

■ 高 等 学 校 教 材

C程序设计

C CHENGXU SHEJI

赵山林 高 媛 等 编著



人民邮电出版社
POSTS & TELECOM PRESS

高等学校教材

C 程序设计

赵山林 高媛 等编著

人民邮电出版社

图书在版编目 (CIP) 数据

C 程序设计/赵山林, 高媛等编著. —北京: 人民邮电出版社, 2004.8

高等学校教材

ISBN 7-115-12552-X

I. C... II. ①赵...②高... III. C 语言—程序设计—

高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 088027 号

内 容 简 介

本书从程序设计的基本思想出发, 以语法、程序设计和计算机思维的培养为重点, 由浅入深、循序渐进地讲述 C 语言的基本概念和程序设计方法。本书列举大量的实例详细讲述解决问题的基本步骤, 同时配有大量的习题, 帮助读者进一步掌握相关的知识。

本书的主要内容包括程序设计的基础知识; 数据类型及相关的类型转换规则, 表达式的构造、计算规则, 指针、结构体、文件的应用; 程序的流程控制语句及结构化的程序设计方法, 归纳了选择、循环程序设计的思路; 函数的设计方法、参数的传递; 中断的基本概念及使用方法; 图形的基本概念及应用等。

本书通俗易懂、图文并茂, 可作为高等学校的教材, 也可供自学者参考阅读。

高等学校教材

C 程序设计

◆ 编 著 赵山林 高 媛 等

责任编辑 邹文波

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

读者热线 010-67129259

北京汉魂图文设计有限公司制作

北京隆昌伟业印刷有限公司印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 21.5

字数: 518 千字

2004 年 8 月第 1 版

印数: 1-9 000 册

2004 年 8 月北京第 1 次印刷

ISBN 7-115-12552-X/TP · 4145

定价: 28.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前 言

随着计算机在各个领域应用的日益普及,必然要求各个专业的学生都应具备一定的计算机基础知识、熟练的操作技能和较强的程序设计能力,因此大学生必须掌握一种或几种程序设计语言以适应各专业的需求。高级语言程序设计是一门重要的必修课。各高等学校对非计算机专业的本科生均开设了“C 语言程序基础”课程。在教学过程中,我们针对非计算机专业学生的特点,对程序设计基础的教学模式进行探讨:语法、解决问题的思维及编写调试程序的能力在程序设计基础中是同等重要的概念,不能只注重语法,也不能只注重程序的编写;学生在编写程序时往往脱离不了数学模式的约束,常常用习惯的逻辑思维代替计算机解决问题的方式,这就要求在教学过程中强调语法的重要性,强调观念的转变,辨析数学模式和计算机模式的相同与不同;学生在编写选择结构及循环结构程序时,感觉无从下手,这就要求在教学过程中培养学生解决问题的能力,给出共性的思维模式,引导学生建立计算机思维;编写程序和调试程序能力的培养是学生程序设计能力的综合体现,教学中应调动学生的积极性,强调理论是基础,实践是关键,培养学生的程序设计能力。

C 语言是一种高效的程序设计语言,广泛应用于多种应用软件的开发,是计算机软件开发的技术基础,也是面向对象程序设计的不可缺少的组成部分。为了让读者较好地掌握 C 语言,掌握程序设计的能力,本书从语法、程序设计角度出发,以应用为主线,较为详细地介绍 C 语言的语法规则、语句执行的流程及编写程序的方法,还简要讲述计算机领域中重要的中断和位的概念、简单的图形设计方法。

本教材的编写充分体现了如下特点。

(1) 通俗易懂。本书的使用对象是非计算机专业的学生,因此避免枯燥的概念及理论的讲解,尽量做到以解释去讲述,使学生易于学习和掌握。

(2) 流程控制和程序设计并进。本书在讲述算法流程时,配有大量的例题,详细阐述解决问题的方法、步骤及利用算法实现语句编写程序的方法、程序和算法的关系。

(3) 由浅入深,循序渐进。本书内容的安排充分考虑初学和非计算机专业的特点,按照基本概念、结构化、模块化、应用的顺序由浅入深地编排,使学生学习本书时不会感到有太多困难。

