

工业和信息化部普通高等教育“十二五”规划教材立项项目

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

# C语言 程序设计 (第2版)

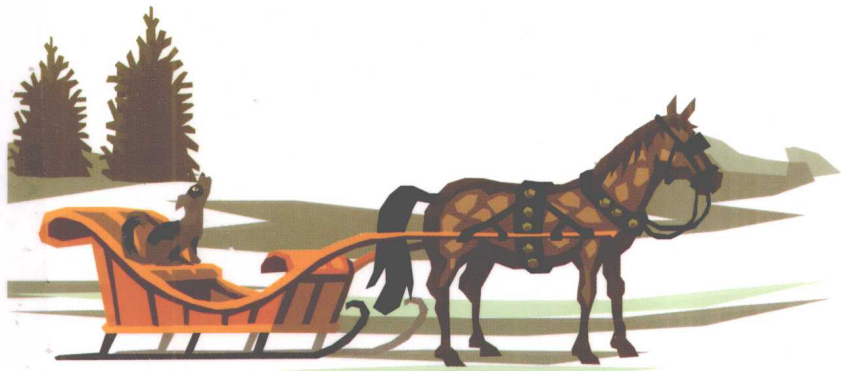
The C Programming Language (2nd Edition)

安俊秀 主编

于华 陈晓红 副主编

相万让 主审

- 重构教学内容和顺序，构成多个学习平面，可根据需求自由组合
- 以理论、实践、工程三大核心要素构成学习平面
- 以模块化程序设计为牵引，形成螺旋学习，实现靶向培养
- 注重教材立体化配套，提供丰富的教学资源



高校系列

人民邮电出版社  
POSTS & TELECOM PRESS

工业和信息化部普通高等教育“十二五”规划教材立项项目

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

# C语言 程序设计 (第2版)

The C Programming Language (2nd Edition)

安俊秀 主编

于华 陈晓红 副主编

相万让 主审



高校系列

人民邮电出版社

北京

## 图书在版编目 (CIP) 数据

C语言程序设计 / 安俊秀主编. — 2版. — 北京 :  
人民邮电出版社, 2010.9  
21世纪高等学校计算机规划教材  
ISBN 978-7-115-23156-7

I. ①C… II. ①安… 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141211号

## 内 容 提 要

本书针对高等院校学生的特点和认识规律,全面、系统地介绍了C程序设计及应用的知识。全书分为三篇,第一篇介绍C程序设计基础知识,包括C程序设计预备知识,程序设计基础,C语言的基本概念,顺序、分支和循环结构程序设计,函数和数组。第二篇介绍C语言高级编程技术知识,包括编译预处理、指针、结构体、共用体和枚举类型、文件。第三篇是C语言综合应用与实践,培养综合编程能力及程序编码的规范性。包括程序编码规范、学生成绩管理系统等知识。

本书结构清晰、语言通俗易懂、实例丰富。每章都配有同步练习,用于熟练和巩固所学知识。附录中提供了相应的常用资料,以方便师生查阅相关内容。

本书可作为大学本科计算机和相关专业的“C程序设计”教材,也可供计算机培训班的学员学习参考。

21世纪高等学校计算机规划教材

C语言程序设计(第2版)

- 
- ◆ 主 编 安俊秀  
副 主 编 于 华 陈晓红  
主 审 相万让  
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>  
北京鑫正大印刷有限公司印刷
  - ◆ 开本: 787×1092 1/16  
印张: 21.25 2010年9月第2版  
字数: 558千字 2010年9月北京第1次印刷

ISBN 978-7-115-23156-7

定价: 37.00元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

C语言是目前国内外广泛使用的计算机程序设计语言之一,是国内外大学都在开设的重要基础课之一。C语言功能丰富、表达能力强、目标程序效率高、可移植性好;既具有高级语言的特点,又具有汇编语言的特点,系统处理能力较强。它既适合编写应用软件,又适合编写系统软件。因此,熟练使用C语言不仅是计算机技术人员应具备的基本功,也是广大计算机应用人员应该具备的基本能力。

由于C语言牵涉到的概念比较复杂,规则繁多,使用灵活,容易出错,不少初学者感到困难。为此,我们总结多年来从事C语言教学工作的经验,希望编写出一本适合于高等院校学生使用的C语言的教程。本书针对高等院校学生的特点和认识规律,精选内容,分散难点,降低台阶,通过丰富的实例和通俗易懂的叙述阐明复杂的概念,并力求做到内容新颖、概念清晰、实用性强、通俗易懂,从而使学生少走弯路,对C语言的感觉由害怕变为兴趣盎然。

