

全国计算机等级考试 (培训辅导教程)

二级教程

基础知识及问题解答

- ▶ 紧扣教学大纲，注重基础知识培养
- ▶ 突出重点难点，深入解答常见问题
- ▶ 强调实际操作，提供系统复习支持

高等教育出版社

(京)112号

图书在版编目(CIP)数据

二级教程基础知识及问题解答/《二级教程基础知识及问题解答》编写组编. —北京:
高等教育出版社, 2000.1

全国计算机等级考试培训辅导教程

ISBN 7-04-007912-7

I.二… II.二… III.电子计算机—水平考试—自学参考资料 IV.TP3

中国版本图书馆 CIP 数据核字(2000)第 01642 号

二级教程基础知识及问题解答

本书编写组

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

邮政编码 100009

电 话 010-64054588

传 真 010-64014048

网 址 <http://www.hep.edu.cn>

经 销 新华书店北京发行所

印 刷 中国青年出版社印刷厂

开 本 787×1092 1/16

版 次 2000 年 1 月第 1 版

印 张 12

印 次 2000 年 1 月第 1 次印刷

字 数 280 000

定 价 18.00 元

凡购买高等教育出版社图书,如有缺页、倒页、脱页等
质量问题,请在所购图书销售部门联系调换。

版权所有 侵权必究

编写说明

全国计算机等级考试是教育部考试中心面向社会推出的一种全国性考试。全国计算机等级考试是一种重视应试人员对计算机和软件应用能力的考试,因此,它不限制报考人员的学历背景和年龄。这种开放性的、公正的、客观的考试为各行各业计算机应用人员能力的测试提供了统一、客观的标准。开考几年来,产生了良好的社会效益。参加该考试的考生从开考第一年1994年的10 000余人上升到1998年的逾50万人,考试也从初期的每年1次发展到每年2次。

随着参加考试的人数的增多,对培训教程和适合不同层次考生的辅导用书的需求数量也在上升;与此同时,对教材的质量也提出了更高的要求。本丛书就是在这种情况下应运而生的。

适逢教育部于1998年9月颁布了新的计算机等级考试大纲,为此,本书的主编约请国内部分高等学校从事计算机等级考试教学第一线工作的教师和一些对计算机普及教育有经验的同仁,根据新的大纲编写了《全国计算机等级考试培训辅导教程》,作为部颁的《全国计算机等级考试培训教程》的配套辅导读物。在编写的过程中主要注意了以下几点:

1. 紧扣教学大纲,注重基础知识培养

本书针对教育部考试中心1998年最新颁布的《计算机等级考试大纲》组织编写,并根据大纲的最新调整适当地调整了教材内容。

2. 突出重点难点,深入解答常见问题

本书的作者既有参与等级考试大纲制订的专家,又有许多来自教学第一线的教师,因此,本教材既能紧扣大纲,深入浅出地讲解基础知识;又能突出重点与难点,为考生系统地复习提供方便,同时给予必要的应试指导。

3. 注重实际操作,提供系统复习指导

本教材分为基础知识、问题与难点解答、习题与例题分析,各部分比例恰当,方便学校教学与学生课后自学。它是集教师教学、学生自学、应试复习于一体的实用教材。

总之,在努力贯彻新的考试大纲,按照新的考试大纲来组织内容的同时,编者兼顾课堂教学和考生考前系统自学或复习的需要,在讲解基本知识的同时,注意分析难点,着力解决易混淆的概念,纠正错误的观点——这是集教师教学、学生自学、考前系统复习为一体的新思维教材。

本书是《全国计算机等级考试培训辅导教程》的一册，该书的主要内容是二级考试所涉及到的基础知识，它也是新的等级考试大纲颁布后增加的一门新课程。主要包括：数制转换的基础知识、计算机系统的组成与应用、DOS 操作系统与 Windows 操作系统的基本操作、计算机安全的基础知识及常用的杀病毒软件的使用、多媒体与网络基础知识。本书可作为广大考生参加等级考试的学习辅导书。也适合有关学校课堂教学使用。

本书是集体劳动的结晶，参与工作的主要有首都师范大学参与计算机等级考试培训一线教学的教师，由王旭仁执笔编写。全书由教育部全国计算机等级考试教材编审委员会成员唐兆亮审阅。