(4) 图文并茂。为了帮助读者更好地掌握算法和程序的关系,在书中有大量的实例程序,部分程序配有对应的 N-S 结构化流程图,便于初学者养成良好的编写程序的习惯。

本书可作为各高等学校程序设计基础课的教材,也可以供自学者参考阅读。本书由赵山林、高媛、杨喜旺和刘变莲编写,同时感谢韩燮老师的指导,感谢刘红敏老师提出的宝贵意见。

由于编者水平有限,编写时间仓促,书中难免存在缺点和错误,恳请读者批评指正!

编 者

2004 年 8 月

目 录

第1章 计算机基础知识	1
1.1 计算机硬件系统的构成及工作原理	1
1.1.1 计算机硬件系统的构成	1
1.1.2 计算机工作原理	2
1.2 计算机软件系统的组成	4
1.3 数进制及数据的存储	4
1.3.1 数进制	4
1.3.2 计算机中数据的表示	5
1.3.3 进制转换	5
1.3.4 数据在内存中的存储	7
1.4 机器语言和高级语言	10
习题	11
第2章 程序设计基础知识	12
2.1 利用计算机解决实际问题的基本步骤	12
2.2 算法的概念	12
2.2.1 算法的特性	13
2.2.2 算法的基本结构	13
2.2.3 算法的表示	16
2.3 结构化程序设计	19
习题	22
第3章 C语言概述	23
3.1 C语言发展过程及特点	23
3.2 C语言的基本词汇	24
3.3 C程序结构	25
3.4 C语言程序上机调试过程	27
习题	32
第4章 数据类型及表达式	33
4.1 数据结构	33
4.2 C语言程序中数据的表示方法	35
4.2.1 常量	35
4.2.2 变量	39
4.3 运算符及其表达式	43

4.3.1 C 运算符概述	43
4.3.2 算术运算符及算术表达式	45
4.3.3 自增自减运算符及含自增自减表达式	46
4.3.4 赋值运算符和赋值表达式	47
4.3.5 逗号运算符和逗号表达式	49
4.3.6 表达式小结	50
4.4 C 语言中的类型转换	50
4.4.1 类型的自动转换	50
4.4.2 强制的类型转换	55
习题	56
第 5 章 顺序结构程序设计	58
5.1 语句	58
5.1.1 语句的概念	58
5.1.2 C 语句的分类	58
5.1.3 语句和表达式的区别	59
5.2 库函数的使用	60
5.2.1 库函数的使用	60
5.2.2 常用的输出函数	60
5.2.3 常用的输入函数	66
5.2.4 数学函数	69
5.3 顺序结构程序设计	70
习题	72
第 6 章 选择结构程序设计	75
6.1 逻辑量的表示方法	75
6.1.1 关系表达式	75
6.1.2 逻辑运算和逻辑表达式	76
6.1.3 实际问题中逻辑量的描述	77
6.2 if 语句	78
6.2.1 if 语句的格式	78
6.2.2 if 语句的嵌套及多条件结构的实现	80
6.2.3 条件表达式	83
6.3 switch 语句	84
6.4 选择结构程序设计举例	86
习题	93
第 7 章 循环结构程序设计	97
7.1 循环的概念	97
7.2 实现循环的语句	98
7.2.1 while 语句	98
7.2.2 do-while 语句	102