本书共14章,分为三篇。第一篇为C语言程序设计基础,在学习了这些内容之后,学生可以编写、调试和运行一般规模和难度的应用程序,并对结构化程序设计方法有所了解,为编写较大规模的应用程序打下基础。第二篇为C语言高级编程技术,这一部分强调对基本概念的理解和掌握,并在理解和掌握的基础上运用高级编程技术的方法,锻炼、培养较大规模、比较复杂的应用程序的编程能力。第三篇为C语言综合应用与实践,以实际案例为主线,引入软件工程的思想,介绍软件开发的方法,培养学生分析问题和解决实际问题的能力。

本书具有以下特点。

(1) 根据学习对象的实际情况,增加了C语言程序设计预备知识的讲解,对于没有任何计算机基础的、不同层次的学生提供了统一的学习平台,使前后知识的学习没有断层,有利于后续章节的理解。

(2) 对C语言公认的难点进行了分解,如数组、函数、指针等,本着先易后难、平滑过渡、循序渐进、易读易懂的思路进行了整体结构设计。在本书的结构体系方面充分注意到章节知识点的渐进性、前后表述的一致性和疑难点的可理解性。

(3) 程序示例中较早引入函数概念,有助于学生理解模块化程序设计的思想。

(4) 针对解题难、实验难的问题,每一章都提供了丰富的典型实例,并配有同步练习题。这对深化许多重要概念,掌握各章节知识要点、解读知识难点有很大的帮助,也使学生们节约许多时间。本书中的全部例题均在Visual C++ 6.0上调试通过。

(5) 注重课程实践性强的特点,引入软件工程的思想,指导C语言的综合应用与实践。教材中增加了程序编码规范方面的内容,有利于培养学生良好的编程习惯。在综合实践中,引入软件工程的思想,使学生初步了解规范的软件开发过程,培养团队协作的能力,加大程序编码量的训练,切实地培养程序设计能力。

(6) 强调引导性教学,注重基于算法的逻辑思维训练,培养学生分析问题的能

力;另一方面强调学生要多上机练习,培养学生良好的自学能力。本教材的参考教学学时数为 108 学时,其中课堂教学 72 学时,上机实验 36 学时。

(7)注重教材的立体化配套,陆续配套课件、习题及解答、学生上机实验指导等辅助教学资源。为方便教师备课,本教材还提供制作多媒体课件的相关素材,供教师索要,邮件地址:anjunxiu@cuit.edu.cn。

本书由安俊秀任主编,于华和陈晓红任副主编。其中,第 2 章、第 4 章、第 5 章、第 10 章由安俊秀编写,第 1 章、第 12 章由赵宇兰编写,第 3 章、第 9 章由董妍汝编写,第 6 章、第 7 章由陈晓红编写,第 8 章、第 13 章由闫俊伢编写,第 11 章、第 14 章、附录 1、附录 2 由于华编写,附录 3 由于华、董妍汝、赵宇兰编写,第 5 章程序例题部分的执行由闫俊伢调试。全书由相万让、安俊秀统稿,相万让审校。

由于编者水平有限,书中难免存在错误之处,敬请读者批评指正。

编者

2010 年 6 月

# 目 录

## 第一篇 C 语言程序设计基础

<b>第 1 章 C 语言程序设计预备知识</b> .....	1
1.1 计算机系统组成及工作原理简介.....	2
1.1.1 硬件系统的组成及其工作原理.....	2
1.1.2 软件系统的组成.....	4
1.2 进位计数制及其转换.....	4
1.2.1 计算机中数制的表示.....	4
1.2.2 非十进制数和十进制数的转换.....	5
1.2.3 二进制数、八进制数和十六进制数的转换.....	6
1.3 机器数的表示形式及其表示范围.....	7
1.3.1 真值与机器数.....	7
1.3.2 数的原码、反码和补码.....	8
1.3.3 无符号整数与带符号整数.....	9
1.3.4 字符的表示法.....	9
习题 1.....	10
<b>第 2 章 C 语言概述</b> .....	11
2.1 C 语言的发展及特点.....	11
2.1.1 程序设计语言的发展.....	11
2.1.2 C 语言的起源与发展.....	12
2.1.3 C 语言的特点.....	13
2.1.4 C 语言学习的相关网站.....	13
2.2 C 语言应用领域概述.....	14
2.2.1 C 语言在系统开发中的应用.....	14
2.2.2 C 语言在嵌入式系统开发中的应用.....	14
2.2.3 C 语言在商业应用软件开发中的应用.....	15
2.2.4 C 语言在硬件驱动开发、游戏设计中的应用.....	15
2.3 C 程序的基本结构.....	16
2.3.1 简单的 C 程序实例.....	16
2.3.2 C 程序的结构特点.....	17
2.4 C 程序的上机步骤.....	19
2.4.1 用计算机解决实际问题的步骤.....	19
2.4.2 运行 C 程序的一般步骤.....	20
2.4.3 Turbo C 集成开发工具简介.....	21
2.5 Visual C++ 6.0 集成环境介绍.....	22
2.5.1 Visual C++ 6.0 界面简介.....	22
2.5.2 Visual C++ 6.0 环境设置.....	23
2.5.3 在 Visual C++ 6.0 中编辑和运行 C 程序.....	25
习题 2.....	28
<b>第 3 章 基本数据类型、运算符与表达式</b> .....	29
3.1 常量与变量.....	29
3.1.1 C 语言的基本元素.....	29

