

高等学校计算机基础教育
改革与实践系列教材

C程序设计

主 编 赵山林
副主编 高 媛



高等教育出版社
HIGHER EDUCATION PRESS

高等学校计算机基础教育改革与实践系列教材

C 程序设计

C Chengxu Sheji

主 编 赵山林

副主编 高 媛



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书以“零基础”为起点,首先阐述了学习 C 语言必备的计算机和程序设计的基础知识,然后以 C 语言语法和句法的规则、程序设计方法和计算机思维的培养为重点,以应用为主线,由浅入深地阐述了 C 语言的数据类型、结构化、模块化等知识点,归纳了选择、循环结构算法的设计思路及函数设计方法,配以大量的实例详细阐述了解决实际问题的程序设计方法。

本书可以作为高等学校计算机及相关专业学生学习 C 语言程序设计的教材,也可作为参加有关考试和自学的参考书。

图书在版编目(CIP)数据

C 程序设计/赵山林主编. —北京:高等教育出版社,2010. 8

ISBN 978 - 7 - 04 - 030153 - 3

I. ①C… II. ①赵… III. ①C 语言 - 程序设计 - 高等学校 - 教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 131219 号

策划编辑	饶卉萍	责任编辑	柳秀丽	封面设计	张志奇	责任绘图	尹 莉
版式设计	余 杨	责任校对	刘 莉	责任印制	朱学忠		

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100120

经 销 蓝色畅想图书发行有限公司
印 刷 肥城新华印刷有限公司

开 本 787 × 1092 1/16
印 张 22.75
字 数 550 000

购书热线 010 - 58581118
咨询电话 400 - 810 - 0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landracom.com>
<http://www.landracom.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2010 年 8 月第 1 版
印 次 2010 年 8 月第 1 次印刷
定 价 29.80 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 30153 - 00

高等学校计算机基础教育改革与实践系列教材 编审委员会

主 任：陈立潮

副 主 任：段 富

委 员：亢临生 赵山林 贾 伟 李祥生

王建国 朱红康 张 森 张英俊

序 言

教育部高等学校计算机基础课程教学指导委员会在《高等学校计算机基础教学发展战略研究报告暨计算机基础课程教学基本要求》(以下简称《基本要求》)中指出,党的十七大提出了我国要从改造传统工业入手,走工业与信息技术相融合的新型工业化道路。为此,需要培养大批新一代“专业+信息”的工程技术人才。作为信息技术的核心,计算机基础教育的重要性被提到了空前的高度,计算机基础课程在高等学校确立了公共基础课的地位。在实施高等学校本科教学质量与教学改革的过程中,计算机基础的教学改革朝着高水平、应用化、规范化方向推进。

在完成教育部高等学校计算机基础课程教学指导委员会课题的过程中,我们组织了十余所高等学校计算机基础教学的负责人和一线教师,对这些高等学校中几千名本科生以问卷的形式对计算机基础教育现状进行了抽样调查,内容涉及了大学生起始计算机技能基本情况、目前课程安排和课程效果评价3个方面,着重了解了当代大学生对计算机基础教育的新需求及对计算机基础课程的意见与建议等。通过对这些调查问卷进行科学的分析,我们得到一些计算机基础教学课程体系改革的启示。在此基础上,按照《基本要求》的精神,结合计算机技术发展和应用的实际,以“知识-技能-能力”培养为目标,对计算机基础课程体系进行了重新的设计和调整,构建了“大学计算机基础+X门计算机应用课程”和“程序设计基础+X门计算机应用课程”两种“1+X”课程体系模式,形成了新型的计算机基础课程教学方案。

在以上课题研究的基础上,我们成立了“高等学校计算机基础教育改革与实践系列教材”编审委员会,希望能编写出一套适合于此教学方案的教材并建设相应的课程教学资源。

该系列教材以“面向应用、强化基础、注重融合”为原则,从面向应用的计算机硬件基础和软件基础两个角度入手,从融合专业技术的发展、社会对现代人才知识结构的要求出发,按照两种“1+X”的课程设置方案,选择了5门比较基础且通用的计算机基础课程来组织编写。

