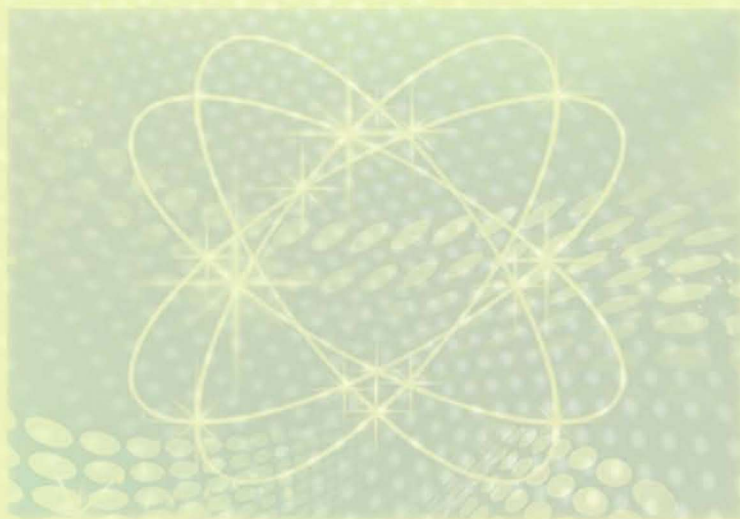


大学计算机基础 实践教程

主 编 何友鸣 李 亮 金大卫



华中科技大学出版社

大学计算机基础 实践教程

主 编 何友鸣 李 亮 金大卫
副主编 童旺宇 宋 洁 张永进

华中科技大学出版社
中国·武汉

内 容 提 要

本书是《大学计算机基础》教材的同步辅导书,由八章组成,包括计算机文化基础、Windows 7、Word 2010、Excel 2010 和 PowerPoint 2010,还有计算机网络基础及应用、多媒体技术基础和信息安全等知识内容。每章介绍的内容大致包括本章主要内容、阅读资料(有的章节没有)、习题解答、实验指导等几个部分,全面地对学习内容进行辅导和指导。

本书可作为大专院校各层次非计算机专业学生的辅助教材,也可以作为相应层次的成人教育、职业教育的辅助教材,亦可供计算机知识学习者、爱好者或 IT 行业工程技术人员等学习参考。对于从事计算机学科教育的教师,本书也是一本很好的参考书。

图书在版编目(CIP)数据

大学计算机基础实践教程/何友鸣,李亮,金大卫主编. —武汉:华中科技大学出版社,2014.8
ISBN 978-7-5680-0327-8

I. ①大… II. ①何… ②李… ③金… III. ①电子计算机-高等学校-教材 IV. ①TP3

中国版本图书馆 CIP 数据核字(2014)第 183244 号

大学计算机基础实践教程

何友鸣 李亮 金大卫 主编

策划编辑:曾 光

责任编辑:史永霞

封面设计:龙文装帧

责任校对:马燕红

责任监印:张正林

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)81321915

录 排:华中科技大学惠友文印中心

印 刷:华中理工大学印刷厂

开 本:787mm×1092mm 1/16

印 张:11.5

字 数:289 千字

版 次:2014 年 9 月第 1 版第 1 次印刷

定 价:28.00 元



本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

前言

PREFACE

本书是《大学计算机基础》的同步辅导书，在结构上与主教材保持一致，由八章组成；每章在内容上与主教材同步辅助，包括每章的主要内容、阅读资料（有的章节没有）、习题解答、实验指导等部分，有效地对教材内容进行辅导和指导。

“大学计算机基础”是一门实践性很强的课程，要求学生不仅掌握计算机的基础知识与理论，而且在计算机的操作上要达到一定的熟练程度，能够运用计算机解决日常工作中的问题，主要是办公事务的处理。按照教学大纲的要求，为了加强实验教学，提高学生的实际动手能力，我们编写了这本《大学计算机基础实践教程》，力求内容新颖、概念准确、通俗易懂、实用性强，在风格上与主教材完全统一。

本书由何友鸣、李亮、金大卫担任主编，童旺宇、宋洁、张永进担任副主编。参加本书编写工作的还有方辉云、胡仁、刘胜艳、刘阳、韩杰、冯浩、王静、鲁星、何苗、甘霞、徐冬、赵清强、肖莹慧、庄超，以及肖加清、强静仁、马建新、王中婧、郭小清、李俊等。首先要感谢的是中南财经政法大学信息与安全工程学院刘腾红教授、王少波老师和范爱萍老师，以及叶焕卓、阮新新、屈振新、周晓华、丁亚兰、李玲、鲁敏等老师，正是由于这些老师的启发，本书才得以在较短的时间内面世。本书的编写和出版，得到了中南财经政法大学武汉学院的领导和信息系教职工们的大力支持，特别是武汉学院教务处领导和同仁们的支持，在此深表感谢！