7.2.3 for 语句	104
7.2.4 if 和 goto 构成的循环	107
7.3 break 和 continue 语句	108
7.3.1 break 语句	108
7.3.2 continue 语句	110
7.4 循环的嵌套	111
7.5 循环结构程序设计举例	112
习题	117
第 8 章 函数	122
8.1 概述	122
8.1.1 模块化程序设计思想	122
8.1.2 C 语言程序结构	123
8.1.3 使用函数的必要性	124
8.1.4 函数的分类	124
8.2 函数的定义	124
8.2.1 函数的定义形式	125
8.2.2 函数的返回值	127
8.2.3 形参和返回值的设定	127
8.3 函数的调用	128
8.3.1 对被调用函数的声明	128
8.3.2 函数调用的一般形式	129
8.3.3 函数调用的具体形式	132
8.4 函数的嵌套及递归调用	132
8.4.1 函数的嵌套调用	132
8.4.2 函数的递归调用	133
8.5 局部变量和全局变量	138
8.5.1 局部变量	138
8.5.2 全局变量	139
8.6 变量的存储类别	139
8.6.1 自动存储类型	140
8.6.2 寄存器存储类型	140
8.6.3 外部存储类型	140
8.6.4 静态存储类型	141
8.7 函数举例	142
习题	145
第 9 章 数组	150
9.1 数组的概念	150
9.2 一维数组	151
9.2.1 一维数组的定义	151

9.2.2 一维数组元素的初始化	152
9.2.3 一维数组元素的引用	153
9.2.4 一维数组举例	154
9.3 多维数组的定义和使用	157
9.3.1 多维数组的定义	157
9.3.2 二维数组元素的初始化	158
9.3.3 多维数组元素的使用	159
9.3.4 二维数组举例	160
9.4 字符数组及字符串	163
9.4.1 字符数组的定义、引用及初始化	163
9.4.2 字符串及字符串处理函数	165
9.4.3 字符数组举例	168
9.5 数组作函数参数	170
9.5.1 数组元素作函数的参数	170
9.5.2 数组名作函数的参数	171
9.5.3 多维数组名作函数参数	176
习题	176
第 10 章 指针	180
10.1 指针的基本概念	180
10.1.1 变量的直接访问和间接访问	180
10.1.2 指针与指针变量	181
10.2 指针变量	182
10.2.1 指针变量的类型和指针变量指向变量的关系	182
10.2.2 指针变量的引用	184
10.2.3 指针的运算	187
10.2.4 指针变量作为函数参数	189
10.3 数组的指针	192
10.3.1 一维数组的指针	193
10.3.2 数组名作为函数参数和指针变量作为函数参数的一致性	196
10.3.3 多维数组的指针	198
10.3.4 字符串的指针	202
10.4 函数的指针和返回指针值的函数	208
10.4.1 函数的指针	208
10.4.2 指向函数的指针变量	208
10.4.3 指向函数的指针变量作函数参数	210
10.4.4 返回指针值的函数	213
10.5 指针数组与指向指针的指针	215
10.5.1 指针数组	215
10.5.2 指向指针的指针	219

10.5.3 命令行参数	220
习题	222
第 11 章 结构体、联合体与枚举类型	225
11.1 结构体的概念	225
11.1.1 结构体类型的定义	225
11.1.2 结构体变量的定义、引用、初始化	226
11.1.3 结构体的嵌套	231
11.2 结构体数组	234
11.2.1 结构体数组的定义	234
11.2.2 结构体数组的引用	235
11.2.3 结构体数组的初始化	236
11.3 结构体指针	237
11.3.1 指向结构体变量的指针	237
11.3.2 指向结构体数组的指针	239
11.3.3 结构体指针作函数参数	241
11.4 链表	242
11.4.1 链表的概念	242
11.4.2 内存的动态存储、分配与释放函数	243
11.4.3 链表的操作	245
11.5 联合体数据类型	254
11.5.1 联合体的定义	254
11.5.2 联合体变量的引用	255
11.6 枚举数据类型	259
11.6.1 枚举类型的定义	260
11.6.2 枚举变量的定义和引用	260
11.7 自定义类型	262
习题	263
第 12 章 文件	265
12.1 文件概述	265
12.1.1 文件的概念	265
12.1.2 缓冲文件系统	266
12.2 文件的使用	266
12.2.1 文件类型指针	266
12.2.2 文件的打开	267
12.2.3 文件的关闭	268
12.2.4 文件的读写	269
12.2.5 文件的定位	275
12.2.6 文件的检测	277
12.3 文件程序设计举例	277