1. 从实践中来,到实践中去。所有教学内容均从应用问题出发,以引例、实例和案例作为背景,提出每章的教学内容与教学目标,使学生对学习什么知识、为什么要学这些知识有一个概括的认识,并通过解决问题使所学基础知识得到强化。所有引例、实例和案例都具有代表性,能激发学习的积极性,达到学以致用目的。

2. 内容新颖,知识结构更加合理。所有教学内容进一步体现了新版《基本要求》的精神,并在此基础上,结合多年来教学改革与实践经验及地方经济结构和行业的需要,并融合相关专业知识,适当地增加了部分内容。同时突破了传统的知识与教学模式,对相关内容的知识结构顺序做了调整,更利于学生对计算机基础知识的理解和掌握。

3. 进一步体现了计算机基础教学的目的和意义。计算机基础课程到底应该学哪些知识?计算机技术的发展水平和社会对计算机知识的需求是什么?计算机基础课程教学的内容怎样适应这种要求?所有这些是衡量计算机基础课程教学成败的关键。要使我们的教学紧跟时代的步

伐,就要在不同的时期调整我们的教学内容。本系列教材在这方面做了大量的调研,对教学内容进行了适当的选择,进一步体现了“大学计算机文化基础”之后的新的知识结构与内容,进一步满足了社会对现代人才面向应用的计算机技能的基本要求。

为了保证教材的编写质量,编委会对本系列教材的编写过程进行了全程把关,各书的主编和编委由来自各个高等学校的计算机基础教学负责人或骨干教师担任,他们都有丰富的教学实践和教材编写经验。可以说本系列教材综合了各高等学校计算机基础教学改革与实践的经验和成果,是集体智慧的结晶。

希望高等学校在使用该系列教材的过程中能够提供有益的帮助和意见,以促进相关课题研究工作的不断深入。

编审委员会
2010年5月

前 言

计算机应用已渗透到科学计算、数据处理、计算机辅助设计与制造、过程控制、人工智能、网络应用等社会生活中的各个领域,正在改变着传统的工作、学习和生活方式,推动着社会的发展。21 世纪的大学生是推动国家经济健康持续发展的主力军,必须掌握计算机这一解决实际问题的工具,才能更好地为国家的经济建设、国防建设、科技进步服务。

C 语言既具备高级语言结构化、模块化的特点,还具备机器语言直接访问计算机硬件的特点,是高效的程序设计语言。C 语言广泛应用于操作系统、大型应用软件、嵌入式系统和工业控制等软件的编程,是应用最为广泛的程序设计语言之一。

“C 语言程序设计”是非计算机专业计算机基础教育的核心课程,也是计算机基础教育的入门课程。学习 C 语言程序设计,不仅要学习 C 语言的语法和句法的规则,还要学习编写程序解决实际问题的程序设计方法。C 语言是面向过程的程序设计语言,C 语言处理问题的方法和计算机的工作原理相类似,通过学习可以建立计算机思维,为学习计算机相关知识打下坚实的基础。C 语言也是进一步学习面向对象程序设计语言、面向 Windows 程序设计语言的基础。

为了让“零基础”的读者能够较好地掌握 C 语言并具备程序设计能力,本书从计算机的基础知识和程序设计的基础知识出发,引入学习 C 语言必备的基础知识。从 C 语言语法和句法、程序设计方法出发,以应用为主线,详细阐述了 C 语言语法和句法的规则和程序设计的方法。

本教材充分体现了以下特点:

1. 通俗易懂。本书是基于“零基础”读者的 C 语言教材,在阐述语法和句法规则时避免了枯燥的、难以理解的概念和理论,以实例和解释的方式阐述规则,使读者易于学习和掌握。
2. 由浅入深。本书首先阐述了学习 C 语言必备的计算机和程序设计的基础知识,然后按照基本类型数据、结构化、模块化、数组、指针、结构体的顺序由浅入深阐述 C 语言的语法和句法的规则。各章节环环相扣,每一章都是下一章的基础。
3. 实例丰富多样。本书给出了大量的解决不同类型实际问题的程序设计实例,详细阐述了使用计算机解决实际问题的数据结构及算法设计,用 N-S 流程图的方式表示了算法,并给出了程序。读者可以从实例中掌握程序设计的方法。
4. 习题覆盖全面。本书精选了选择题、填空题和编程题,几乎覆盖了 C 语言的所有知识点。

本书可作为计算机专业和非计算机专业的教材,也可作为自学教材。建议讲授学时为 48 ~ 64 学时,教师可根据教学对象合理安排讲授学时。

本书由赵山林任主编,高媛任副主编。赵山林编写第 1、3、8 章,高媛编写第 9、10、11 章,孟

朝霞编写第2、6章和附录3,刘静编写第4、5章,乔志会编写第12章和附录1、2、4,吴瑞编写第7章和附录3。全书由赵山林、高媛统稿。

感谢读者选择使用本书,欢迎读者对本书内容提出批评和修改建议。作者的电子邮件地址:shanxy@163.com。

编 者
2010年3月

目 录

第1章 计算机基础知识	1	3.2 C 程序中数据的表示方法	28
1.1 计算机硬件系统的组成及工作 原理	1	3.2.1 常量	29
1.1.1 计算机硬件系统的组成	1	3.2.2 变量	33
1.1.2 计算机的工作原理	2	3.3 运算符及其表达式	36
1.2 数制及数据的存储	3	3.3.1 C 运算符概述	36
1.2.1 数制	3	3.3.2 算术运算符及算术表达式	37
1.2.2 计算机中数据的表示	4	3.3.3 赋值运算符和赋值表达式	38
1.2.3 数制间的转换	4	3.3.4 自增自减运算符及含自增 自减运算的表达式	40
1.2.4 数据在内存中的存储	5	3.3.5 位运算	41
1.3 计算机软件系统的组成	8	3.3.6 逗号运算符和逗号表达式	43
1.3.1 软件系统	8	3.3.7 表达式小结	44
1.3.2 机器语言和高级语言	9	3.4 C 语言中的类型转换	44
第2章 C 语言程序设计基础	11	3.4.1 类型的自动转换	44
2.1 利用计算机解决实际问题的具体 步骤	11	3.4.2 强制类型转换	50
2.2 算法初步	13	习题	51
2.2.1 算法的概念和特性	13	第4章 顺序结构程序设计	53
2.2.2 算法的表示	14	4.1 C 语句概述	53
2.3 结构化程序设计	17	4.1.1 语句的概念	53
2.3.1 结构化程序设计概念	17	4.1.2 C 语句的分类	54
2.3.2 结构化程序设计原则	18	4.2 输出和输入函数	56
2.4 C 语言简介	20	4.2.1 putchar() 和 getchar() 函数	56
2.4.1 C 语言的产生与发展	20	4.2.2 printf() 和 scanf() 函数	58
2.4.2 C 语言的特点	20	4.3 顺序结构程序设计示例	62
2.5 C 语言的基本符号	21	习题	66
2.5.1 C 语言的字符集	21	第5章 选择结构程序设计	69
2.5.2 C 语言的词汇	22	5.1 条件的表示	69
2.6 C 语言程序的基本结构	23	5.1.1 逻辑量的概念	69
第3章 数据类型及表达式	27	5.1.2 关系表达式	69
3.1 数据结构	27	5.1.3 逻辑表达式	70