3.1.2	数据和数据类型	30
3.1.3	常量	31
3.1.4	变量	32
3.2	基本数据类型	33
3.2.1	整型数据	33
3.2.2	实型数据	36
3.2.3	字符型数据	37
3.2.4	不同类型数据之间的混合运算	40
3.3	三大运算符及其表达式	40
3.3.1	算术运算符及其算术表达式	41
3.3.2	关系运算符及其关系表达式	42
3.3.3	逻辑运算符及其逻辑表达式	42
3.4	其他运算符及其表达式	44
3.4.1	赋值运算符及其赋值表达式	44
3.4.2	自增自减运算符及其表达式	45
3.4.3	条件、强制类型转换运算符及其表达式	47
3.4.4	求字节、逗号运算符及其表达式	48
3.4.5	取地址运算符	49
3.5	运算符的优先级与结合性	49
	习题 3	51
<b>第 4 章</b>	<b>算法与结构化程序设计</b>	<b>54</b>
4.1	算法的概念及特征	54
4.1.1	程序和程序设计	54
4.1.2	算法的概念	55
4.1.3	算法的特征	55
4.2	算法的表示方法	56
4.2.1	用自然语言表示算法	56
4.2.2	用传统流程图表示算法	57
4.2.3	用 N-S 图表示算法	59
4.2.4	用伪代码表示算法	61
4.3	结构化程序设计方法	62
4.3.1	结构化程序设计的概念	62
4.3.2	结构化程序设计方法	63
4.3.3	算法的合理性与优化	66
	习题 4	68
<b>第 5 章</b>	<b>C 语言程序的基本控制结构</b>	<b>70</b>
5.1	C 语句概述	70
5.2	顺序结构程序设计及其函数	72
5.2.1	顺序结构程序设计思想	72
5.2.2	字符输入/输出函数的应用	73
5.2.3	按格式输出函数的应用	74
5.2.4	按格式输入函数的应用	77
5.3	顺序结构程序举例	80
5.4	选择结构程序设计及其语句	82
5.4.1	选择结构程序设计思想	82
5.4.2	if 语句的应用	83

5.4.3	switch 开关语句的应用	88
5.5	选择结构程序举例	91
5.6	循环结构程序设计及其语句	95
5.6.1	while 循环语句的应用	95
5.6.2	do ~ while 循环语句的应用	96
5.6.3	for 循环语句的应用	97
5.6.4	循环的嵌套	100
5.6.5	几种循环的比较	101
5.7	辅助控制语句及循环结构程序举例	101
5.7.1	辅助控制语句的应用	101
5.7.2	循环结构程序举例	104
5.8	程序的调试	106
5.8.1	编译出错信息理解与调试	106
5.8.2	Visual C++ 6.0 中的程序调试	109
	习题 5	110
<b>第 6 章</b>	<b>数组</b>	<b>115</b>
6.1	问题的提出	115
6.2	一维数组	116
6.2.1	一维数组的定义	116
6.2.2	一维数组的引用	117
6.2.3	一维数组的初始化	117
6.2.4	一维数组的应用	119
6.3	二维数组和多维数组	125
6.3.1	二维数组的定义	125
6.3.2	二维数组的引用	126
6.3.3	二维数组的初始化	126
6.3.4	二维数组的应用	128
6.3.5	多维数组的定义和引用	132
6.4	字符数组和字符串	133
6.4.1	字符数组的定义、初始化和引用	133
6.4.2	字符串与字符数组	134
6.4.3	字符数组的输入/输出	134
6.4.4	常用字符串处理函数	136
6.4.5	字符数组的应用	140
	习题 6	142
<b>第 7 章</b>	<b>函数和变量的作用域</b>	<b>146</b>
7.1	函数概述	146
7.1.1	模块化程序设计方法	146
7.1.2	C 是模块化程序设计语言	147
7.1.3	函数的分类	148
7.2	函数的定义与调用	149
7.2.1	函数的定义	149
7.2.2	函数的参数和返回值	151
7.2.3	函数声明	153
7.2.4	函数的调用和参数传递	155
7.3	函数的嵌套调用和递归调用	157
7.3.1	函数的嵌套调用	158