虽然全书编写组作了大量细致的工作，但肯定还有不少谬误之处，欢迎广大读者多提意见，以利再版更正。

编者

1999 年 10 月

目 录

第1章 数制、数据单位及编码

1.1 计算机的数制	1
1.1.1 什么是进位计数制	1
1.1.2 十进制数与二进制数	2
1.1.3 八进制数与十六进制数	3
1.2 二进制数与十进制数的转换	4
1.3 十六进制与其他进制的数	6
1.4 数据单位、字符编码、汉字编码	7
1.4.1 计算机中的数据单位	7
1.4.2 字符编码	8
1.5 问题与解答	11
1.6 习题与答案	16
1.6.1 填空题	16
1.6.2 选择题	16
1.6.3 答案	22

第2章 计算机系统基础知识

2.1 计算机系统的组成	23
2.2 微型计算机的硬件系统	24
2.2.1 中央处理器	24
2.2.2 存储器	25
2.2.3 输入和输出设备	27
2.3 微型计算机的软件系统	27
2.3.1 软件系统的分类	27
2.3.2 操作系统	27
2.3.3 程序设计语言	27
2.4 计算机的应用领域	28
2.5 问题与解答	29
2.6 习题与答案	30
2.6.1 填空题	30
2.6.2 选择题	31
2.6.3 答案	37

第3章 DOS 操作系统

3.1 了解 DOS 操作系统	39
3.1.1 DOS 的组成	40
3.1.2 常用功能键和控制键	40
3.1.3 DOS 的启动	41
3.1.4 DOS 文件	42
3.1.5 盘符、目录和路径	44
3.1.6 DOS 的命令	46
3.2 文件操作命令	47
3.2.1 文件复制命令 COPY	47
3.2.2 显示文件内容命令 TYPE	52
3.2.3 删除文件命令 DEL	54
3.2.3 改变文件名命令 REN	56
3.2.4 复制文件和目录命令 XCOPY	57
3.2.5 设置和查看文件属性命令 ATTRIB	59
3.3 目录操作命令	61
3.3.1 文件和目录列表命令 DIR	61
3.3.2 显示和改变当前目录命令 CD	63
3.3.3 建立子目录命令 MD	66
3.3.4 删除子目录命令 RD	68
3.3.5 显示目录结构命令 TREE	69
3.3.6 设置查找目录命令 PATH	71
3.4 磁盘操作命令	72
3.4.1 格式化磁盘命令 FORMAT	72
3.4.2 整盘复制软盘命令 DISKCOPY	76
3.4.3 检查磁盘状态命令 CHKDSK	76
3.5 功能操作命令	79
3.5.1 显示 DOS 版本号命令 VER	79
3.5.2 设置系统日期命令 DATE	79
3.5.3 设置系统时间命令 TIME	80
3.5.4 清除屏幕命令 CLS	81
3.5.5 帮助命令 HELP	81

3.6 批处理命令	84	5.4 多媒体技术	130
3.6.1 批处理的概念	84	5.4.1 媒体的概念	130
3.6.2 建立批处理文件	85	5.4.2 多媒体技术的概念	131
3.6.3 自动批处理文件	86	5.4.3 多媒体计算机系统	131
3.6.4 批处理子命令	86	5.5 习题与答案	131
3.7 输入输出改向	87	5.5.1 填空题	131
3.7.1 输入改向命令	87	5.5.2 选择题	132
3.7.2 输出改向命令	87	5.5.3 答案	134
3.8 问题与解答	89		
3.9 习题与答案	93		
3.9.1 填空题	93		
3.9.2 选择题	94		
3.9.3 答案	101		
第4章 计算机安全			
4.1 计算机的安全操作	103	6.1 了解 Windows	136
4.2 计算机病毒的防治	104	6.1.1 DOS 操作系统	136
4.2.1 计算机病毒的概述	104	6.1.2 Windows 3.x 操作系统	136
4.2.2 计算机病毒的检测与消除	105	6.1.3 安装 Windows 3.x	137
4.3 常用杀毒软件	106	6.1.4 Windows 3.x 的启动和退出	138
4.3.1 KILL 的使用	106	6.2 Windows 3.x 的基本组成	140
4.3.2 新版 KV300 的使用	108	6.2.1 窗口	140
4.4 问题与解答	111	6.2.2 对话框	141
4.5 习题与答案	116	6.2.3 菜单	142
4.5.1 填空题	116	6.3 Windows 基本操作	144
4.5.2 选择题	116	6.3.1 使用窗口	144
4.5.3 答案	117	6.3.2 使用菜单	145
第5章 计算机网络与多媒体技术			
5.1 计算机网络常识	119	6.4 程序管理器	146
5.1.1 网络的发展	119	6.4.1 程序组和程序项	146
5.1.2 计算机网络的一般概念	121	6.4.2 运行应用程序	151
5.1.3 网络的传输介质	122	6.5 了解多任务系统和文件管理器	153
5.1.4 网络中的数据通信	123	6.5.1 前台和后台程序	153
5.1.5 网络的拓扑结构	123	6.5.2 切换前后台程序	153
5.2 局域网	126	6.5.3 了解文件管理器	154
5.2.1 局域网的概念	126	6.6 输入汉字	155
5.2.2 局域网的软硬件设备	126	6.6.1 启动汉字输入法	156
5.3 Internet 简介	129	6.6.2 输入汉字	158
5.3.1 Internet 的起源和发展	129	6.6.3 设置输入法	159
5.3.2 Internet 提供的服务方式	129	6.7 了解 Windows 95	160
		6.7.1 Windows 95 的特点	160
		6.7.2 Windows 95 的桌面系统	161
		6.7.3 关于“我的电脑”	164
		6.7.4 “开始”按钮	166
		6.8 问题与解答	169
		6.9 习题与答案	178
		6.9.1 填空题	178