5.1.4 条件表达式	72	7.5 函数示例	148
5.2 if 语句	73	习题	149
5.2.1 if 语句	73	第 8 章 数组	154
5.2.2 if 语句的嵌套	75	8.1 数组的基本概念	154
5.2.3 选择结构程序设计示例	76	8.2 一维数组	155
5.3 switch 语句	80	8.2.1 一维数组的定义	155
5.3.1 switch 语句	80	8.2.2 一维数组元素的初始化	156
5.3.2 switch 语句在多分支程序 设计中的应用	83	8.2.3 一维数组元素的引用	157
习题	88	8.2.4 一维数组程序设计示例	159
第 6 章 循环结构程序设计	91	8.3 二维数组	166
6.1 循环的概念	91	8.3.1 二维数组的定义	166
6.2 实现循环的语句	94	8.3.2 二维数组元素的引用	167
6.2.1 while 语句	94	8.3.3 多维数组	171
6.2.2 do-while 语句	99	8.4 字符数组及字符串	172
6.2.3 for 语句	101	8.4.1 字符数组的定义、引用及 初始化	172
6.3 break 和 continue 语句	103	8.4.2 字符串处理函数	175
6.4 循环的嵌套	106	8.4.3 字符数组示例	178
6.5 循环结构程序设计	111	习题	181
习题	116	第 9 章 指针	186
第 7 章 函数	122	9.1 指针的基本概念	186
7.1 函数概述	122	9.1.1 变量的直接访问和间接 访问	186
7.1.1 模块化程序设计思想	122	9.1.2 指针与指针变量	187
7.1.2 函数概述	123	9.2 指针变量	188
7.1.3 函数的分类	124	9.2.1 指针变量的类型和指针变量 指向变量的关系	188
7.2 函数的定义	125	9.2.2 指针变量的引用	190
7.2.1 函数的定义形式	125	9.2.3 指针的运算	194
7.2.2 函数的返回值	127	9.2.4 指针变量作为函数参数	196
7.3 函数的声明及函数调用	128	9.3 数组与指针	199
7.3.1 函数声明	128	9.3.1 指向数组元素的指针变量	200
7.3.2 函数的调用及参数传递	130	9.3.2 一维数组作函数参数和指向 变量的指针变量作函数参数 的一致性	204
7.3.3 函数的调用形式及规范	133	9.3.3 多维数组的指针	208
7.3.4 函数的嵌套及递归调用	134	9.3.4 字符串和指针	213
7.4 变量的作用域与生命期	139	9.4 指向函数的指针和返回指针值的	
7.4.1 变量的属性	139		
7.4.2 局部变量和全局变量	140		
7.4.3 变量的存储类型与 生命期	144		

函数	220	10.5.3 链表的操作	269
9.4.1 指向函数的指针	220	10.6 联合体数据类型	280
9.4.2 指向函数的指针变量	221	10.6.1 联合体的定义	280
9.4.3 指向函数的指针变量作函数 参数	223	10.6.2 联合体变量的引用	282
9.4.4 返回指针值的函数	226	10.7 自定义类型	287
9.5 指针数组与多级指针	229	习题	288
9.5.1 指针数组	229	第 11 章 文件	292
9.5.2 多级指针	234	11.1 文件概述	292
9.5.3 命令行参数	236	11.1.1 文件的概念	292
习题	237	11.1.2 缓冲文件系统	293
第 10 章 结构体与联合体	242	11.2 文件的使用	293
10.1 结构体的概念	242	11.2.1 文件类型指针	294
10.1.1 结构体类型的定义	242	11.2.2 文件的打开	294
10.1.2 结构体变量的定义、引用、 初始化	244	11.2.3 文件的关闭	296
10.1.3 结构体的嵌套	249	11.2.4 文件的读写	296
10.2 结构体数组	253	11.2.5 文件的定位	304
10.2.1 结构体数组的定义	253	11.2.6 文件的检测	307
10.2.2 结构体数组的引用	254	11.3 文件程序设计示例	307
10.2.3 结构体数组的初始化	256	习题	312
10.3 结构体指针	256	第 12 章 编译预处理	313
10.3.1 指向结构体变量的指针	256	12.1 宏定义	313
10.3.2 指向结构体数组的指针	259	12.1.1 不带参数的宏定义	314
10.3.3 结构体指针作函数参数	261	12.1.2 带参数的宏定义	318
10.4 位字段	262	12.2 文件包含	320
10.4.1 位字段结构体的定义	262	12.3 条件编译	322
10.4.2 位字段结构体变量成员 的引用	265	习题	324
10.5 链表	265	附录 1 ASCII 字符表	327
10.5.1 链表的概念	266	附录 2 C 语言的运算符及优先级和 结合性	328
10.5.2 内存的动态存储、分配与 释放函数	267	附录 3 Visual C++ 6.0 集成开发 环境	329
		附录 4 常用标准库函数	344