7.3.2 函数的递归调用	159
7.4 数组作为函数的参数	162
7.4.1 数组元素作函数实参	162
7.4.2 一维数组名作函数实参	162
7.4.3 二维数组名作函数实参	163
7.5 变量的作用域与生存期	164
7.5.1 局部变量及其存储类型	166
7.5.2 全局变量及其存储类型	170
7.6 内部函数和外部函数	174
7.6.1 内部函数	174
7.6.2 外部函数	175
7.6.3 如何运行一个多文件的程序	176
7.7 程序综合示例	177
习题 7	179

## 第二篇 C 语言程序设计基础

第 8 章 编译预处理	184
8.1 宏定义	184
8.2 “文件包含”处理	188
8.3 条件编译	190
8.4 程序示例	193
习题 8	194
第 9 章 指针的应用	196
9.1 指针概述	196
9.1.1 变量与地址	196
9.1.2 指针与指针变量	196
9.1.3 &与*运算符	197
9.1.4 直接访问与间接访问	197
9.2 指针变量	197
9.2.1 指针变量的定义、初始化及引用	197
9.2.2 零指针与空类型指针	200
9.2.3 指针变量作为函数参数	200
9.3 指针与数组	202
9.3.1 指向数组元素的指针变量的定义与赋值	202
9.3.2 数组元素的表示方法	203
9.3.3 指针变量的运算	204
9.3.4 指针与二维数组	205
9.3.5 指针数组	209
9.4 指针与字符串	211
9.4.1 字符串的表示形式及其相关操作	211
9.4.2 字符指针作函数参数	213
9.5 函数指针与指针函数	214
9.5.1 函数指针及指向函数的指针变量	214
9.5.2 指针函数	215
9.5.3 指向指针的指针	216
9.6 带参数的 main 函数	217

9.7 指针的应用举例	219
习题 9	221
<b>第 10 章 结构体、共用体及枚举类型的应用</b>	<b>224</b>
10.1 结构体的应用	224
10.1.1 结构体类型的定义	224
10.1.2 结构体变量的声明	226
10.1.3 结构体变量的初始化	227
10.1.4 结构体变量的引用	227
10.2 结构体数组	228
10.3 指向结构体的指针	230
10.4 结构体与函数	232
10.4.1 函数的形参与实参是结构体	232
10.4.2 函数的返回值类型是结构体	233
10.5 共用体的应用	235
10.5.1 共用体类型的定义	236
10.5.2 共用体变量的声明和引用	236
10.5.3 共用体变量程序举例	238
10.6 单链表的应用	239
10.6.1 链表概述	239
10.6.2 动态分配内存库函数	240
10.6.3 单链表的基本操作	241
10.6.4 单链表的应用举例	245
10.7 枚举类型	248
10.8 类型定义	250
习题 10	251
<b>第 11 章 文件</b>	<b>253</b>
11.1 C 文件概述及文件类型指针	253
11.1.1 C 文件概述	253
11.1.2 文件的分类	253
11.1.3 文件类型指针	254
11.2 文件的操作	254
11.2.1 文件的打开和关闭操作	254
11.2.2 文件读写操作	256
11.2.3 文件的定位	264
11.2.4 文件出错的检测	266
11.3 库文件	267
11.4 文件操作应用举例	268
习题 11	271
<b>第三篇 C 语言综合应用与实践</b>	
<b>第 12 章 程序编码规范</b>	<b>274</b>
12.1 标识符命名规范	274
12.2 代码编写格式	276
12.2.1 清晰的表达式	277
12.2.2 语句的规范性	278

12.2.3 缩进的书写格式	280
12.2.4 一致性和习惯用法	282
12.2.5 程序描述的层次	283
12.3 文档注释	284
12.3.1 注释	284
12.3.2 注释的书写格式	284
12.3.3 注释的分类及使用	285
习题 12	287
<b>第 13 章 位运算</b>	<b>288</b>
13.1 位运算符及应用	288
13.2 位运算应用举例	291
13.3 位段	292
习题 13	296
<b>第 14 章 学生成绩管理系统</b>	<b>298</b>
14.1 需求分析	298
14.1.1 需求分析的基本概念	298
14.1.2 需求分析的方法	298
14.1.3 学生成绩管理系统需求分析	299
14.2 总体设计	299
14.2.1 模块设计	300
14.2.2 系统流程图	301
14.2.3 数据结构设计	301
14.3 详细设计	303
14.3.1 创建学生成绩信息文件 creatstu.c	303
14.3.2 学生成绩浏览 printstu.c	305
14.4 核心代码实现	307
14.4.1 安全验证模块	307
14.4.2 学生成绩管理模块	309
14.4.3 用户管理模块	317
14.4.4 系统主控平台	318
<b>附录 1 常用字符与 ASC II 代码对照表</b>	<b>320</b>
<b>附录 2 Visual C++ 6.0 常见错误信息表</b>	<b>322</b>
<b>附录 3 Visual C++ 常用库函数一览表</b>	<b>324</b>
<b>参考文献</b>	<b>329</b>