6.9.2 判断题	178	6.9.5 操作练习	179
6.9.3 选择题	179	6.9.6 答案	179
6.9.4 简答题	179		

第 1 章

数制、数据单位及编码

学习指导

本章主要介绍计算机的数制基础、二进制数的运算、数制间的转换、计算机的数据单位、字符编码、汉字编码。

计算机等级考试大纲中对本章的要求是：

- 计算机的常用数制
- 二进制、十六进制及其与十进制之间的转换
- 数据基本单位

1.1 计算机的数制

计算机的基本功能之一就是进行计算。计算机由数量巨大的电子元器件与集成电路组成，那么在这些设备中如何表示数字呢？这就涉及到二进制，它是计算机的数学基础。

1.1.1 什么是进位计数制

数制有非进位计数制和进位计数制两种。

1. 非进位计数制

非进位计数制特点是表示数值大小的数码与它在数中的位置无关。典型的非进位计数制是罗马数字。例如，罗马数字中：I 总是代表 1，II 总是代表 2，IV 总是代表 4，V 总是代表 5，X 总是代表 10，C 总是代表 100 等等……

非进位计数制表示数据不便、运算困难，现已不常用。

2. 进位计数制

进位计数制的特点是表示数值大小的数码与它在数中所处的位置有关。例如，十进制数 123.45，数码 1 处于百位上，它代表 $1 \times 10^2 = 100$ ，即 1 所处的位置具有 10^2 权；2 处于十位

上,它代表 $2 \times 10^1 = 20$,即 2 所处的位置具有 10^1 权;以此类推,3 代表 $3 \times 10^0 = 3$;而 4 处于小数点后第一位,代表 $4 \times 10^{-1} = 0.4$;最低位 5 处于小数点后第二位,代表 $5 \times 10^{-2} = 0.05$,如此等等……

十进制运算中,凡是达到 10 就向高位进一位,相邻两位间是十倍的关系,10 称为进位“基数”。可以想像:若是二进制,则进位基数应该是 2;八进制进位基数为 8;十六进制则进位基数应该是 16。

十进制数共有 0~9 十个数码,十进制数就是由这十个数码及其他一些符号(小数点、正负号)组成。相应地,二进制数的数码为:0 与 1;八进制数有八个数码:0~7;十六进制数有 16 个数码:0~15(10 至 15 分别由 A~F 表示)。

1.1.2 十进制数与二进制数

人们习惯于使用十进制数 0~9。逢十进一,借一当十,这完全是现在人们的习惯。其实,古埃及人与古巴比伦人就曾经使用过六十进制与十二进制。那么为什么在计算机中却偏偏采用古怪的二进制呢?其主要原因有以下四点:

(1)电路简单。

计算机是由逻辑电路组成的,逻辑电路通常只有两种状态。例如:开关的接通与断开,晶体管的饱和与截止,电压电平的高与低等。这两种状态正好与二进制数的两个数码 0 和 1 相似,可由这两种状态分别代表两数码。若是采用十进制,则需表示十个数码,这是很困难的。