本书也注意到高职高专计算机信息技术教材的特点，在编写过程中体现了这一方面的要求，尽力使得教学体系更加完备，有利于提高学生的实际动手能力。

由于水平所限，书中错误和不足之处在所难免，恳请读者提出宝贵意见。最后，由衷地感谢那些支持和帮助我们所有朋友们！谢谢你们使用和关心本书，并预祝你们教学或学习成功！

编 者

2014年夏日于中南财经政法大学武汉学院

目
录

CONTENTS

第 1 章	计算机文化基础	1
1.1	本章主要内容	1
1.2	计算机概论辅导	1
1.3	信息表示辅导	4
1.4	文件系统简介	11
1.5	常用操作系统简介	14
1.6	习题解答	17
1.7	实验指导	22
第 2 章	Windows 7	30
2.1	本章主要内容	30
2.2	习题解答	30
2.3	实验指导	32
第 3 章	Word 2010	48
3.1	本章主要内容	48
3.2	习题解答	48
3.3	实验指导	53
3.4	一个实用技术——邮件合并与打印	61
第 4 章	Excel 2010	64
4.1	本章主要内容	64
4.2	习题解答	64
4.3	实验指导	68
第 5 章	PowerPoint 2010	85
5.1	本章主要内容	85
5.2	习题解答	85
5.3	实验指导	92
第 6 章	计算机网络基础及应用	102
6.1	本章主要内容	102
6.2	计算机网络	102

6.3	计算机网络体系结构	104
6.4	数据通信基础	107
6.5	习题解答	111
6.6	实验指导	115
第7章	多媒体技术基础	125
7.1	本章主要内容	125
7.2	多媒体技术	125
7.3	超文本与超媒体简介	135
7.4	虚拟现实技术简介	136
7.5	流媒体技术简介	138
7.6	习题解答	140
7.7	实验指导	143
第8章	信息安全	162
8.1	本章主要内容	162
8.2	习题解答	162
8.3	实验指导	167
参考文献		176

第1章 计算机文化基础



1.1 本章主要内容

本章为计算机文化基础知识，主要对计算机的发展、特点、应用和分类做了简单的概述，并重点介绍了从 1946 年第一台电子数字积分计算机 ENIAC 诞生起，至今已有近 60 年的历史，已经经历了四代。

本章还介绍计算机中的信息表示方法，包括数字的表示方法、数值的表示方法、字符的表示方法、声音的表示方法和图形图像的表示方法等。计算机采用二进制，这就涉及二进制编码问题。

本章最后介绍计算机系统。一个完整的计算机系统是由硬件系统和软件系统两大部分组成的。硬件(hardware)也称硬设备，是计算机系统的物质基础。软件(software)，从广义上来说，是计算机中运行的所有程序以及各种文档资料的总称。硬件和软件相结合才能充分发挥电子计算机的系统功能，两者缺一不可。



1.2 计算机概论辅导

1.2.1 世界第一台计算机

1. ENIAC

现代科学技术的发展及信息在社会中的重要地位，导致了计算工具的创新。1946 年 2 月世界上第一台电子数值计算机 ENIAC 在美国宾夕法尼亚大学诞生，它标志着科学技术的发展进入新的时代——电子计算机时代。从第一台电子计算机的诞生到现在，计算机的发展已经历四代，并正在向第五代发展。

1936 年英国数学家阿兰·图灵(Alan Turing)提出了计算机理论模型：只要能够被分解为有限步骤，就能够实现自动计算。这就是图灵机。如后面实现的 ABC 计算机(Atanasoff Berry computer)、ENIAC(electronic numerical integrators and calculation)，都是图灵机的代表。特别是 ENIAC，是世界上第一台可以真正运算、全部是电子装置的计算机。它在计算机的发展史上具有里程碑式的意义。

图 1-1 所示是关于 ENIAC 计算机的照片。

2. 计算机之父——冯·诺依曼

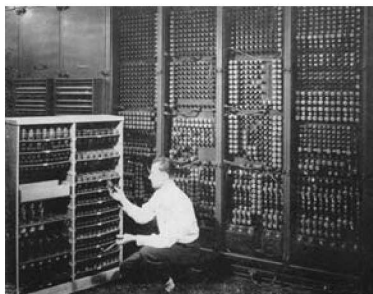
冯·诺依曼(von Neumann, 1903—1957)，美籍匈牙利科学家，是计算机科学的创始人之一。他不仅在计算机方面，而且在数学、逻辑、物理等领域都作出了巨大的贡献，他的杰出成就使他成为科学上的巨人，被誉为“计算机之父”。