第 1 章

计算机基础知识

计算机已广泛应用于社会生活的各个领域,已成为人们工作、学习不可缺少的重要工具!掌握计算机基础知识是学习 C 语言程序设计、使用 C 语言解决实际问题的基础。本章将着重讲解和 C 语言程序设计相关的计算机基础知识。

1.1 计算机硬件系统的组成及工作原理

计算机是一种具有高度智能化的电子设备。简单地讲,一个完整的计算机系统由硬件系统和软件系统两部分构成。在这里姑且抛开硬件和软件系统复杂的具体形态,从理论角度出发讲述计算机系统的基本组成和工作原理。

计算机硬件系统是由复杂的电子器件和机械部件有机组合而成的,用以实现复杂而精巧的机械和电气功能。硬件是整个计算机系统的基础,是软件系统的载体,它的优良与否直接影响到整个计算机系统的稳定性和安全性。

软件是一种无形的资源,它以硬件系统为平台,用以实现对于计算机硬件的控制和管理,发挥硬件资源的效用,实现一定的功能。软件执行过程及结果可以通过文字、图像、声音等方式体现。软件是人类智慧在充分利用计算机系统资源方面的结晶。人们通过软件让计算机代替人类完成各种复杂的任务,从而提高了劳动效率,缩短了劳动时间,降低了劳动成本。

计算机按其规模和运算速度可分为巨型机、大型机、工作站和微型机。由于巨型机和大型机功能强大、价格昂贵,一般只用于科研、国防和复杂应用等领域。工作站和微型计算机,特别是微型计算机,由于其具有体积小、运算速度快、操作灵活以及功能齐全等优点,近几年得到了广泛应用。

1.1.1 计算机硬件系统的组成

计算机硬件系统的主要组成部分,主要包括中央处理器(CPU)和内存存储器(简称内存)。中央处理器是计算机硬件的核心部件,计算机的绝大多数功能由它来完成。中央处理器主要由运算器和控制器两部分组成。运算器完成数据的运算、逻辑判断等工作,计算机大量的计算工作就是在运算器中完成的。控制器通过指令完成对各个部件的控制以协调计算机各部件的工作。

CPU 的具体形态为一个超大规模的集成电路,包含大量的辅助电路,通过外部引脚与硬件系统电路相连接。

内存储器是计算机硬件系统中极其重要的部件。根据其对数据的存储的方式不同可分为只读存储器(ROM)和随机存储器(RAM)。只读存储器只存储计算机的基本硬件信息,其信息不可改变,这些基本信息对计算机各部件的协调工作具有极其重要的作用,控制器正是通过这些硬件的基本信息完成了对计算机硬件的控制。随机存储器(习惯上称为内存)是计算机数据交换的中心,用以完成计算机数据的临时存放,计算机输入输出的数据都是通过内存到达各部件的。之所以增加内存这一重要部件是由于计算机的 CPU 与外部设备工作速度不同造成的,CPU 的工作速度快,而外部设备的工作速度相对较慢,若不设内存部件,CPU 大量的时间处于等待,形成资源浪费;其次外部设备的种类繁多,通过内存大量的数据信息就可以在内存中暂时存放,有利于充分利用 CPU 资源和协调外部设备各部件之间的工作,从而极大地提高计算机的工作效率。