(2)工作可靠。

两种状态表示的二进制两个数码,数字传输和处理不容易出错,因而电路更加可靠。

(3)运算简单。

二进制运算法则简单。例如,求和法则只有 3 个,求积法则也只有 3 个。若采用十进制,则其运算法则在电路上实现是很困难的。

(4)逻辑性强。

计算机的工作机理是建立在逻辑运算基础上的,逻辑代数是逻辑运算的理论依据。二进制只有两个数码。正好代表逻辑代数中的“真”与“假”。

二进制数只有“0”和“1”两个数码,而且由低位向高位进位时逢二进一。像 101、110、110.011 等都是二进制数,但是以上三个数也可以认为是十进制数,为了表示它们的区别,可以给这些数字加上括号和下标,标明是几进制的数,例如:

$(101)_2$ 、 $(110)_2$ 表示二进制的数; $(110)_{10}$ 、110 表示十进制的数。

一个十进制数 525,在十进制中说它是 5 个百、2 个十、5 个一的和,也就是:

$$525 = 500 + 20 + 5 = 5 \times 100 + 2 \times 10 + 5 \times 1 = 5 \times 10^2 + 2 \times 10^1 + 5 \times 10^0$$

又如:

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

所以任意一个十进制数都可以表示成:

$$N = d_m \times 10^m + d_{m-1} \times 10^{m-1} + \dots + d_0 \times 10^0 + d_{-1} \times 10^{-1}$$

$$+ d_{-2} \times 10^{-2} + \cdots + d_{-n} \times 10^{-n} = \sum_{i=-n}^m d_i 10^i \quad (n, m \geq 0) \quad (1-1)$$

上式中, $\sum$ 是求和符号; d_i 表示各个位上的数字; m 表示 10 的次幂。

对于第一个例子的十进制数 525:

$$n = 0, m = 2, d_2 = 5, d_1 = 2, d_0 = 5;$$

对于第二个例子的十进制数 123.45:

$$n = 2, m = 2, d_2 = 1, d_1 = 2, d_0 = 3, d_{-1} = 4, d_{-2} = 5。$$

这里把 10 叫做权, 把式(1-1)叫做十进制数的按权展开式。基数实际上表明了每一位上可取的数字的个数, 如 10 进制; 每位上可以有 0, 1, 2, $\cdots$, 9 十个数字; 二进制每一位上可以有 0, 1 两个数字。于是, 可得到一个结论: 对于任意 r 进制数, 可能出现的数字是 0, 1, 2, $\cdots$, $r-1$, 共 r 个。

把式(1-1)中的 10 用 r 来代替:

$$N = d_m \times r^m + d_{m-1} \times r^{m-1} + \cdots + d_0 \times r^0 + d_{-1} \times r^{-1} + d_{-2} \times r^{-2} + \cdots + d_{-n} \times r^{-n} = \sum_{i=-n}^m d_i r^i \quad (m \geq 0, n \geq 0) \quad (1-2)$$

式(1-2)是任意进制的按权展开式。取式中 $r=2$, 那么每一位上可取的数字就只有 0 和 1, 这就是计算机中广泛使用的二进制数。对于二进制数可以把式(1-2)写成:

$$N_2 = b_m \times 2^m + b_{m-1} \times 2^{m-1} + \cdots + b_0 \times 2^0 + b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + \cdots + b_{-n} \times 2^{-n} = \sum_{i=-n}^m b_i 2^i \quad (m, n \geq 0) \quad (1-3)$$

那么, 上面提到的几个二进制数可以表示成:

$$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0; \quad (110)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$(110.011)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$$

每一个十进制数都能找到相对应的二进制数, 一些简单数字的二进制和十进制对照见表 1-1 所示。

表 1-1 十进制与二进制对照

十进制	10	9	8	7	6	5	4	3	2	1	0	0.5	0.25	0.125	0.0625
二进制	1010	1001	1000	111	110	101	100	11	10	1	0	0.1	0.01	0.001	0.0001

1.1.3 八进制数与十六进制数

二进制的缺点是书写较长, 不便于阅读, 为此人们常用八进制数与十六进制数来表示二进制数。

对于式(1-2)取 $r=8$, 就得到八进制数的展开形式。八进制数有 0, 1, 2, 3, $\cdots$, 7 八个数字, 运算规则是“逢八进一, 借一当八”。

因为八与十六都是二的整数倍, 所以在计算机中也有广泛的应用。八进制与十六进制书写容易、易读、易记, 这是通常一些二进制代码都用八进制和十六进制来表示的原因。

对于十六进制数, 按照式(1-2)取 $r=16$, 就得到 16 进制数的展开形式。但是十六进制数

要有十六个数字,而常用的阿拉伯数字只有 0~9 十个数字,另外的几个数字怎么表示呢?我们采用 A~F 来表示其余的 6 个数字。

表 1-2 给出八进制数、十六进制数、十进制数和二进制数的对照情况。

表 1-2 八进制数、十六进制数、十进制数和二进制数的对照

十进制数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
二进制数	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
八进制数	000	001	002	003	004	005	006	007	010	011	012	013	014	015	016	017
十六进制数	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F