(a)



(b)



(c)



(d)

图 1-1 世界上第一台可以真正运算全部是电子装置的计算机 ENIAC

1.2.2 电子器材

用电子管组成的计算机在 1952 年美国大选中预测艾森豪威尔获胜——预测结果和实际统计结果完全相同；1957 年 IBM 公司生产的第一台商用计算机 IBM701，一共生产了 19 台。但是电子管体积大，故障率高。图 1-2 所示为电子管。

晶体管比电子管体积小，稳定性高，如图 1-3 所示。1948 年 6 月，贝尔实验室研制成功了世界上第一只晶体管。第一台晶体管计算机是 CDC 制造的 1604。这时开始使用高级语言，开始通过电话线进行数据交流，虽然速度很慢，但这已经是网络的萌芽。并行处理被所有大型计算机和超级计算机所使用。麻省理工学院在这期间提出“多道程序”方案。



图 1-2 电子管

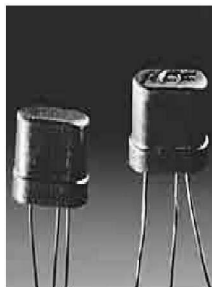


图 1-3 晶体管

集成电路(integrated circuits, IC)于 1958 年发明，如图 1-4 所示。接着就有大规模集成电路(LSIC)和超大规模集成电路，如图 1-5 所示。这时的缩微技术高度发展。

摩尔博士预言，IC 上能被集成的晶体管数目将会以每 18 个月翻一番的速度稳定增长(摩尔法则)。



图 1-4 集成电路

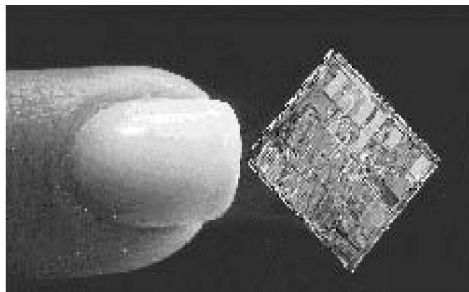


图 1-5 大规模和超大规模集成电路

这个阶段有代表性的计算机是 IBM 推出的著名的 360 系列计算机，如图 1-6 所示。这时的计算机发展到不再捆绑销售它的语言软件，开创了计算机语言市场，最终使软件成为一个巨大的产业。



图 1-6 著名的 IBM 360 计算机

大规模集成和超大规模集成电路的出现，导致使用大规模集成电路的标志着第四代计算机出现的处理器 Intel 系列处理器的出现，从而导致在 1976—1977 年，第一台真正意义上的微机 Apple I 的实现，它有显示器、键盘、软盘和操作系统软件，如图 1-7 所示。

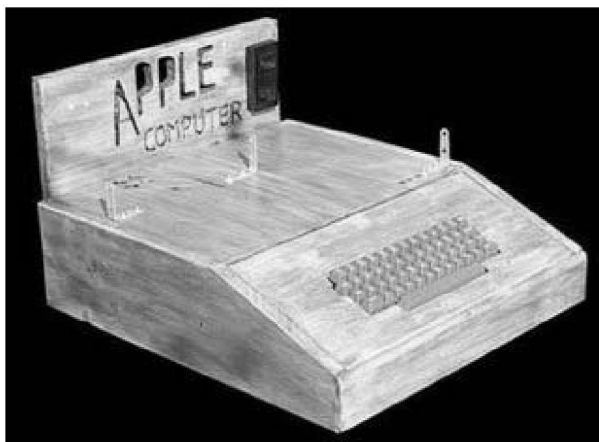


图 1-7 微机 Apple I

1980 年, IBM 公司做出了两个决定: 一是选择 Intel 8088 芯片作为它的微机处理器——PC(personal computer); 二是委托 Microsoft 公司设计操作系统。

IBM 公司的这两个决定带来了历史性的巨大影响:

IBM 公司商标的 PC 成为微型计算机的同义词;