习题	280
第 13 章 中断和位运算	281
13.1 中断	281
13.1.1 中断的概念	281
13.1.2 BIOS 中断	282
13.1.3 DOS 中断	283
13.2 位运算	284
13.2.1 位运算符	284
13.2.2 位运算举例	291
13.3 位段	293
13.3.1 位字段结构体的定义	293
13.3.2 位字段结构体变量成员的引用	295
习题	296
第 14 章 C 语言图形设计	298
14.1 显示器的显示属性	298
14.1.1 图形的初始化及关闭	298
14.1.2 图形状态下的色彩选择及设置	301
14.2 常用的图形处理函数	302
习题	310
第 15 章 编译预处理	311
15.1 概述	311
15.2 宏定义	311
15.2.1 不带参数的宏定义	311
15.2.2 带参数的宏定义	315
15.3 文件包含	317
15.4 条件编译	319
习题	321
附录 A 常用字符 ASCII 码表	324
附录 B C 的运算符、优先级和结合性	326
附录 C Turbo C 常用库函数	327
参考文献	332

计算机发展到今天已经形成一个完整的、复杂的体系，其涉及门类广，内涵层次深，已经成为一门综合性学科，与其相关的研究已经深入到了各个基础学科和高尖端学科中。而对计算机系统基础知识的了解是掌握计算机在各学科上应用的基础。因此，本章将着重讲解计算机基础知识。

1.1 计算机硬件系统的构成及工作原理

计算机是一个具有高度智能化的电子设备。这一高度智能化的电子设备所形成的系统是极其复杂的。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，这两部分相辅相成。在这里，我们姑且抛开硬件系统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。软件是人类智慧充分利用计算机系统资源的结晶。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，节约了劳动成本。

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。工作站和微型机，特别是微型机，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。下面以微型机为例讲述计算机的工作原理和数据在内存中的存储等基础知识。

1.1.1 计算机硬件系统的构成

计算机硬件系统按其基本框架主要由主机和外部设备两部分构成，图 1.1 列出了计算机硬件系统的构成。

主机是计算机硬件系统的主要组成部分，主要包括中央处理器（CPU）和内存储器。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单元（运算器）和控制逻辑单元（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

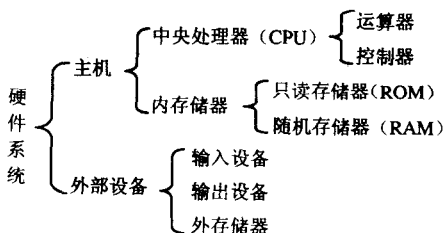


图 1.1 计算机硬件系统的构成

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，运算器和控制器被集成在其中，同时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方式的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，将造成资源浪费；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，从而极大地发挥计算机的工作效率。

外部设备（I/O 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、输出设备是人与计算机对话的重要途径，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，输入方式变得简单化，输入形式日趋多样化。如鼠标的应用、手写板的出现和语音系统的发展都使得人与计算机的交互从频繁的键盘敲击中解脱出来，提高了计算机在非计算机领域的应用范围和便捷程度。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作出判断。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久性存放数据的设备，外存储器主要有软磁盘（简称软盘）、硬磁盘（简称硬盘）、光盘和磁带等，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的数据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）。当前，硬盘已成为计算机硬件系统中最重要数据存放介质。计算机工作时，硬盘会产生大数据量的读写操作，硬盘一旦工作异常，轻者计算机工作不稳定，重者整个计算机系统可能造成瘫痪。光盘是利用光学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。光盘可以永久性地大量保存数据，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。磁带是通过卷状磁带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，但由于其存储和读取速度较慢，已趋于淘汰。目前外存储器在稳定性、容量和存储速度上都取得了巨大的进步，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分离用以存储数据的外存储设备）的外存储器，更加方便了计算机数据的导入、导出。