八进制数可用括号加上下标来表示,如: $(123)_8$, $(376)_8$ 等,以便与 10 进制数区别开。

十六进制可以用相同的方法来表示,如: $(3FD)_{16}$, $(068E)_{16}$ 等。

由于十进制数的英文是“Decimal”,所以有的书上也用数字后加上英文“d”或“D”来表示,如: $126 = (126)_{10} = 126D = 126d$ 。

二进制数的英文是“Binary”,所以用二进制数后加上“B”或“b”来表示,如: $(11000)_2 = 11000B = 11000b$;同样,十六进制数可以在数字后加上“H”或“h”来表示。八进制数可以在数字后加上“O”或“o”来表示,如: $(3FD)_{16} = 3FDH = 3FDh$, $(321)_8 = 321O = 321o$ 。

1.2 二进制数与十进制数的转换

1. 二进制数转换为十进制数

由公式(1-2)不难得出二进制数转换为十进制数的如下规则。

规则:先将二进制数按权展开成式(1-3)的形式,然后再把式(1-3)的各项相加,即得二进制数的十进制表示形式。

这个规则显而易见,在此不做证明。例:

$$(101.11)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = (5.75)_{10}$$

$$\begin{aligned} (101.11101)_2 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} \\ &\quad + 1 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5} \\ &= (5.90625)_{10} \end{aligned}$$

2. 十进制数转换为二进制数

十进制数转换为二进制数要分整数部分和小数部分来转换。

(1) 整数部分的转换。

由

$$S = K_n 2^n + K_{n-1} 2^{n-1} + \cdots + K_0 2^0$$

得

$$\frac{S}{2} = (K_n 2^{n-1} + K_{n-1} 2^{n-2} + \cdots + K_1 2^0) + \frac{K_0}{2}$$

显然,括号内为商, K_n 是余数, $K_n = 0$ 或 $K_n = 1$ 。

继续以商为被除数,令:

$$S_1 = K_n 2^{n-1} + K_{n-1} 2^{n-2} + \cdots + K_1 2^0$$

$$\frac{S_1}{2} = (K_n 2^{n-2} + \cdots + K_2 2^0) + \frac{K_1}{2}$$

这样进行 n 次后

$$\frac{S_n}{2} = \frac{K_n}{2}$$

就得到了一系列的数字:

$$K_0, K_1, K_2, \cdots, K_n$$

$$K_i = 0 \text{ 或 } K_i = 1, \quad (i = 0, 1, \cdots, n);$$

将这些数字反序排列就能得到:

$$K_n K_{n-1} \cdots K_0$$

这便是所要求的二进制数。

于是我们得到将十进制数转换为二进制数的规则。

规则:十进制整数转化为二进制数时,该十进制数除以 2,并记录余数,然后继续用所得的商数除以 2,并记录余数,如此反复下去一直到商数为零,将所得余数反序排列,就得到该十进制数的二进制表示形式。这种转换的方法叫做“除基倒取余法”。

例:

$$(326)_{10} = (101000110)_2$$

$$326 \div 2 = 163 \cdots 0$$

$$163 \div 2 = 81 \cdots 1$$

$$81 \div 2 = 40 \cdots 1$$

$$40 \div 2 = 20 \cdots 0$$

$$20 \div 2 = 10 \cdots 0$$

$$10 \div 2 = 5 \cdots 0$$

$$5 \div 2 = 2 \cdots 1$$

$$2 \div 2 = 1 \cdots 0$$

$$1 \div 2 = 0 \cdots 1$$

$$(31)_{10} = (11111)_2$$

$$31 \div 2 = 15 \cdots 1$$

$$15 \div 2 = 7 \cdots 1$$

$$7 \div 2 = 3 \cdots 1$$

$$3 \div 2 = 1 \cdots 1$$

$$1 \div 2 = 0 \cdots 1$$

(2) 小数部分的转换。

那么,十进制的纯小数应如何转换为二进制数的表示形式呢? 设:

$$S' = K_{-1} 2^{-1} + K_{-2} 2^{-2} + \cdots + K_{-m} 2^{-m}$$

于是

$$2S' = K_{-1} + (K_{-2} 2^{-1} + \cdots + K_{-m} 2^{-m+1})$$

$$K_{-1} = 0 \text{ 或 } K_{-1} = 1;$$

令括号中的 $K_{-2} 2^{-1} + K_{-3} 2^{-2} + \cdots + K_{-m} 2^{-m+1} = S'_1$, 得到:

$$2S'_1 = K_{-2} + (K_{-3} 2^{-1} + K_{-4} 2^{-2} + \cdots + K_{-m} 2^{-m+2});$$

反复 m 次以后:

$$2S_m = K_m$$

于是得到一组数字:

$$K_1, K_2, \dots, K_m$$

$$K_i = 0 \quad \text{或} \quad K_i = 1, (i = 0, 1, \dots, m)$$

将这些数正向排列就能得到 $K_1, K_2, \dots, K_m$, 这便是所要求的二进制数。由上面的推导可以得到将十进制的纯小数转化为二进制的小数的规则。

规则: 十进制小数转换为二进制小数时, 将十进制小数乘以 2, 把积的整数部分记录下来, 再将积的小数部分继续乘以 2, 如此下去, 直到小数部分为零或二进制小数部分达到精度要求。这种方法叫做“乘基取整法”。

例如:

$$\begin{aligned} (0.8125)_{10} &\rightarrow (0.1101)_2 \\ 0.8125 \times 2 &= 1.625 \quad \dots\dots\dots 1 \\ 0.625 \times 2 &= 1.25 \quad \dots\dots\dots 1 \\ 0.25 \times 2 &= 0.5 \quad \dots\dots\dots 0 \\ 0.5 \times 2 &= 1.0 \quad \dots\dots\dots 1 \end{aligned}$$

又如:

$$\begin{aligned} (0.6)_{10} &\rightarrow (0.1001\dots\dots)_2 \\ 0.6 \times 2 &= 1.2 \quad \dots\dots\dots 1 \\ 0.2 \times 2 &= 0.4 \quad \dots\dots\dots 0 \\ 0.4 \times 2 &= 0.8 \quad \dots\dots\dots 0 \\ 0.8 \times 2 &= 1.6 \quad \dots\dots\dots 1 \\ 0.6 \times 2 &= 1.2 \quad \dots\dots\dots 1 \end{aligned}$$

有些十进制小数转换为二进制小数时, 可能无法用有限长的位数表示, 这时往往按照要求精确到小数点后若干位, 具体精确的位数应由实际需要或机器的字长决定。

(3) 整体的转换。

将整数与小数两部分的转换合起来, 即可得到整体的转换值。如将 $(326.8125)_{10}$ 转换为二进制数, 则:

$$(326)_{10} = (101000110)_2$$

$$(0.8125)_{10} = (0.1101)_2$$

故

$$\begin{aligned} (326.8125)_{10} &= (326)_{10} + (0.8125)_{10} = (101000110)_2 + (0.1101)_2 \\ &= (101000110.1101)_2 \end{aligned}$$

1.3 十六进制与其他进制的数

1. 采用十六进制的原因

至于计算机为什么采用十六进制, 如果你已理解计算机采用二进制的原因, 则此问题就迎刃而解了。认真分析一下二进制数和十六进制数, 你就会发现二进制数与十六进制数有太多的共同点, 甚至可以说十六进制数是二进制数的简写形式, 例如, 有这么一个二进制数:

$$011111011110010110100011110010001111$$

这密密麻麻的一长串 1 和 0, 对阅读与记忆是十分不便的。于是人们采用十六进制数来

缩短它,得如下数:

7DE5A3C8F

2. 其他进制的数转换为十进制数

与二进制数转换为十进制数的方法一样,八进制、十六进制的数都可以按照权展开的方法来转换为十进制数。

例如:

$$(2B30)_{16} = 2 \times 16^3 + 11 \times 16^2 + 3 \times 16^1 + 0 \times 16^0 = (11056)_{10}$$

$$(24)_{16} = 2 \times 16^1 + 4 \times 16^0 = (36)_{10}$$

$$(81)_8 = 8 \times 8^1 + 1 \times 8^0 = (65)_{10}$$

$$(134)_8 = 1 \times 8^2 + 3 \times 8^1 + 4 \times 8^0 = (92)_{10}$$

3. 十进制数转换为其他进制的数

与十进制数转换为二进制数相类似,将十进制数转换为十六进制数或八进制数时,也应分成整数部分和小数部分来实现。对于整数部分采用“除基倒取余法”;而对小数部分采用“乘基取整法”,只不过这里的基不是2,而是8或16。