Microsoft 公司和 Intel 公司则在计算机软件和硬件方面成为和 IBM 公司分庭抗礼的业界巨头。

图 1-8 所示即为微型计算机 IBM-PC。

电子缩微技术的高度发展, 促使了掌上电脑的出现, 如图 1-9 所示。



图 1-8 IBM-PC



图 1-9 掌上电脑



1.3 信息表示辅导

1.3.1 计算机为什么要采用二进制

尽管计算机可以处理各种数据和信息, 但是计算机内部使用的是二进制数。二进制并不符合人们的习惯, 计算机中采用二进制主要原因如下。

1. 物理实现简单

计算机采用物理元件的状态来表示计数制中各位的值和位权, 绝大多数物理元件都只有两种状态; 如果计算机中采用十进制, 势必要求计算机有能够识别 0~9 共 10 种状态的装置; 在实际工作中, 是很难找到能表示 10 种不同稳定状态的电子器件的。虽然可以用电子线路的组合来表示, 但是线路非常复杂, 所需的设备量大, 而且十分不可靠。而二进制中只有 0 和 1 两种数字符号, 可以用电子器件的开关两种不同状态来表示一位二进制数。例如, 可以用晶体管的截止和导通表示 1 和 0, 或者用电平的高和低表示 1 和 0 等。所以, 在数字系统中普遍采用二进制。

2. 运算规则简单

二进制数只有 0 和 1 两个数字符号, 因此运算规则比十进制简单得多。二进制的加减乘除运算规则分别如表 1-1 所示。

表 1-1 二进制的加减乘除运算规则

加法	$0+0=0$	$0+1=1$	$1+0=1$	$1+1=10$ (进位)
减法	$0-0=0$	$0-1=1$ (借位)	$1-0=1$	$1-1=0$
乘法	$0\times 0=0$	$0\times 1=0$	$1\times 0=0$	$1\times 1=1$
除法	$0\div 0=0$ (无意义)	$0\div 1=0$	$1\div 0=0$ (无意义)	$1\div 1=1$

3. 适合逻辑运算

逻辑是指条件与结论之间的关系。因此,逻辑运算是对于因果关系进行分析的一种运算,运算结果并不表示数值大小,而是表示逻辑概念,即成立还是不成立。计算机的逻辑关系是一种二值逻辑,二值逻辑可以用二进制的1或0来表示,例如,1表示“成立”“是”或“真”,0表示“不成立”“否”或“假”等。若干位二进制数组成逻辑数据,位与位之间无“权”的内在联系。对两个逻辑数据进行运算时,每位之间相互独立,运算是按位进行的,不存在算术运算中的进位和借位,运算结果仍是逻辑数据。

逻辑运算主要包括三种基本运算:逻辑加法(又称“或”运算)、逻辑乘法(又称“与”运算)和逻辑否定(又称“非”运算)。此外,“异或”运算也很有用。

1) 逻辑加法(“或”运算)

逻辑加法通常用符号“+”或“ $\vee$ ”来表示。逻辑加法运算规则如下:

$$0+0=0, \quad 0\vee 0=0$$

$$0+1=1, \quad 0\vee 1=1$$

$$1+0=1, \quad 1\vee 0=1$$

$$1+1=1, \quad 1\vee 1=1$$

逻辑加法有“或”的意义,也就是说,在给定的逻辑变量中,只要有一个为1,其逻辑加的结果为1;两者都为1则逻辑加为1。

2) 逻辑乘法(“与”运算)

逻辑乘法通常用符号“ $\times$ ”或“ $\wedge$ ”或“ $\cdot$ ”来表示。逻辑乘法运算规则如下:

$$0\times 0=0, \quad 0\wedge 0=0, \quad 0\cdot 0=0$$

$$0\times 1=0, \quad 0\wedge 1=0, \quad 0\cdot 1=0$$

$$1\times 0=0, \quad 1\wedge 0=0, \quad 1\cdot 0=0$$

$$1\times 1=1, \quad 1\wedge 1=1, \quad 1\cdot 1=1$$

不难看出,逻辑乘法有“与”的意义,它表示只当参与运算的逻辑变量都同时取值为1时,其逻辑乘积才等于1。

3) 逻辑否定(“非”运算)

逻辑非运算又称逻辑否运算。其运算规则为:

$$\bar{0}=1 \quad \text{非 } 0 \text{ 等于 } 1$$

$$\bar{1}=0 \quad \text{非 } 1 \text{ 等于 } 0$$

4) 异或逻辑运算(半加运算)

