

C语言 程序设计

第三版

陈文字 黄迪明 侯孟书 霍永青 熊志斌 编著

C YUYAN CHENGXU
SHEJI



电子科技大学出版社

C语言

程序设计 第三版

陈文字 黄迪明 侯孟书 霍永青 熊志斌 编者

C YUYAN CHENGXU
SHEJI

常州大学图书馆
藏书章



电子科技大学出版社

图书在版编目 (CIP) 数据

C 语言程序设计 / 陈文字等编著. —3 版. —成都:

电子科技大学出版社, 2011. 5

ISBN 978-7-5647-0826-9

I. ①C… II. ①陈… III. ①C 语言—程序设计 IV.

①TP312

中国版本图书馆 CIP 数据核字 (2011) 第 078664 号

内 容 摘 要

本书详细介绍 C 语言及其程序设计方法。全书共 10 章, 主要内容包括: C 语言程序设计基础; 基本数据类型与运算; 控制语句; 数组与结构; 指针; 函数; 变量的存储类型; 位域、联合、枚举及定义类型; 输入输出及文件管理。此外, 本书还为读者介绍了 Turbo C 及 Visual C++ 编译系统的使用及标准库函数在动态内存分配、BIOS 接口及 DOS 系统调用和声音程序中的各类实例。

本书为读者展示了 C 语言灵活、精致的编程方法和在工程、科研中的应用, 力求做到 C 语言知识和应用开发能力的融会贯通。为了帮助读者学习, 每章设有小结和习题, 并配有程序设计题解与上机指导辅助教材。

本书是高等学校程序设计方法的入门教材, 亦可作为学习 C 语言人员的自学教材。

C 语言程序设计

(第三版)

陈文字 黄迪明 侯孟书 霍永青 熊志斌 编著

出 版: 电子科技大学出版社 (成都市一环路东一段 159 号电子信息产业大厦 邮编: 610051)

策划编辑: 吴艳玲

责任编辑: 吴艳玲

主 页: www.uestcp.com.cn

电子邮箱: uestcp@uestcp.com.cn

发 行: 新华书店经销

印 刷: 成都蜀通印务有限责任公司

成品尺寸: 185mm×260mm 印张 26 字数 650 千字

版 次: 2011 年 5 月第三版

印 次: 2011 年 5 月第三次印刷

书 号: ISBN 978-7-5647-0826-9

定 价: 46.00 元

■ 版权所有 侵权必究 ■

◆ 本社发行部电话: 028-83202463; 本社邮购电话: 028-83208003。

◆ 本书如有缺页、破损、装订错误, 请寄回印刷厂调换。

前 言

C 语言是一种通用的程序设计语言，能够有效地用来编制各种系统软件和应用软件，是当今最为流行的计算机编程语言之一。

本教材以美国国家标准 C 语言 (ANSI C) 为基本内容，以当前广为使用的 Visual C++6.0 编译系统为实现的版本，全面系统地介绍了 C 语言及其程序设计方法。本书的第一版是国家“九五”电子信息类规划教材，第二版是国家“十一五”规划教材。全书共 10 章，主要内容包括：C 语言程序设计基础；基本数据类型与运算；控制语句；数组与结构；指针；函数；变量的存储类型；位域、联合、枚举及定义类型；输入输出及文件管理。本书为读者展示了 C 语言灵活、精致的编程方法和在工程、科研中的应用，力求做到 C 语言知识和应用开发能力的融会贯通。

本教材是作为程序设计的入门教材而编写的，如果读者对第 1 章中的第一节和第二节内容已经了解，可直接阅读后面的内容。本教材的参考学时数为 68 学时（含上机 20 学时），书中标注“*”的内容，可根据教学实际情况进行取舍，既可作为基本教学内容的扩展，亦可作为自学内容。此外，本书在各章节重要知识点添加了良好编程习惯和编程错误提示，以便初学者能借鉴从而提高学习效率和编程能力。

为了帮助读者更好地理解 C 语言，提高读者开发应用程序的能力，本教材以典型案例图书管理系统应用程序贯穿各章内容，使读者循序渐进地学习和掌握 C 语言开发应用程序的方法与技巧。

为了帮助读者学习，每章设有小结和习题，并配有程序设计题解与上机指导辅助教材，介绍了编译系统的使用方法，使学生在课堂学习过程中能迅速掌握 C 语言程序的编制、编译、调试和运行方法。

在第二版基础上，还增加了语言的本质、数据在内存中的存储方式、控制结构的抽象性和参数传递方式等内容。

谨向本教材的第二版编著者黄迪明、许家珩和胡德昆和对参与编写第二版的阿都建华、李玉柏、杨国炜、张建中、杜海涛、王波、刘家芬、陈琼、张大愚、邹波、曾烨表示感谢。

本教材（第三版）由陈文字、侯孟书、霍永青、熊志斌编写。黄迪明教授审阅了全书。侯孟书编写第 1~3 章，陈文字编写第 4~6 章，霍永青编写第 7~8 章；熊志斌编写第 9~10 章。感谢电子科技大学出版社吴艳玲为本书的出版做了大量的工作。本书在编写过程中，还得到了刘峤、曾茹、赵艳丽、李维顺、郭凌立、李文、韩东普、姚为光、谢文治等的热情帮助。在此对他们及所有为本书的出版付出了辛勤劳动的同志表示衷心的感谢。