外部设备可以划分为输入设备、输出设备和外存储器。外部设备中的输入输出(I/O)设备是人与计算机交互的重要途径,通过它们就可以输入数据、控制计算机的工作,查看计算机的工作过程和得到计算结果。输入设备是人与计算机交互的第一通道,人对计算机的控制需通过输入设备来完成,随着计算机功能的强大,输入方式变得更加简单,输入形式日趋多样化。例如,鼠标的应用、手写板的出现和语音系统的发展都使人与计算机的交互从频繁的键盘敲击中解脱出来。输出设备是人与计算机交互的结果性终端设备,计算机工作所得到的文字、图像和声音等结果需通过输出设备体现出来。通过这些终端设备用户得以监视计算机的工作过程,进而为进一步的工作做出判断。目前输出设备向着人性化和多功能化方向发展。

外存储器是计算机用以长期存放数据的设备,外存储器主要有软磁盘(简称软盘)、硬磁盘(简称硬盘)、光盘、磁带等。其中硬盘是计算机最重要的外存储器,几乎所有系统软件和应用软件以及用户数据都存放在这里(网络中的无硬盘瘦客户机除外),硬盘存储器已成为计算机硬件系统中最重要数据存储介质。光盘是利用化学材质和激光原理在盘片上记录数据的存储设备,其数据需通过光学镜头读取,所以称为光盘,光盘可以大量地永久性保存数据,已成为计算机外存储设备中重要的介质。随着计算机技术及相关技术的提高,大容量光盘已成为永久性数据备份和存储的重要途径。磁带是顺序存储数据的介质,磁带可以进行大规模数据的存储和备份,但由于其存取速度较慢,已趋于淘汰。目前外存储器在稳定性、容量和存储速度上都得到了巨大的进步,同时出现了各类大容量可热拔插(计算机工作时可任意插入和拆除的外存储设备)的外存储器,更加方便了数据的输入输出。

1.1.2 计算机的工作原理

冯·诺依曼(John von Neumann,1903—1957年),美籍匈牙利数学家,是计算机基本理论架构的奠基人,他提出了存储程序控制的计算机工作原理。冯·诺依曼计算机工作原理的提出奠定了计算机技术的理论基础,使计算机技术在短短半个世纪得到了突飞猛进的发展。学习冯·诺依曼计算机的工作原理是了解计算机技术的基础,是掌握计算机的工作模式,理解计算机硬件系统和软件系统,学习计算机程序设计的基础。

冯·诺依曼计算机的工作原理的核心思想是二进制和存储程序控制。二进制思想指的是所

有外部世界的信息,如数据、程序、声音、图像、图形等,都必须以二进制的形式表示才能够被计算机传输和处理;存储程序控制指将编写好的程序和原始数据,输入并存储在计算机的内存储器中(即“存储程序”);计算机按照程序逐条取出指令加以分析,并执行指令规定的操作(即“程序控制”)。程序是指令的有序集合,指令是计算机解决实际问题的具体操作。

计算机工作原理如图 1-1 所示。

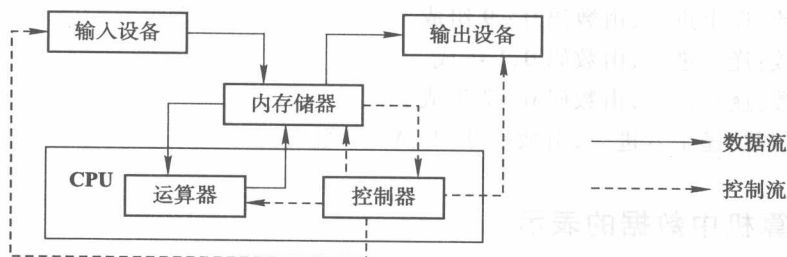


图 1-1 计算机工作原理

从计算机诞生的那一天起,计算机工作的这一原理就决定了人们使用计算机的主要方式——编写程序和运行程序。科学家们一直致力于提高程序设计的自动化水平,改进用户的操作界面,提供各种开发工具、环境与平台,其目的都是为了让人们更加方便地使用计算机,可以少编程甚至不编程来使用计算机,因为计算机编程毕竟是一项复杂的脑力劳动。但不管用户的开发与使用界面如何演变,“存储程序”原理没有变,这一原理仍然是理解计算机系统功能与特征的基础。