异或逻辑运算通常用符号“ $\oplus$ ”表示,其运算规则为:

$$0\oplus 0=0, \quad 0 \text{ 同 } 0 \text{ 异或, 结果为 } 0$$

$0 \oplus 1 = 1$, 0 同 1 异或, 结果为 1

$1 \oplus 0 = 1$, 1 同 0 异或, 结果为 1

$1 \oplus 1 = 0$, 1 同 1 异或, 结果为 0

两个逻辑变量相异, 输出才为 1。

二进制的主要缺点是数位太长, 不便于阅读和书写, 人们也不习惯使用。因此, 常用八进制和十六进制作为二进制的缩写方式。另外, 为了适应人们的习惯, 通常在计算机内部都采用二进制数的同时, 输入和输出环节则采用十进制数, 由计算机自己完成二进制与十进制之间的相互转换。

1.3.2 存储单位与机器数

1. 存储单位

机器一次能表示的二进制数的位数叫机器的字长, 一台机器的字长是固定的。8 位长度的二进制数称为一个字节(byte), 机器字长一般都是字节的整数倍, 如字长 8 位、16 位、32 位、64 位。

我们要处理的信息在计算机中常常被称为数据。所谓数据, 是可以由人工或自动化手段加以处理的那些事实、概念、场景和指示的表示形式, 包括字符、符号、表格、声音和图形等。数据可在物理介质上记录或传输, 并通过外围设备被计算机接收, 经过处理而得到结果, 计算机对数据进行解释并赋予一定意义后, 便成为人们所能接受的信息。计算机中数据的常用单位有位、字节和字。

计算机存储信息的最小单位是进制的一个数位, 简称为位(bit), 音译比特, 二进制的“0”或一个“1”叫一位。一个二进制位可以表示两种状态(0 或 1), 两个二进制位可以表示四种状态(00、01、10、11)。显然, 位越多, 所表示的状态就越多。

计算机存储容量的基本单位是字节(byte), 音译为拜特, 8 个二进制位组成 1 个字节(1 byte = 8 bit)。计算机存储容量大小以字节数来度量。例如, 计算机内存的存储容量、磁盘的存储容量等都是以字节为单位进行表示的。

除了用字节为单位表示存储容量外, 还可以用千字节(kilobyte, KB)、兆字节(megabyte, MB)以及吉字节(gigabyte, GB)、太字节(terabyte, TB)等表示存储容量。它们之间存在下列换算关系:

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

$$1\text{MB} = 2^{10}\text{KB} = 2^{20}\text{B} = 1\,048\,576\text{B}$$

$$1\text{GB} = 2^{10}\text{MB} = 2^{30}\text{B} = 1\,073\,741\,824\text{B}$$

一个标准英文字母占一个字节位置, 一个标准汉字占两个字节位置。

字和计算机中字长的概念有关。字长是指计算机在进行处理时一次作为一个整体进行处理的二进制数的位数, 具有这一长度的二进制数则被称为该计算机中的一个字。字通常取字节的整数倍, 是计算机进行数据存储和处理的运算单位。

计算机按照字长进行分类, 可以分为 8 位机、16 位机、32 位机和 64 位机等。字长越长, 计算机所表示数的范围就越大, 处理能力也越强, 运算精度也就越高。在不同字长的计算机中, 字的长度也不相同。例如, 在 8 位机中, 一个字含有 8 个二进制位, 而在 64 位机中, 一个字则含有 64 个二进制位。

2. 机器数

机器数的特点是：

- (1) 符号数值化，0 代表正，1 代表负，通常将表示符号的代码放在数据的最高位；
- (2) 小数点是隐藏的，不占用存储空间；
- (3) 每个机器数所占据的二进制位数受机器硬件条件的限制，与机器字长有关，超过机器字长的数值要舍去；
- (4) 因为机器数的长度是由机器的硬件规模规定的，所以机器数表示的数值是不连续的。

1.3.3 原码、反码和补码

1. 概念

显然，计算机是对机器数进行运算的，所以机器数的运算规则要简单些。对于用 0、1 表示“+”“-”号的机器数，其符号虽然为数，但是仅仅有符号的含义，因此进行加减乘除等运算时，符号位的数不能像数值位的数一样运算，而要单独处理。

为了便于数值的运算和处理，对机器数定义了不同的表示方法，其中包括了数的原码、反码和补码。在实际计算机中广泛采用补码来进行运算，主要是因为补码的运算规则较原码和补码简单。

2. 原码表示法

原码表示法是机器数的一种简单的表示法。其符号位用 0 表示正号，用 1 表示负号，数值一般用二进制形式表示。设有一数为 X ，则 X 的原码可记作 $[X]_{\text{原}}$ 。