# 第一篇

## C 语言程序设计基础

C 语言是目前世界上使用最广泛的高级程序设计语言之一，既适合开发系统程序，又适合开发应用程序。因此，深受广大计算机应用人员的青睐。本篇主要介绍 C 语言程序设计的预备知识和基础知识，包括：计算机系统组成及工作原理简介、进位数制及其转换、基本数据类型与运算符、简单的算法设计、程序的基本控制结构、数组、函数调用和变量的作用域等知识点。在学习了这些内容之后，应能编写、调试和运行一般规模和难度的应用程序（如数值计算类程序），并对结构化程序设计方法有所了解，为编写系统程序和较大规模的应用程序打下基础。

# 第 1 章

## C 语言程序设计预备知识

计算机是以硬件系统为基础，能够通过各种系统软件和应用软件对信息进行自动处理的电子装置。这里的“信息”包括数值、文字、符号、语言、图形和图像等，不论哪种信息在计算机内部最终都必须以数字化编码的形式来表示和处理。程序设计语言就是对这些信息进行处理的软件工具。学习 C 程序设计语言之前，除了要了解计算机的工作原理之外，更重要的是要了解各种信息在计算机中的表示形式，只有了解了这些底层细节，才会理解计算机系统和计算机语言对数据做了什么，才会知道如何最大限度地利用可用的硬件资源编写出高效的程序代码。

本章重点介绍计算机内各种信息的常用编码方案，这些内容是进行程序设计的基石，是学习 C 语言必须掌握的基础知识。

### 1.1 计算机系统组成及工作原理简介

一个完整的计算机系统由硬件和软件两大部分组成。硬件是构成计算机系统的物理装置，是看得见摸得着的设备实体。软件是计算机程序及相关的技术文档资料。软件依赖硬件的物质条件，而硬件则需要软件的支配下才能有效地工作。

#### 1.1.1 硬件系统的组成及其工作原理

计算机问世 60 多年来，尽管计算机硬件系统在性能指标、运算速度、工作方式、应用领域和价格等方面存在差异，但是它们的基本体系结构却是相同的，都是冯·诺依曼机。冯·诺依曼机由微处理器、存储器及 I/O 接口等大规模或超大规模集成电路芯片组成，各部分之间通过“总线”连接在一起，实现信息的交换。图 1-1 给出了计算机硬件系统的基本组成图。

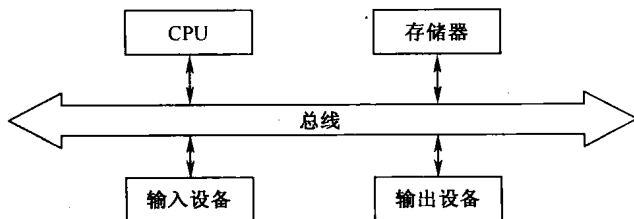


图 1-1 计算机硬件系统的基本组成

### 1. 微处理器 (CPU)

运算器和控制器被集成在同一块微处理器芯片上, 统称为微处理器或 CPU 芯片。微处理器是计算机硬件系统的核心, 其重要性好比大脑对于人一样。它是计算机的运算和控制中心, 负责处理、计算计算机内部的所有数据。

运算器是对二进制数据进行加工和处理的逻辑部件。因为计算机内部是依靠模拟电路来存储和计算的, 电路的开关状态正对应二进制的 0 和 1。运算器根据器件的物理状态表示和处理二进制数, 不仅能够非常容易地实现基本的算术运算和逻辑运算, 而且具有高的可靠性。

控制器是计算机的“神经中枢”, 是协调指挥计算机各部件和谐工作的元件。它能够综合有关的逻辑条件与时间条件, 并按照主频的节拍产生相应的微控制信号, 以指挥计算机各部件按照指令功能的要求自动执行指定的操作。