1.2 数制及数据的存储

1.2.1 数制

1. 数制的定义

按进位的原则进行记数,称为进位记数制,简称“数制”。在日常生活中最常用的数制为十进制。除了十进制以外,还有许多非十进制的记数方法。例如,60 分钟为 1 小时,用的是六十进制记数法。

(1) 逢 N 进一

N 为数制中所需要的数码的总个数,称为基数。例如,十进制数用 0、1、2、3、4、5、6、7、8、9 这 10 个不同的数码来表示。十进制数码的个数称为十进制的基数。

(2) 位权表示法

位权是指一个数字在某个固定位置上所代表的值,处在不同位置上的数码符号所代表的值不同,每个数码的位置决定了它的值或者位权。而位权与基数的关系是:各进位制中位权的值是基数的若干次幂。因此,用任何一种数制表示的数都可以写成按位权展开的多项式之和。

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

位权表示法的原则是数码的总个数等于基数;每个数符都要乘以基数的幂次,而该幂次是由

每个数所在的位置所决定的。排列方式是以小数点为界,整数部分自右向左 0 次方、1 次方、2 次方……小数部分自左向右负 1 次方、负 2 次方、负 3 次方……

2. 计算机中常用的数制

日常生活中使用的数制有很多种,而在计算机中采用二进制。由于二进制数与八进制数和十六进制数具有特殊的关系,所以在计算机应用中常常根据需要使用八进制或十六进制。

- ① 十进制数:逢十进一,由数码 0~9 组成。
- ② 二进制数:逢二进一,由数码 0、1 组成。
- ③ 八进制数:逢八进一,由数码 0~7 组成。
- ④ 十六进制数:逢十六进一,由数码 0~9、A~F 组成。

1.2.2 计算机中数据的表示

计算机是以对数据的运算来完成各项工作的,但计算机由复杂的电子电路构成,如何让计算机通过电子电路进行计算和数据存储是非常困难的。十进制的数码有 10 个,通过电子元器件来描述这 10 个数码相对困难。由于电子器件有两个相对稳定的状态,可以用 0 和 1 来描述这两个状态,通过 0 和 1 的组合表示不同的数据。所以,计算机内部采用二进制来描述数据。

计算机内部采用二进制方式来描述数据,使用二进制有如下优点:

- ① 便于实现。用具有两种稳定状态的电子元器件就可以表示二进制的两个数码。
- ② 运算简单。二进制的运算由于只有 2 个数码参与运算,相对简单。
- ③ 便于存储。每一个电子元器件可以存储一位二进制数。

但是二进制数也存在数据过长不便记忆的缺点。因此,人们采用八进制和十六进制来简化二进制的书写。例如,记录 100101100 就不如记录 454 的八进制数和 12C 的十六进制数方便,又由于二进制同八进制和十六进制之间的转换非常方便,所以对于计算机中数据的表示常以八进制和十六进制为主。

1.2.3 数制间的转换

1. 十进制数与其他进制数之间的转换

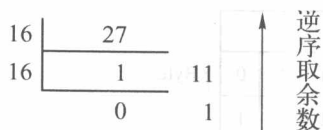
转换规则:除基逆序取余法。

【例 1-2】 $(35)_{10} = (100011)_2$ 。

2	35	
2	17	1
2	8	1
2	4	0
2	2	0
2	1	0
	0	1

逆序取余数

【例 1-3】 $(27)_{10} = (1B)_{16}$



2. 其他进制数转换为十进制数

转换规则:按权展开相加。

【例 1-4】 $(11011)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = (27)_{10}$

【例 1-5】 $(6A)_{16} = 6 \times 16^1 + 10 \times 16^0 = (106)_{10}$

3. 二进制数、八进制数和十六进制数之间的转换

除上述用脚标方式表示不同进制外,还可以使用后缀字母(一般大写)的方式表示。后缀 B 表示二进制,后缀 O 表示八进制,后缀 D 表示十进制,后缀 H 表示十六进制,如二进制数 10101010B、八进制数 123O、十进制数 123D、十六进制数 123H。也可以在后缀字母两侧加小括号,如 123(O)、10(B)等。