例如:

$$(266)_{10} = (10A)_{16}$$

$$266 \div 16 = 16 \quad \dots\dots A$$

$$16 \div 16 = 1 \quad \dots\dots 0$$

$$1 \div 16 = 0 \quad \dots\dots 1$$

$$(72)_{10} = (110)_8$$

$$72 \div 8 = 9 \quad \dots\dots 0$$

$$9 \div 8 = 1 \quad \dots\dots 1$$

$$1 \div 8 = 0 \quad \dots\dots 1$$

1.4 数据单位、字符编码、汉字编码

1.4.1 计算机中的数据单位

计算机中用到的信息单位主要有位、字节、字等。

位(Bit)是计算机存储设备中的最小的信息容量单位,用0或1二进制数位来表示。如二进制数10011101是由8个位组成的,位常用b表示。

字节(Byte)是计算机的最小存储单元,常用B表示。微型机中由8个二进制位组成一个字节。如8位二进制数“10011101”构成一个字节。一个字节可存放一个半角英文字符的编码(如ASCII码)。两个字节可存放一个汉字编码。

一个字节表示的无符号整数,可以从最小的00000000至最大的11111111,共 2^8 个。习惯上, 2^{10} (即1024)个字节称为1K字节,记为1KB。随着存储容量的增大,还有下列计量单位,它们之间的关系如下:

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

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

$$2^{20}\text{B}=1024\times 1024\text{B}=1\text{MB}$$

$$2^{30}\text{B}=1024\times 1024\times 1024\text{B}=1\text{GB}$$

字(Word)是计算机信息交换、加工、存储的基本单元。通常将组成一个字的位数叫该字的字长,用来表示数据或信息的长度。如一台计算机的字长为 32 位,则表示该机的一个字由 4 个字节组成。不同级别的计算机的字长是不同的。

为什么不取整数 1000,而取一个如此难记的数字 1024 来表示 1 KB 字节呢?细心的读者一定不难发现: $2^{10}=1024$ 。正是由于计算机中采用的是二进制数,用 1000 来表示 1 千字节反而不方便了。

表示千字节的 KB、兆字节的 MB 以及千兆字节的 GB 可以简写成 K、M、G,即:1K=1KB,1M=1MB,1G=1GB。本书在以后的章节中将采用 KB、MB 与 GB 表示。

1.4.2 字符编码

1. 英文字符编码

在计算机中不仅是数字,所有的数据都是用二进制数来表示的。长期以来,存在各种字符编码,难于统一,为此美国国家标准局提出了一套编码方案,它叫做 ASCII 码(American Standard Code for Information Interchange,美国信息交换标准代码)。它收录了 128 个基本字符,其中包括了数字 0~9,英文大小写字母,一些运算符号如+、-、*、/和一些常用符号如\$、%、#等。每一个字符用一个八位二进制数来表示,如二进制的 01000001 表示英文大写字母 A;二进制的 00110001 表示数字字符 1 等等。为了便于记忆,常将这些字符编码以十进制形式表示。

表 1-3 中给出了常用字符与 ASCII 码对照。请注意在 ASCII 编码中所列的前 32 个编码所表示的字符都是计算机信息传送、加工过程中使用的一些控制字符,在屏幕上是看不出来的,打印机上也打印不出来。

2. 汉字编码

汉字是方块的,而且结构千变万化,要将它输入计算机并且表示出来,确实是一个难题。经过我国科研工作者几代的努力,这个问题已被解决。

人们习惯采用一种点阵方案来表示汉字。如图 1-1 所示的汉字“永”。图中白点的部分用“0”表示,黑点部分用“1”表示,这样就可对一个汉字字型用二进制数进行描述了。

1981 年,我国制定了“中华人民共和国国家标准信息交换汉字编码”,代号为 GB2312-80,这种编码称为国标码,是所有汉字编码都必须遵循的一个共同标准。GB2312-80 以 94 个可以显示的