### 2. 存储器

存储器是计算机的“记忆系统”, 是存放程序和数据的逻辑部件。根据作用不同, 存储器分为内存储器 (简称内存) 和外存储器 (简称外存)。内存是 CPU 能根据地址直接寻址的存储空间, 它用来存放当前正在使用的或者随时要使用的程序或数据。其特点是速度快、容量小, 价格较高。外存 (如硬盘) 用来存放内存的副本和暂时不用的程序或数据。当需要处理外存中的程序或数据时, 必须通过输入输出指令, 将其调入内存中才能被 CPU 执行处理。外存的存取速度比内存慢, 但容量比内存大得多, 并且可以永久保存信息。

### 3. 输入设备/输出设备

输入设备与输出设备是实现人机交互的主要部件。输入设备用来接收用户输入的原始程序或数据, 并将它们转变为计算机能识别的二进制数据存入到内存中, 其功能类似于人的“眼”和“耳”——既能看又能听。输出设备用来将计算机处理的结果转变为人们能接受的形式输出, 功能类似于人的“手”和“嘴”——既能写又能说。目前常用的输入设备有键盘、鼠标、触摸屏、光笔、画笔、图形板、摄像机、图文扫描仪和图文传真机等, 输出设备有显示器、打印机、绘图仪和音箱等。

冯·诺依曼体系结构的设计思想可以简单地概括为“指令存储, 顺序执行”。按照冯·诺依曼存储程序的原理, 计算机系统的工作工程就是反复进行“取指令→分析指令→执行指令”的过程, 具体过程如图 1-2 所示, 即在控制器的控制下, 计算机通过输入设备输入程序或数据, 并自动存放在存储器中, 然后控制器通过地址访问存储器, 逐条取出指令、分析指令, 并根据指令产生的控制信号控制其他部件执行这条指令中规定的任务。这一过程周而复始, 实现了程序的自动执行。

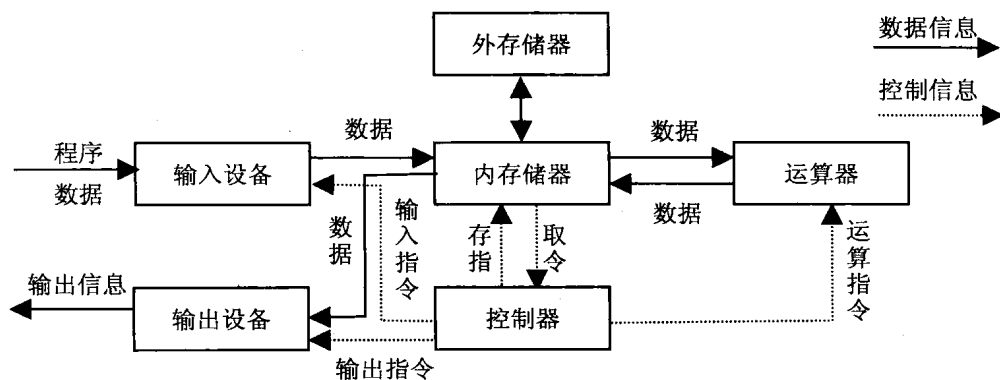


图 1-2 计算机硬件系统的工作原理

## 1.1.2 软件系统的组成

只有硬件的计算机（裸机）是无法运行的，还需要软件的支持。所谓软件是指使计算机正常运行需要的程序及相关技术文档的总称，其中更为重要的是程序，在不严格的情况下，通常认为程序就是软件。如今，软件技术已经渗透到生活、生产的各个领域，即便人们不了解计算机的内部结构和原理，也可以灵活地使用计算机为人们工作、服务。计算机能够协助人类完成各种各样的任务，正是依赖于各种用途的软件。

根据软件用途，计算机的软件分为系统软件和应用软件。

系统软件是为了计算机能正常、高效工作所配备的各种管理和监控程序，在系统一级上提供服务。主要分为两类：一类是面向计算机的软件，如操作系统、各种服务性程序；另一类是面向用户的软件，如编程语言处理程序，以及为适应事务处理的需要而开发的数据库管理系统。

应用软件是为解决用户的特殊需要而开发的特定程序。这类软件通常与具体的应用相关，可以应用于科学计算、数据处理、商业经营、经济管理、工程控制、企业管理等各个领域。应用软件种类非常多，包括办公软件、电子商务软件、通信软件、行业软件和游戏软件等。随着软件技术的发展，这类软件将越来越多。初学 C 语言的读者的主要任务是学习如何编写应用软件。

## 1.2 进位计数制及其转换

什么是进位计数制？进位计数制是利用固定的数字符号和统一的规则来表示数值的方法，可使用的数字符号的个数称为基数，基数为  $n$ ，即可称  $n$  进位制，简称  $n$  进制。