1.1.2 计算机工作原理

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，他提出了以内存为中心进行数据交换的计算机工作原理，这一原理得到了广泛应用，特别是在微型机上一直沿用到现在。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,使得计算机技术在短短半个世纪得到了突飞猛进的发展。冯·诺依曼计算机工作原理是了解计算机技术的基础,通过对冯·诺依曼计算机工作原理的学习可以掌握计算机的工作模式,加强对计算机硬件系统和软件系统的理解,从而为进一步学习计算机程序设计打下良好的基础。

冯·诺依曼计算机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、图像和图形等,都必须以二进制的形式存储到计算机的存储介质上;存储程序控制指的是利用计算机解决实际问题时,必须将解决实际问题的具体操作步骤以程序的方式存储到计算机中,才能完成对实际问题的解决。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。

按照冯·诺依曼的核心思想,计算机工作原理主要概括为:数据通过输入设备或外存储器进入内存,内存中的数据通过运算器进行运算,运算的结果再返回到计算机的内存中,需要输出的数据通过输出设备输出或存储在外存储器中。这一切工作均在控制器的控制下完成,如图 1.2 所示。

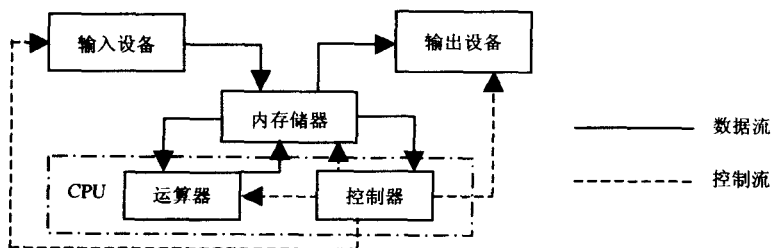


图 1.2 冯·诺依曼计算机工作原理

根据计算机工作原理,CPU 和内存是计算机硬件系统中的核心部件,几乎所有的运算和硬件部件的控制都通过 CPU 来完成,CPU 的工作速度越快,整个计算机的工作能力越强。内存是几乎所有数据和控制指令的“集散地”,内存越大,CPU 和各部件的等待时间就会尽可能少,计算机的工作效率就越高。但当数据的存储和读取与内存的容量达到最大程度上的平衡时,内存容量的额外加大对计算机的性能发挥是无补的。

以冯·诺依曼计算机工作原理为计算机的理论框架基础,计算机各部件逐步发展并形成模块,CPU 与内存、输入设备、输出设备和外存储器等各模块之间以简单的物理接口连接在复杂的电子线路上,该电子线路统称为总线。为了更合理地、稳定地进行数据的传输,数据按类型的不同在 3 类总线上分别传输。此 3 类总线为:地址总线、数据总线和控制总线。

(1) 地址总线

地址总线传送地址信息。地址是识别信息存放位置的编号,内存的每个存储单元及 I/O (输入/输出) 接口中不同的设备都有各自不同的地址。地址总线是 CPU 向内存和 I/O 接口传送地址信息的通道,它是自 CPU 向外传输的单向总线。

(2) 数据总线

数据总线传送系统中的数据或指令。数据总线是双向总线,一方面作为 CPU 向内存和 I/O 接口传送数据的通道;另一方面,是内存和 I/O 接口向 CPU 传送数据的通道,数据总线的宽度与 CPU 的字长有关。

(3) 控制总线

控制总线传送控制信号。控制总线是 CPU 向内存存储器和 I/O 端口发出命令信号的通道。

各类数据通过 3 种总线高速传输,从而保证了计算机高速、正确地运行。模块化和总线理论的提出有利于计算机在不同工作环境、不同用途条件下的任意组合,从而最大程度地利用计算机各部件资源,体现了计算机应用的灵活性,扩大了计算机的应用领域。