例如， $X_1 = +1010110$

$X_2 = -1001010$

其原码记作：

$[X_1]_{\text{原}} = [+1010110]_{\text{原}} = 01010110$

$[X_2]_{\text{原}} = [-1001010]_{\text{原}} = 11001010$

原码表示数的范围与二进制位数有关。当用 8 位二进制来表示小数原码时，其表示范围：

最大值为 0.1111111 ，其真值约为 $(0.99)_{10}$ ；

最小值为 1.1111111 ，其真值约为 $(-0.99)_{10}$ 。

当用 8 位二进制来表示整数原码时，其表示范围：

最大值为 01111111 ，其真值为 $(127)_{10}$ ；

最小值为 11111111 ，其真值为 $(-127)_{10}$ 。

在原码表示法中，对 0 有两种表示形式：

$[+0]_{\text{原}} = 00000000$

$[-0]_{\text{原}} = 10000000$

3. 反码表示法

机器数的反码可由原码得到。如果机器数是正数，则该机器数的反码与原码一样；如果机器数是负数，则该机器数的反码是对它的原码各位(符号位除外)取反而得到的。设有一数 X ，则 X 的反码记作 $[X]_{\text{反}}$ 。

例如: $X1=+1010110$

$X2=-1001010$

则 $[X1]_{\text{原}} = 01010110$

$[X1]_{\text{反}} = [X1]_{\text{原}} = 01010110$

$[X2]_{\text{原}} = 11001010$

$[X2]_{\text{反}} = 10110101$

反码通常作为求补过程的中间形式。

4. 补码表示法

机器数的补码可由原码得到。如果机器数是正数,则该机器数的补码与原码一样;如果机器数是负数,则该机器数的补码是对它的原码各位(符号位除外)取反,并在末位加 1 而得到的。设有一数 X ,则 X 的补码记作 $[X]_{\text{补}}$ 。

例如, $[X1]=+1010110$

$[X2]=-1001010$

$[X1]_{\text{原}} = 01010110$

$[X1]_{\text{补}} = 01010110$

即 $[X1]_{\text{原}} = [X1]_{\text{补}} = 01010110$

$[X2]_{\text{原}} = 11001010$

$[X2]_{\text{补}} = 10110101 + 1 = 10110110$

补码表示数的范围与二进制位数有关。当采用 8 位二进制表示时,小数补码的表示范围:

最大值为 0.1111111 , 其真值为 $(0.99)_{10}$;

最小值为 1.0000000 , 其真值为 $(-1)_{10}$ 。

采用 8 位二进制表示时,整数补码的表示范围:

最大值为 01111111 , 其真值为 $(127)_{10}$;

最小值为 10000000 , 其真值为 $(-128)_{10}$ 。

在补码表示法中,0 只有一种表示形式:

$[+0]_{\text{补}} = 00000000$

$[-0]_{\text{补}} = 11111111 + 1 = 00000000$ (由于受设备字长的限制,最后的进位丢失)

所以有 $[+0]_{\text{补}} = [-0]_{\text{补}} = 00000000$, 即补码解决了原码中存在 +0 和 -0 的问题。

【例 1-1】 已知 $[X]_{\text{原}} = 10011010$, 求 $[X]_{\text{补}}$ 。

【分析】 由 $[X]_{\text{原}}$ 求 $[X]_{\text{补}}$ 的原则是:若机器数为正数,则 $[X]_{\text{原}} = [X]_{\text{补}}$;若机器数为负数,则该机器数的补码可对它的原码所有位(符号位除外)取反,再在末位加 1 而得到。现给定的机器数为负数,故有 $[X]_{\text{补}} = [X]_{\text{反}} + 1$, 即:

$[X]_{\text{原}} = 10011010$

$[X]_{\text{反}} = 11100101$

$[X]_{\text{补}} = 11100110$

【例 1-2】 已知 $[X]_{\text{补}} = 11100110$, 求 $[X]_{\text{原}}$ 。

【分析】 对于机器数为正数,则 $[X]_{\text{原}} = [X]_{\text{补}}$;对于机器数为负数,则有 $[X]_{\text{原}} = [[X]_{\text{补}}]_{\text{补}}$ 。