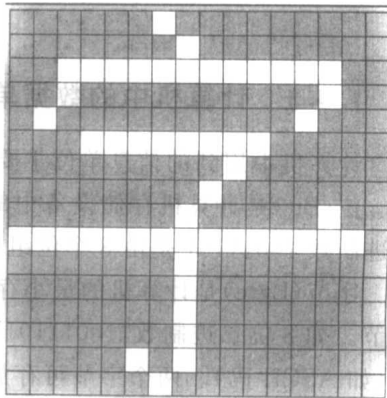


图 1-1 16×16 汉字点阵字形实例

ASCII 码作为基本集,共收录了汉字和图形符号 7 445 个,每个汉字用两个字节表示。汉字分为两级,一级汉字 3 755 个,按汉语拼音字母排列;二级汉字 3 008 个,按部首排列;非汉字字符 682 个。

表 1-3 ASCII 字符编码

ASCII 值 *	字符	控制字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符
000	(null)	NUL	032	(space)	064	@	096	'
001	○	SOH	033	!	065	A	097	a
002	●	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	\$	068	D	100	d
005	♣	END	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	*	071	G	103	g
008		BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	044	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014		SO	046	.	078	N	110	n
015		SI	047	/	079	O	111	o
016	►	DIE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018		DC2	050	2	082	R	114	r
019	!	DC3	051	3	083	S	115	s
020		DC4	052	4	084	T	116	t
021		NAK	053	5	085	U	117	u
022		SYN	054	6	085	V	118	v
023		ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	X	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028		FS	060	<	092	\	124	
029	◆	GS	061	=	093	]	125	}
030	▲	RS	062	>	094	^	126	~
031	▼	US	063	?	095	--	127	--

GB2312-80 规定,所有的国标汉字与符号组成一个 94×94 的矩阵。在此矩阵中,每行称为一个“区”,每列称为一个“位”,因此,这个矩阵实际上组成了一个有 94 个区(区号从 01 至 94),每个区内有 94 个位(位号从 01 至 94)的汉字字符集。一个汉字所在的区号与位号简单

地组合在一起就构成了该汉字一种外码——“区位码”，它用高低两个字节来表示，高字节表示汉字所在的区号，低字节表示汉字所在的位号。如汉字“啊”在 GB2312-80 中的位置是第 16 区的第 1 位，则它的区位码就是 1601，用十六进制数表示就是 1001H。汉字的区位码是唯一的。国标码与区位码之间存在如下换算关系：

国标码高字节 = 区码 + 20H 国标码低字节 = 位码 + 20H

例如，汉字“啊”的国标码用十六进制表示就是 3021H。

GB2312-80 编码的安排情况如下：

- (1) 1~9 区非汉字字符 682 个；
- (2) 10~15 区，空位 564 个；
- (3) 16~55 区，一级汉字，也称为常用字，按汉字的拼音排列；
- (4) 56~87 区，二级汉字，也称为次常用字，按汉字的部首排列；
- (5) 88~94 区，空位。

近年来，为便于和加强国际间信息交流，国家制定了新的汉字编码标准 GB-13000，国际上称为 ISO/IEC10646，这种汉字编码用 3 个字节表示一个汉字，汉字编码容量大大增加，最大的特点是包括了中、日、韩等许多国家的文字。

汉字编码分为内码和外码。内码是计算机系统存储、处理汉字信息时所用的代码。汉字的输入码要转换成内码才能在计算机内存储和处理，一个内码占两个字节。汉字国标码的高低字节的取值范围在 33~126 之间，每个字节最高位都是 0，正好和 ASCII 码相冲突，故不能作为机内码使用。国标码经过变换之后才能作为机内码使用，机内码与国标码之间的变换关系如下：

内码高字节 = 国标码高字节 + 80H 内码低字节 = 国标码低字节 + 80H

外码是指输入码及打印码、显示码等，用于人与计算机进行交互（汉字输入/输出）时所用的代码。就输入码来说，国内外有几百种编码方案，常用的有区位码、拼音码、五笔字形码、自然码等。

汉字是一种像形文字，每一个汉字可以看成是一个特定的图形，这种图形一般用点阵信息来描述。所有汉字的点阵信息按国标码规定的先后顺序组合在一起，就形成了汉字的字库。

例如，如果用 16×16 点阵来表示一个汉字，则一个汉字占 16 行，每一行上有 16 个点，每一个点用一个二进制位来表示，0 表示无笔画，1 表示有笔画，形成字形。每一行的 16 个点用两个字节来存放，共 16 行，那么，一个 16×16 点阵的汉字需要 32 个字节来存放。同样的道理，24×24 点阵的一个汉字需要 72 字节来存放。

在需要输出一个汉字时，例如显示或打印，首先根据该汉字的国标码或内码找出其点阵信息在汉字库中的位置，然后取出该汉字的点阵信息作为图形在屏幕显示或在打印机上输出。