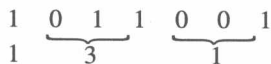
(1) 二进制数、八进制数之间的转换

二进制数转换为八进制数的规则是从低位起每 3 位组合成一位八进制数,八进制数转换为二进制数的规则是从低位起每一位八进制数转换为 3 位相应二进制数,不足 3 位左侧补 0。

(2) 二进制数、十六进制数之间的转换

二进制数转换为十六进制数的规则是从低位起每 4 位组合成一位十六进制数,十六进制数转换为二进制数的规则是从低位起每一位十六进制数转换为 4 位相应二进制数,不足 4 位左侧补 0。

【例 1-6】 $1011001B = 131O$ 。



【例 1-7】 $1011001B = 59H$ 。



【例 1-8】 $53H = 1010011B, 1234O = 1010011100B$ 。

1.2.4 数据在内存中的存储

内存是由一系列的电子电路单元组成,每一个电子电路单元只有两种稳定的状态,即 0 和 1,用于表示二进制位,称之为位(bit)。通常,将连续的 8 个二进制位称之为一个字节(Byte)。连续的两个字节称为字(Word),连续的两个字称为双字(Double Word)。内存的组织形式如图 1-2 所示。

为了实现对内存数据的读写必须明确数据在内存中的地址。地址是针对字节的连续的编号。图 1-2 中所表示的 2000H、2001H 表示的就是所在字节的地址。

为了描述数据在内存中的存储,人们引入了机器码和真值的概念。数据在内存中的存储形

式称为机器码,机器码所表示的实际值称为真值。

	...							
2000H	1	0	0	1	1	1	1	} Byte
2001H	0	0	0	0	1	1	1	
2002H	0	1	1	1	0	0	0	} Word
2003H	1	1	1	0	0	1	1	
	...							

图 1-2 内存的组织形式

1. 整数的存储

假定某计算机系统中一个整数用两个字节存储。

(1) 有符号整数的存储

高字节的最高位为符号位(1 表示负数,0 表示非负数),高字节的其余 7 位和低字节的全部 8 位是数据位。

CPU 中的运算器没有执行减法的部件,要实现减法功能,必须把减法转换为加法运算才能完成。那么如何进行转换呢,先举一个例子加以说明。现在是北京时间 4 点,而你手表的时间却停留在 2 点上,如何拨动手表进行校正?众所周知有两种方法可以实现这一校正过程:第一种方法顺时针拨动两小时,即 $2 + 2 = 4$;第二种方法逆时针拨动 10 小时即 $2 - 10 = 4$,因为 2 点可理解为 14 点, $(14 - 10 = 4)$ 。不论哪一种方法均可完成正确的校正过程,当遇到减法时可以转换为相应的加法运算,就如把 $2 - 10 = 4$ 转换为 $2 + 2 = 4$ 一样,但必须知道为什么可以完成这一转换。观察下列式子:

2-

2+

10

2

=4

=4

之和等于12

$10 + 2 = 12$,而钟表的满刻度即为 12,可见相对于满刻度 12 来说 10 和 2 互补。人们正是根据这一原理提出了补码的概念,利用补码可以把减法转换为加法。

计算机中有符号整数的存储是以补码形式存储的。一个整数有以下 3 种编码。

① 原码。原码是符号位数码化了的二进制。运算规则是将某一有符号整数变为二进制,然后补足 15 位数据位,添上符号位。

如:十进制数 123 转换为二进制数为:1111011
补足数据位添上符号位后得到原码:(0)0000000 01111011

可以简写为:1730 或 7BH。

十进制数 -123 的二进制数为:-1111011

补足数据位添上符号位后得到原码:(1)0000000 01111011

可以简写为:1001730 或 807BH。