1.2 计算机软件系统的组成

软件系统是计算机系统的灵魂,就像人的躯体与人的精神一样,计算机系统的硬件与软件系统是相辅相成的,没有软件的计算机系统是没有能力为用户做任何事情的。计算机软件的应用结果是通过计算机硬件反映出来的。例如,利用计算机播放 VCD 影碟这一软件资源,需要通过计算机媒体播放软件利用光盘驱动器读出 VCD 影碟上的信息并进行处理,然后通过计算机的显示器看到图像,通过声音系统听到声音。所以计算机工作实质上是通过软件驱动计算机硬件来为用户服务的。计算机软件按其作用的不同一般可分为系统软件和应用软件。

系统软件是计算机软件系统的基础部分,通过它可以驱动计算机的硬件部件,使硬件部件处于一个良好的、有序的可应用状态,达到对计算机硬件的管理;在此基础上管理计算机软件系统的数据,合理分配各软件在运行时的权限、规则和检测各软件的运行状态,达到对计算机软件系统的管理。

计算机系统软件根据其功能的不同可分为操作系统软件、诊断系统、语言编译解释系统等。其中操作系统软件是每一台计算机必不可少的,它是计算机系统中计算机得以应用的必要的和重要的平台。目前,计算机操作系统主要有 Windows 操作系统和 UNIX 操作系统等几种,由于操作系统管理的对象是计算机硬件,而计算机硬件系统的基本框架和基本工作原理又大致相同,所以计算机操作系统的种类较少。目前在微型机上以应用 Windows 操作系统居多,网络服务器以应用 UNIX 操作系统为主。

目前,应用领域有大量的应用软件,如用于办公自动化的 Office 系统,用于图像处理方面的 PhotoShop、3DS 等软件,面对工程设计领域的 AutoCAD、Ansys 等。图 1.3 说明了计算机软件系统的分类。

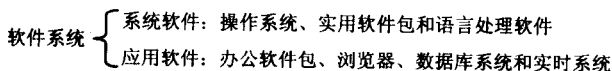


图 1.3 软件系统分类

1.3 数进制及数据的存储

1.3.1 数进制

1. 数制定义

按进位的原则进行计数,称为进位计数制,简称“数制”。在日常生活中经常要用到数制,通常以十进制进行计数,除了十进制计数以外,还有许多非十进制的计数方法。例如,

60 分钟为 1 小时,用的是 60 进制计数法。

(1) 逢 N 进一。

N 为数制中所需要的数码符号的总个数,称为基数。例如,十进制数用 0、1、2、3、4、5、6、7、8、9 这 10 个不同的符号来表示。十进制数码符号的个数,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权值不同,每个数码的位置决定了它的权值或者位权。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。如十进制数“123.45”可以表示如下。

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,每个数码符号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,整数部分自右向左分别为 0 次方、1 次方、2 次方……小数部分自左向右分别为负 1 次方、负 2 次方、负 3 次方……

2. 常用的数制

日常生活中使用的数制有很多种,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,由数字 0~9、A~F 组成。

1.3.2 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,让计算机通过电子电路进行计算和存储数据是非常困难的。十进制的数码有 10 个,通过电子器件来描述这 10 个数码相对困难,由于电子器件有两个相对稳定的状态,我们可以用 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。

二进制的应用具有如下优点。

(1) 便于实现,用具有两种稳定状态的电气元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两个数码参与运算,相对简单。

(3) 便于存储,每一个电气元件可以存储一位二进制数。

但是,二进制数也存在人为记录数据大,不便记忆的缺点,为了便于记录,人们采用了八进制和十六进制。如:记录 100101100 就不如记录八进制数 454 和十六进制数 12C 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这两种进制为主。