由于编者水平有限，书中难免存在缺点和错误，殷切希望广大读者批评指正。

编者

2011 年 3 月于电子科技大学

目 录

第1章 C语言程序设计基础知识.....	1
1.1 计算机基础知识概述.....	1
1.1.1 计算机与信息社会.....	1
1.1.2 计算机中信息的表示.....	4
1.1.3 计算机系统的组成.....	9
1.2 软件开发过程.....	12
1.2.1 计算机求解问题的步骤.....	12
1.2.2 算法的表示.....	16
1.3 C语言概述.....	20
1.3.1 C语言简史及特点.....	20
1.3.2 C程序基本结构.....	21
1.3.3 基本语法单位.....	24
1.4 C语言程序的编写和运行.....	26
1.4.1 C程序的编写和运行步骤.....	26
1.4.2 Visual C++ 6.0 介绍.....	28
1.5 案例研究.....	31
1.6 常见的问题.....	33
小结一.....	34
习题一.....	35
第2章 基本数据类型及运算.....	36
2.1 基本数据类型.....	36
2.1.1 整型.....	37
2.1.2 浮点型.....	38
2.1.3 字符型.....	38
2.2 常量.....	38
2.2.1 整型常量.....	39
2.2.2 浮点型常量.....	39
2.2.3 字符型常量.....	40
2.2.4 字符串常量.....	41
2.2.5 符号常量.....	42

2.2.6 常量的存储	43
2.3 变量	43
2.3.1 变量的定义	43
2.3.2 变量的初始化	44
2.3.3 变量地址	44
2.3.4 变量的存储	45
2.4 运算符与表达式	46
2.4.1 算术运算符和算术表达式	46
2.4.2 赋值运算符和赋值表达式	49
2.4.3 关系运算符和关系表达式	52
2.4.4 逻辑运算符和逻辑表达式	55
2.4.5 位运算符和位表达式	57
2.4.6 条件运算符和条件表达式	61
2.4.7 逗号运算符和逗号表达式	63
2.4.8 其他运算符	65
2.5 混合运算与类型转换	66
2.5.1 自动类型转换	66
2.5.2 强制类型转换	67
2.5.3 赋值运算中的类型转换	68
2.6 运算的优先级与结合性	71
2.6.1 运算符汇总	71
2.6.2 运算符嵌套	72
2.6.3 表达式的运算顺序	73
2.7 数据的输入/输出	74
2.7.1 字符输出函数 putchar 和格式输出函数 printf	74
2.7.2 字符输入函数 getchar 和格式输入函数 scanf	81
2.8 案例研究	85
小结二	86
习题二	87
第 3 章 控制语句	92
3.1 程序的三种基本结构	92
3.2 复合语句	94
3.3 if 条件分支语句	94
3.3.1 if 流程 (单选控制结构)	94
3.3.2 if else 流程 (二选一控制结构)	96
3.3.3 else if 流程 (多选一控制结构)	98
3.3.4 if 语句嵌套	100
3.4 switch 多路开关语句	102

3.5 for 循环（直到型循环）	107
3.6 while 循环和 do while 循环（当型循环）	112
3.6.1 while 语句	112
3.6.2 do while 语句	118
3.7 循环嵌套	121
3.8 break, continue 和 goto 语句	122
3.8.1 break 语句	122
3.8.2 continue 语句	123
3.8.3 goto 语句	125
3.9 控制结构分析	126
3.10 案例研究	128
小结三	133
习题三	134
第 4 章 数组和结构	136
4.1 一维数组	136
4.1.1 一维数组的定义	137
4.1.2 一维数组元素的引用	138
4.1.3 一维数组的初始化	139
4.1.4 一维数组程序举例	140
4.2 二维数组	145
4.2.1 二维数组的定义	146
4.2.2 二维数组元素的引用	146
4.2.3 二维数组的初始化	147
4.3 字符数组	148
4.3.1 字符数组的定义和初始化	149
4.3.2 字符数组的输入/输出	150
4.3.3 与字符串处理有关的几个函数	152
4.3.4 字符串应用举例	157
4.4 结构及结构变量	159
4.4.1 结构及结构变量的定义	159
4.4.2 结构成员的访问	162
4.4.3 结构变量的初始化	163
4.5 结构数组	164
4.6 程序举例	166
4.7 案例研究	175
小结四	177
习题四	178