在日常生活中，常见的数值计算都是十进制，如“十角等于 1 元”、“十两等于 1 斤”，这种数制系统使用阿拉伯数字 0~9 组成的数字序列来表示数值。除此之外，还会遇到其他进制，如“每小时 60 分钟”是六十进制，“一年 12 个月”是十二进制，古衡器流行时期的“半斤八两”是指十六进制。在计算机系统中，由于数字电路采用电压的高低或通断表示两种状态（计算机内部采用 0、1 表示），所以计算机内部的所有信息都是采用二进制表示。

### 1.2.1 计算机中数制的表示

#### 1. 十进制表示

十进制（基数为 10）是使用阿拉伯数字字符串来表示数值的方法。它包括“十进位”和“位值制”两条规则。其中，“十进位”是指按“满十进一”、“借一当十”的原则进行计数。“位值制”是指每一个数码符号根据它在这个数中所处的位置来决定其实际数值，即各数位的位权是以 10 为底的幂次方。

按照“十进位”与“位值制”的规则，任意一个十进制数  $D$ ，可表示成如下形式：

$$(D)_{10} = D_{n-1} \times 10^{n-1} + \cdots + D_1 \times 10^1 + D_0 \times 10^0 + D_{-1} \times 10^{-1} + \cdots + D_{-m} \times 10^{-m}$$

其中， $(D)_{10}$  为十进制数的下标表示法； $D$  为数位上的数码，其取值范围为 0~9； $n$  为整数位个数， $m$  为小数位个数；10 为基数； $10^i$  是十进制数的位权（ $i$  为  $n-1$ 、 $\cdots$ 、1、0、 $-1$ 、 $\cdots$ 、 $-m$ ）。

如十进制序列  $(123.45)_{10}$  可以表示为：

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

## 2. 二进制表示

二进制（基数为2）是指使用“0”和“1”两个基本符号来表示数值，按照“二进位”规则进行计数的方法。二进制是计算机信息表示和信息处理的基础。

与十进制数“形象”类似，任意一个二进制数 $B$ ，按照位权 $2^i$ 展开，可以表示成如下形式：

$$(B)_2 = B_{n-1} \times 2^{n-1} + \dots + B_1 \times 2^1 + B_0 \times 2^0 + B_{-1} \times 2^{-1} + \dots + B_{-m+1} \times 2^{-m+1} + B_{-m} \times 2^{-m}$$

如二进制序列 $(1001.1)_2$ 可以表示为：

$$(1001.1)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1}$$

## 3. 十六进制表示

由于二进制书写冗长、易错、难记，为了提高程序的可读性，程序员一般在程序源文件中使用二进制的“短格式”——十六进制。十六进制（基数为16）是使用阿拉伯数字0~9和A~F（依次对应于十进制数10~15）来表示数值，并按照“十六进位”的原则进行计数的方法，如 $123_{16}$ 、 $ABFC_{16}$ 、 $8976_{16}$ 、 $20FA.BC_{16}$ 都是合法的十六进制数。

同样，可以将十六进制序列 $(20A5.BC)_{16}$ 按位权 $16^i$ 展开，表示成如下形式：

$$(20A5.BC)_{16} = 2 \times 16^3 + A \times 16^1 + 5 \times 16^0 + B \times 16^{-1} + C \times 16^{-2}$$

## 4. 八进制表示

八进制（基数为8）表示法在早期的计算机系统中应用非常广泛，适用于12位和36位的计算机系统。它是使用阿拉伯数字0~7来表示数值的方法，权为 $8^i$ 。如： $123_8$ 、 $-456_8$ 、 $10007_8$ 、 $17777_8$ 都是合法的八进制数。

八进制序列 $(17777)_8$ 按位权展开，可以表示成如下形式：

$$(17777)_8 = 1 \times 8^5 + 7 \times 8^4 + 7 \times 8^3 + 7 \times 8^2 + 7 \times 8^1 + 7 \times 8^0$$

十进制、二进制、八进制与十六进制的特征对照表如表1-1所示。

表1-1 二进制、八进制、十进制与十六进制的特征对照表

进 制	数 码	进 位 规 则	数的地址表示法
十进制	0、1、2、3、4、5、6、7、8、9	满十进一	$(4096)_{10}$
二进制	0、1	满二进一	$(1011)_2$
八进制	0、1、2、3、4、5、6、7	满八进一	$(6777)_8$
十六进制	0~9、A、B、C、D、E、F	满十六进一	$(AF8E)_{16}$