1.3.3 进制转换

1. 十进制与其他进制之间的转换

转换规则:整数部分除基逆序取余法,小数部分乘基顺序取整法。

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 $(27)_{10}=(1B)_{16}$

16	27		
16	1	11	
	0	1	

↑
逆序取余数

例 1.4 $(0.6875)_{10}=(0.1011)_2$

	0.6875	
×	2	
	1.3570	
×	2	
	0.7140	
×	2	
	1.4280	
×	2	
	0.8560	
×	2	
	1.7120	

↓
顺序取整数

2. 其他进制转换为十进制

转换规则：按权展开相加。

例 1.5 $(11011)_2=1\times 2^4+1\times 2^3+0\times 2^2+1\times 2^1+1\times 2^0=(27)_{10}$

例 1.6 $(6A)_{16}=6\times (16)^1+10\times (16)^0=(106)_{10}$

3. 二进制、八进制和十六进制之间的转换

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，例如：二进制 10101010B，八进制 123O，十进制 123D，十六进制 123H。也可以在后缀字母两侧加小括号，如 123(O)，10(B)。

(1) 二进制、八进制之间的转换

二进制转换为八进制的规则是从低位起每 3 位组合成 1 位八进制数，八进制转换为二进制的规则是从低位起每一位八进制位转换为 3 位相应二进制数，不足 3 位左侧补 0。

(2) 二进制、十六进制之间的转换

二进制转换为十六进制的规则是从低位起每 4 位组合成一位十六进制数，十六进制转

换为二进制的规则是从低位起每1位十六进制位转换为4位相应二进制数,不足4位左侧补0。

例 1.7 $1011001\text{B}=131\text{O}$

1 0 1 1 0 0 1
 1 3 1

例 1.8 $1011001\text{B}=59\text{H}$

0 1 0 1 1 0 0 1
 5 9

例 1.9 $53\text{H}=1010011\text{B}$, $1234\text{O}=1010011100\text{B}$

1.3.4 数据在内存中的存储

先看一下内存的组织形式。计算机对数据的存储并不是杂乱无章的,对于存储的数据必须知道存储到何处去,对该数据进行提取时需从何处提取,因此必须建立一个完整的存储体系结构,以保证数据的完整性和安全性。这一整套体系就是内存的组织形式。

内存的组织形式如图 1.4 所示。系统把内存划分成等大小的许许多多存储单元,每个单元可以存放 8 个二进制位,每个二进制位实际是一个电气元件用以表示二进制 0 或 1,称之为一个位 (bit)。为了找到某一数据存放的具体内存单元,必须能惟一地标识该存储单元。对内存存储单元的标识是通过地址实现的,地址是存储单元在内存中的位置,每 8 个连续的二进制位有惟一确定的地址编号。这 8 个连续的二进制位所组成的存储单元称之为一个字节 (Byte)。由于计算机存储的数据有可能很大,而一个字节存储的数据的大小是有限的,为了存储更大的数据,往往需要几个字节结合起来存放一个数据,称之为字节扩展。连续的两个字节称为字 (Word),连续的两个字称为双字 (Double Word)。

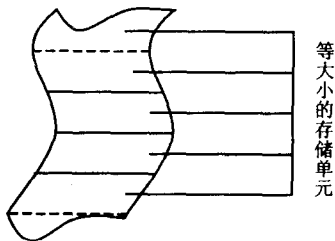


图 1.4 内存的组织形式

为了描述数据在内存中的存储,首先介绍一下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。下面来看一下数据是如何存储在计算机内存中的。

1. 整数的存储

假定某计算机系统中一个整数用一个字节存储。当存储有符号数时,该字节的最高位为符号位 (1 表示负数,0 表示非负的数),其余位是数据位;当存储无符号数时,该字节的所有位都是数据位。

(1) 有符号数的存储

CPU 中的运算器没有执行减法的部件,要实现减法功能,必须把减法转化为加法运算才完成。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:第一种是顺时