的数值部分,不是它所表示的数的真值。

我们在计算两个数的减法运算中,一般要先比较两个数的大小,用大的数减去小的数,然后根据两个数绝对值的大小来决定符号。能否统一用加法来实现呢? 这里先来看一个事实:对于一个钟表,将时针从 6 拨到 9,可以顺拨 3,也可以倒拨 9,其效果是一样的,用式子表示就是: $6+3$ 等价于 $6-9$ 。对于钟表来说 $[-9]$ 的补码就是 3。

例 1.7:利用补码表示法计算 $6-9$

$$[6]_{\text{补}} = 00000110$$

$$[-9]_{\text{补}} = 11110111$$

$$\begin{array}{r} 00000110 \\ +11110111 \\ \hline 11111101 \end{array}$$

$$\text{所以 } 6-9=[6]_{\text{补}}+[-9]_{\text{补}}=00000110+11110111=11111101=[-3]_{\text{补}}$$

采用补码表示法的最大优点就是:可以把减法运算转换为加法运算,符号位一起参加运算,不需要比较大小,运算结果的符号自动确定。采用补码进行运算,所得结果仍为补码。表 1-4 列出了 8 位二进制数码的各种表示方法。

表 1-4 8 位二进制数码的各种表示方法

二进制数码	无符号二进制数	原码	反码	补码
00000000	0	+0	+0	+0

续 表

二进制数码	无符号二进制数	原码	反码	补码
00000001	1	+1	+1	+1
00000010	2	+2	+2	+2
...	...	...	...	...
01111110	126	+126	+126	+126
01111111	127	+127	+127	+127
10000000	128	-0	-127	-128
10000001	129	-1	-126	-127
10000010	130	-2	-125	-126
...	...	...	...	...
11111101	253	-125	-2	-3
11111110	254	-126	-1	-2
11111111	255	-127	-0	-1

(2) 西文字符的编码。计算机中主要用 ASCII 码 (American Standard Code for Information Interchange) 来表示西文字符。ASCII 码即美国标准信息交换码, 已被国际标准化组织 (ISO) 定为国际标准。

7 位 ASCII 码称为基本 ASCII 码, 是国际通用的。用 7 位二进制编码 0000000~1111111 可以表示 128 种字符, 其中包括 34 种控制字符, 52 个英文大小写字母, 10 个数字, 32 个字符和运算符 (详见表 1-5)。计算机中用一个字节 (8 位二进制位) 表示 7 位 ASCII 码, 这个字节的最高位为 0, 其余 7 位为 ASCII 编码, 范围为 00000000B~01111111B。

表 1-5 ASCII 码表 (设编码为: $b_7 b_6 b_5 b_4 b_3 b_2 b_1$)

$b_7 b_6 b_5$ $b_4 b_3 b_2 b_1$	000	001	010	011	100	101	110	111
0000	NUL	DLE	空格	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS 退格	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y

续 表

$b_7 b_6 b_5$ $b_4 b_3 b_2 b_1$	000	001	010	011	100	101	110	111
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR 回车	GS	_	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	-	o	DEL

(3) 汉字编码。

① 汉字输入码, 又称“外部码”, 简称“外码”, 指用户从键盘上输入代表汉字的编码。根据所采用输入方法的不同, 外码大体可分为数字编码(如区位码)、字形编码(如五笔字型)、字音编码(如各种拼音输入法)和音形码等。如汉字“啊”采用五笔字型输入时, 编码为“kbsk”, 用区位码方式输入时编码为“1601”, 那么这里的“kbsk”和“1601”就称为外码。

区位码是一种最通用的汉字输入码。它根据我国国家标准 GB2312-80(《信息交换用汉字编码字符集》), 按照将 6763 个汉字和一些常用的图形符号, 分为 94 个区, 每区 94 个位的方法, 把它们定位在一张表上, 成为区位码表。其中 1~9 区分布的是一些符号; 16~55 区为一级字库, 共 3755 个汉字, 按音序排列; 56~87 区为二级字库, 共 3008 个汉字, 按部首排列。

区位码表中, 每个汉字或符号的区位码由两个字节组成, 第一个字节为区码, 第二个字节为位码, 区码和位码分别用一个两位的十进制数来表示, 这样区码和位码合起来就形成了一个区位码。如“啊”字位于第 16 区第 01 位, 则“啊”字的区位码为: 区码+位码, 即 1601。

国家标准 GB2312-80 中的汉字代码除了十进制形式的区位码外, 还有一种十六进制形式的编码, 称为国标码。国标码是在不同汉字信息系统间进行汉字交换时所使用的编码。需要注意的是, 在数值上, 区位码和国标码是不同的, 国标码是在十进制区位码的基础上, 其区码和位码分别加上十六进制数 20H 所得来的。