## 1.2.2 非十进制数和十进制数的转换

非十进制数转换成十进制数采用“按权相加”法，即将非十进制数按位权写成加权系数展开式，然后按十进制加法规则进行求和。

【例1-1】 将 $(1011.01)_2$ 、 $(FD5)_{16}$ 和 $(475)_8$ 转换成十进制数。

解： $(1011.01)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = (11.25)_{10}$

$$(FD5)_{16} = 15 \times 16^2 + 14 \times 16^1 + 5 \times 16^0 = (4069)_{10}$$

$$(475)_8 = 4 \times 8^2 + 7 \times 8^1 + 5 \times 8^0 = (317)_{10}$$

十进制数转换成非十进制数时，由于整数部分与小数部分转换算法不同，需要分别进行。整数部分采用“除基取余”法，直至商为0，先得到的余数为低位，后得到的余数为高位。小数部分采用“乘基取整”法，直至乘积为整数或达到控制精度。

【例1-2】 将十进制数100和0.625分别转换成二进制数、八进制数与十六进制数。



解:

$$\begin{array}{r}
 2 \overline{) 100} \quad \dots 0 \\
 \underline{2 \quad 50} \quad \dots 0 \\
 2 \overline{) 25} \quad \dots 0 \\
 \underline{2 \quad 12} \quad \dots 1 \\
 2 \overline{) 6} \quad \dots 0 \\
 \underline{2 \quad 3} \quad \dots 0 \\
 2 \overline{) 1} \quad \dots 1 \\
 \underline{0} \quad \dots 1
 \end{array}
 \qquad
 \begin{array}{r}
 8 \overline{) 100} \quad \dots 4 \\
 \underline{8 \quad 12} \quad \dots 4 \\
 8 \overline{) 1} \quad \dots 4 \\
 \underline{0} \quad \dots 1
 \end{array}
 \qquad
 \begin{array}{r}
 16 \overline{) 100} \quad \dots 4 \\
 \underline{16 \quad .6} \quad \dots 4 \\
 \quad \quad \underline{0} \quad \dots 6
 \end{array}$$

所以  $(100)_{10} = (1100100)_2$ ,  $(100)_{10} = (144)_8$ ,  $(100)_{10} = (64)_{16}$ 

$$\begin{array}{r}
 0.625 \\
 \times 2 \\
 \hline
 \underline{1.250} \quad \dots 1 \\
 \times 2 \\
 \hline
 \underline{0.500} \quad \dots 0 \\
 \times 2 \\
 \hline
 \underline{1.000} \quad \dots 1
 \end{array}
 \qquad
 \begin{array}{r}
 0.625 \\
 \times 8 \\
 \hline
 \underline{5.000} \quad \dots 5
 \end{array}
 \qquad
 \begin{array}{r}
 0.625 \\
 \times 16 \\
 \hline
 \underline{10.000} \quad \dots A
 \end{array}$$

同样,  $(0.625)_{10} = (0.101)_2$ ,  $(0.625)_{10} = (0.5)_8$ ,  $(0.625)_{10} = (0.A)_{16}$ 

### 1.2.3 二进制数、八进制数和十六进制数的转换

八进制与十六进制之所以流行不仅在于它们易于在程序中更紧凑地表示数值, 还在于八进制与十六进制的基数恰好是 2 的幂, 所以这两种数制与二进制之间的转换十分容易, 只需要记住几条简单的规则, 如表 1-2 所示, 就能实现二进制数与八进制数、十六进制数之间的快速转换。

表 1-2

常用数制对照表

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0000	0	0	9	1001	11	9
1	0001	1	1	10	1010	12	A
2	0010	2	2	11	1011	13	B
3	0011	3	3	12	1100	14	C
4	0100	4	4	13	1101	15	D
5	0101	5	5	14	1110	16	E
6	0110	6	6	15	1111	17	F
7	0111	7	7	16	10000	20	10
8	1000	10	8	17	10001	21	11

#### 1. 二进制数与八进制数的相互换算

二进制数转换成八进制数采用“三合一”法, 即将 3 位二进制数转换成 1 位八进制数。具体转换过程是: 将二进制数从小数点位置向左和向右按 3 位一组进行划分, 向左不足 3 位按在数的最左侧补零的方法处理, 向右不足 3 位按在数的最右侧补零的方法处理, 然后按照“常用数制对照表”将 3 位一组的二进制数分别转换成相应的八进制数。

【例 1-3】将二进制数  $(1010101111.110101)_2$  转换成八进制数。

解:  $\begin{array}{ccccccc} 001 & 010 & 101 & 111 & . & 110 & 101 & \text{二进制数} \\ 1 & 2 & 5 & 7 & . & 6 & 5 & \text{八进制数} \end{array}$

所以  $(1010101111.110101)_2 = (1257.65)_8$