现给定的数为负数,故有:

$$[X]_{\text{补}} = 11100110$$

$$[[X]_{\text{补}}]_{\text{反}} = 10011001$$

$$[[X]_{\text{补}}]_{\text{补}} = 10011010 = [X]_{\text{原}}$$

5. 作用

原码、反码和补码表示的均是机器数，但是各有特点，下面通过实际例子来说明补码较原码和反码的加减运算要简单。

【例 1-3】 已知 $x = +1101, y = +0110$ ，用原码、反码和补码计算 $x - y$ 的值。

(1) 原码运算。

计算机采用原码运算时，需将真值表示为原码：

$$[x]_{\text{原}} = 0, 1101 \quad [y]_{\text{原}} = 0, 0110$$

原码运算的方法与手算相似，先要判别相减的两数是同号还是异号。若为同号，则进行减法；若为异号，则进行加法。其次判别 x 、 y 两数的大小，确定用大数减小数。

在本例中， x 、 y 同号，且 $|x| > |y|$ ，所以有如下算式：

$$\begin{array}{r} [x]_{\text{原}} = 0, 1101 \\ - [y]_{\text{原}} = 0, 0110 \\ \hline [x-y]_{\text{原}} = 0, 0111 \end{array}$$

求得 $x - y = +0111$ 。

(2) 反码运算。

计算机采用反码运算时，需将真值表示为反码：

$$[x]_{\text{反}} = 0, 1101 \quad [y]_{\text{反}} = 0, 0110$$

由于反码加减运算有如下规则：

$$[x+y]_{\text{反}} = [x]_{\text{反}} + [y]_{\text{反}} + \text{符号位进位}$$

所以要求 $x - y$ 时，可以做变换 $x - y = x + (-y)$ ，因此先求 $(-y)$ 的反码：

$$[-y]_{\text{反}} = 1, 1001$$

再进行 $[x]_{\text{反}}$ 和 $[-y]_{\text{反}}$ 的加法运算：

$$\begin{array}{r} [x]_{\text{反}} = 0, 1101 \\ + [-y]_{\text{反}} = 1, 1001 \\ \hline 10, 0110 \\ + \quad \quad \quad 1 \quad \text{加上一行的循环进位} \\ \hline [x-y]_{\text{反}} = 0, 0111 \end{array}$$

即求得 $x - y = +0111$ 。

(3) 补码运算。

计算机采用补码运算时，需将真值表示为补码：

$$[x]_{\text{补}} = 0, 1101 \quad [y]_{\text{补}} = 0, 0110$$

由于补码加减运算规则为：

$$[x+y]_{\text{补}} = [x]_{\text{补}} + [y]_{\text{补}}$$

$$[x-y]_{\text{补}} = [x+(-y)]_{\text{补}} = [x]_{\text{补}} + [-y]_{\text{补}}$$

根据补码的编码规则有 $[-y]_{\text{补}} = 1, 1010$ ，此时的加法运算如下：

$$\begin{array}{r} [x]_{\text{补}} = 0, 1101 \\ + \quad [-y]_{\text{补}} = 1, 1010 \\ \hline 10, 0111 \end{array}$$

将最开始的进位 1 丢弃, 所以求得 $x-y=+0111$ 。

由例 1-3 可以看出, 补码的加减运算最简单, 反码的次之(因为要加上循环进位), 原码最复杂。所以, 计算机中普遍采用补码。

1.3.4 汉字输入方法

计算机汉字输入的方法很多, 可以归纳为三类。

1. 键盘输入法

目前的键盘输入法种类繁多, 但常用的有以下两类。

(1) 拼音输入法: 利用汉语拼音直接输入汉字的输入法。几乎不用花费很多时间学习, 只要汉语拼音掌握得好, 就可以使用。但汉字的同音字比较多, 重码率很高, 输入速度很慢。为了改进拼音输入法的重码, 提高输入速度, 不少拼音输入法做了改进, 提高了智能化程度, 提高了输入效率。如搜狗拼音输入法、谷歌拼音输入法、智能 ABC、紫光拼音输入法和微软拼音输入法, 改进后的拼音输入法每分钟可以输入几十个汉字甚至上百个汉字。

拼音输入法不适于专业的打字员, 但非常适合普通的计算机操作者, 如果用户拼音基础比较好, 不妨选用一种适合自己的拼音输入法。

(2) 笔画输入法: 笔画或笔画组合(称为字根或部件)对应计算机键盘的字母或数字键进行文字输入。

在全国盛极一时的这类输入法可能就是五笔了, 如王码五笔、万能五笔、极品五笔等。好的形码输入法具有重码少、输入效率高的特点, 但输入规则比较繁复, 键盘与字根(部件)的对应关系比较复杂, 学习起来比较困难, 而且容易遗忘。