② 汉字机内码, 又称“汉字 ASCII 码”“机内码”, 简称“内码”, 由扩充的 ASCII 码组成, 指计算机内部存储、处理加工和传输汉字时所用的由 0 和 1 符号组成的代码。输入码被接受后就由汉字操作系统的“输入码转换模块”转换为机内码, 与所采用的键盘输入法(汉字输入码)无关。

机内码是汉字最基本的编码, 不管是什么汉字系统和汉字输入方法, 输入的汉字外码到机器内部都要转换成机内码, 才能被存储和进行各种处理。我们通常所说的内码是指国标内码, 即 GB 内码。GB 内码用两个字节来表示(即一个汉字要用两个字节来表示), 每个字节的高位为 1, 以确保 ASCII 码的西文与双字节表示的汉字之间的区别。

机内码与区位码的转换过程是: 将十进制区位码的区码和位码部分首先分别转换成十六进制, 再在其区码和位码部分分别加上十六进制数 A0H 构成。

③ 汉字的区位码、国标码与机内码的关系。将一个汉字的区号和位号分别转换成十六进制数, 然后分别加上十六进制数 20H, 就成为此汉字的国标码。

例如：“中”字的输入区位码是 5448，分别将其区号 54 转换为十六进制数 36H，位号 48 转换成十六进制数 30H，即 3630H；然后，再把区号和位号分别加上 20H，得到“中”字的国标码： $3630H + 2020H = 5650H$ 。

汉字的内码 = 汉字的国标码 + 8080H。

例如：已知“中”字的国标码为 5650H，则根据上述公式得出：

“中”字的内码 = “中”字的国标码 5650H + 8080H = D6D0H

④ 汉字字模和汉字字库。汉字字模也称汉字字形码，是指存放在字库中的汉字字形点阵码。不同的字体有不同的字库，如黑体、仿宋体、楷体等是不同的字体。点阵的点数越多，一个字的表达质量也越高。一般用于显示的字形码是 16×16 点阵的，每个汉字在字库中占 $16 \times 16 / 8 = 32$ 个字节；一般用于打印的是 24×24 点阵字形的，每个汉字占 $24 \times 24 / 8 = 72$ 个字节；在一个 48×48 点阵字形中，每个汉字占 $48 \times 48 / 8 = 288$ 个字节。

一个汉字的点阵信息叫作该字的字形或字模。存放在存储器中的常用汉字和符号的字模集合就是汉字字形库，也称汉字字模库，简称汉字库。

例 1.8：已知一个汉字的字形码用 24×24 点阵表示，则在内存中存储 100 个这样的汉字需要使用多少存储空间？

计算步骤如下：

存储空间 = $24 \times 24 / 8 \times 100 = 7200$ 字节

(4) 图形和图像信息的数字化。图像数字化是计算机处理图像信息之前的基本步骤，目的是把真实的图像转变成计算机能够接受的形式。自然界的图像要数字化，首先就要通过数码相机、扫描仪等设备对其进行采样，方法是将画面分成一个个的像素，每个像素就是一个采样点，水平方向上像素的个数称为水平分辨率，垂直方向上像素的个数就是垂直分辨率。对于同一张图像来说，用越多的点来记录，其效果越真实，这也是为什么像素高的数码相机照出来的图像比较清晰的缘故。所以说采样的实质是确定用多少点来描述一张图像，实际上就是图像的分辨率，比如一张分辨率为 640×480 的图像，表示这幅图由 307200 个点组成。

采样以后要进行量化，量化指的是确定要使用多大范围的数值，来表示图像采样之后的每一个点，这个数值范围就是图像所能使用的颜色总数。例如，如果用 8 位二进制数来表示一个点，则表示每个点可以有 256 种颜色。数值范围越大，表示图像可以拥有更多的颜色，图像效果更为细致，当然需要的存储空间也越大。一张没有经过压缩的图像所占用的存储空间用如下公式计算：

字节数 = 水平分辨率 $\times$ 垂直分辨率 $\times$ 颜色深度(位) / 8

如一张分辨率为 640×480 、256 色的图像，需要的存储空间是 $640 \times 480 \times 8 / 8 = 307200$ 字节。

在采样和量化以后，才能产生一张数字化的图像，再利用计算机处理软件，对图像进行修饰或转换，以达到所希望的图像效果。

数字化的图像在计算机中有两种存储方式，一种是位图模式，另一种是矢量模式。位图又称点阵图，是将图像中每一点的数值存放在以字节为单位的内存单元中。当图像是单色时，一个字节可以存放 8 个采样点的图像数据；对于 16 色图像，一个字节可以存放两个点的数据。位图模式适合存储复杂图像，比如照片等图像，但图像在放大和缩小的过程中容易失真，再就是所占据的存储空间较大。矢量模式下，计算机不存储图像中的每个点，而只存储