笔画输入法适合于专职打字人员, 用户熟练了可以实现盲打, 输入速度较快, 但需要花费较多的时间去掌握和熟练。

2. 手写输入法

手写输入法: 一种笔式环境下的手写中文识别输入法。

手写输入法符合中国人用笔写字的习惯, 需要配套的手写板。在配套的手写板上用笔(可以是任何类型的硬笔)来书写汉字, 计算机就能将其识别并显示出来, 不仅方便、快捷, 而且错字率也比较低。如汉王笔、紫光笔、文通笔等, 可以像在信纸上写字一样, 采用平常写字的方式, 轻松自然地往计算机里输入文字。但早期的手写输入设备因辨识时间较长会拖慢输入速度。为了缩短辨识时间, 提高输入速度, 不少手写输入设备做了改进, 增加了智能化程度, 提高了输入效率。如汉王笔, 可全屏幕重叠书写, 识别速度在每秒 12 个汉字以上。先进的手写输入设备还具有功能强大的图像处理等功能, 可以自如地上面绘画。

手写输入法适合于普通的计算机操作者及有绘画图像特殊要求的使用者, 输入者不必经过培训, 只要会写字, 就可无师自通地轻松自然地进行汉字输入了。

3. 语音输入法

语音输入法: 将声音通过话筒转换成文字的一种输入方法。

语音输入法必须采用麦克风, 用户可以用正确的读音将文件输入计算机。在硬件方面, 要求计算机必须配备能进行正常录音的声卡, 调试好麦克风, 用户就可以对着麦克风用普通话进行文字录入了。即便是用户的普通话不标准, 只要用语音输入软件提供的语音训练程序, 进行一段时间的训练, 让软件熟悉用户的口音, 也同样可以通过讲话来实现文字输

入。语音识别系统以 IBM 推出的 ViaVoice 为代表,国内的语音识别系统有 Dutty++语音识别系统、天信语音识别系统、世音语音识别系统等。虽然使用起来很方便,但错字率仍然比较高,特别是一些未经训练的专业名词及生僻字。

语音输入法适合于不便使用键盘鼠标操作计算机的人及要求降低文字输入工作量的计算机操作者。

可以这样说,到目前为止,还没有哪一种中文输入法是没有缺点的,用户只能够根据自己的实际情况,选择一种适合于自己的中文输入法。

根据计算机软件的发展趋势,最终将会是拼音输入法、手写输入法和语音输入法的天下,而且输入的单位不再是字,而是词和句。但总体来说,现在我们还处于一个比较落后的水平。



1.4 文件系统简介

1.4.1 文件的概念

计算机中的程序和数据需要存储,然后再送到 CPU 中执行。信息可以保存在计算机的内存中,也可以保存在外存中。但保存在内存中的信息在计算机关机后将全部丢失,在实际应用中,大量的信息必须保存在外存中。存放在外存中相关数据的集合就是文件,或者说,文件是存放在外存中的程序、数据、符号等信息的集合。

具体的文件可以表现为一个用户程序、一个软件系统、一篇文稿、一张表格、某某单位的财务数据,等等。即使是 CD 或 VCD 等图像、声音,也是以文件的形式存放的。

1. 文件的特征

- (1) 文件是能长期保存在外存中的信息。
- (2) 保存文件时,要给文件命名,即文件名。
- (3) 即使文件同名,文件也能被唯一地确定。通常是以文件存放的位置来确定的。

2. 文件的命名及类别

文件是以文件名和文件存放的位置来确定的。用户使用文件时通过文件名和文件存放的位置等信息,来确定文件并读取文件至内存中。这种方式如同实际中给人起名一样,中国人的名字一般是姓在先,名在后,不同的家庭中可以有同名的人,但同一家庭中不可以出现同名的人。

操作系统中对文件的命名有一定的规定。文件的名称可由两个部分组成:文件名和扩展名。文件名与扩展名之间以小数点“.”连接。文件名通常表示一个文件的名称,扩展名往往表示一个文件的属性。例如,声音文件的扩展名通常是.wav(wave),而文档文件的扩展名通常是.doc(document),BASIC 语言书写的程序的扩展名是.bas。

操作系统中文件的命名格式:

<文件名>[.<扩展名>]

文件名是某个文件的名称,DOS 规定由 1~8 个字符(或者 1~4 个汉字)组成;在 Windows 中,文件名允许达到 255 个字符。文件名在文件命名中必须给出。扩展名由 1~3 个字符组成,扩展名在文件命名时可以给出,也可以没有。换句话说,文件名是必须有的,而扩展名是可有可无的。