第 5 章 指针	182
5.1 指针的概念和定义	182
5.1.1 指针的概念	182
5.1.2 指针的定义	183
5.1.3 指针的赋值	183
5.2 指针运算	187
5.3 指针和数组	191
5.3.1 指针与一维数组	191
5.3.2 指针与结构 (数组)	195
5.4 字符串指针	199
5.4.1 指向字符数组的指针	199
5.4.2 指向字符串常量的指针	202
5.4.3 字符指针和字符数组	204
5.5 指针数组	205
5.6 指向指针的指针	208
5.7 程序举例	210
5.8 案例研究	216
小结五	219
习题五	220
第 6 章 函数	224
一、模块化程序设计方法	224
二、函数的分类	225
三、主函数	225
6.1 函数定义和调用	226
6.1.1 函数定义	226
6.1.2 函数调用	229
6.1.3 函数的数据存储区	231
6.2 函数参数传递	232
6.2.1 传数据值	233
6.2.2 传地址值	234
6.3 函数与数组	236
6.3.1 传数据值 (数组元素作函数实参)	236
6.3.2 传地址值 (形参指针指向实参数组)	236
6.3.3 传地址 (函数的形参和实参都是数组)	237
6.4 函数与指针	241
6.4.1 返回指针的函数	241
*6.4.2 指向函数的指针	242

6.5 函数与结构	244
6.5.1 结构指针及结构变量的传地址值调用	245
6.5.2 结构型函数	246
6.5.3 结构指针型函数	247
6.6 递归函数	248
*6.7 命令行参数	253
6.8 标准库函数	255
6.9 程序举例	257
6.10 案例研究	262
小结六	265
习题六	266
第7章 变量的存储类型	268
7.1 C 程序的结构	268
7.1.1 C 程序的组成	268
7.1.2 变量的作用域和生存(命)期	269
7.1.3 变量的存储类型	269
7.2 内部变量	269
7.3 外部变量	271
7.3.1 在同一个源程序文件中使用外部变量	271
7.3.2 在不同源程序文件中使用外部变量	274
7.4 静态变量	276
7.4.1 静态局部变量	276
7.4.2 静态全局变量	278
7.5 寄存器变量	278
7.6 变量的初始化	279
7.7 动态内存分配函数	280
7.8 预处理功能	282
7.8.1 宏定义——#define	282
7.8.2 文件包含——#include	288
7.8.3 条件编译——#if、#ifdef、#ifndef	289
7.9 程序举例	292
小结七	295
习题七	296
第8章 位域、联合、枚举和定义类型名	299
8.1 位域及结构嵌套	299
8.1.1 位域	299
8.1.2 结构嵌套	302

8.2 联合	303
8.3 枚举	309
8.4 定义类型新名字——typedef	311
小结八	313
习题八	313
第 9 章 输入、输出及文件管理	315
9.1 流和文件	315
9.2 控制台 I/O	316
9.2.1 字符输入输出函数——getchar、putchar	316
9.2.2 字符串输入输出函数——gets、puts	318
9.3 文件	319
9.3.1 打开文件函数——fopen	320
9.3.2 关闭文件函数——fclose	321
9.3.3 标准流式文件 stdin、stdout 和 stderr	321
9.4 用于文件的输入输出函数	322
9.4.1 文件单字符输入输出函数——getc、putc	323
9.4.2 文件行输入输出函数——fgets、fputs	325
9.4.3 文件数据块输入输出函数——fread、fwrite	326
9.4.4 流式文件数据格式化输入输出函数——fscanf、fprintf	328
9.4.5 文件随机访问函数——fseek	328
9.5 程序举例	329
9.6 案例研究	332
小结九	340
习题九	341
第 10 章 C 语言高级程序应用	343
*10.1 链表	343
10.1.1 引用自身的结构	343
10.1.2 单向链表	344
10.1.3 双向链表	346
10.1.4 循环链表	348
10.1.5 链表应用程序举例	349
*10.2 与系统有关的库函数	351
10.2.1 BIOS 接口调用函数	353
10.2.2 DOS 系统调用函数	354
10.2.3 案例研究	359
10.3 声音程序	364
10.3.1 声音函数	364

10.3.2 音乐	364
10.3.3 应用举例	365
10.4 案例研究	367
附录	393
附录 A ANSI C 语言的关键字	393
附录 B 运算符的优先级与结合性 (见表 B-1)	393
附录 C 常用字符 ASCII 表 (见表 C-1)	394
附录 D C 语言中常用库函数	395
参考文献	402

第1章 C语言程序设计基础知识

C语言是一种通用的高级程序设计语言。随着C语言在系统软件和应用软件中的广泛应用，已经成为当今世界上最流行的程序设计语言之一。

本章简要介绍程序设计和C语言相关的基础知识。

1.1 计算机基础知识概述

1.1.1 计算机与信息社会

电子计算机的出现和发展是近代科学技术的最伟大成就之一。从1946年第一台电子计算机问世以来，计算机的发展取得了令人瞩目的成就。今天，计算机科学与技术已作为一门先进的学科独立存在；计算机工业已成为改造传统工业、振兴国民经济的重要支柱产业；计算机在科学研究、工农业生产、国防建设以及社会各个领域的广泛应用已成为国家现代化的一个重要标志。

人类在改造客观世界的过程中，已经认识到文字、物质材料和能源是构成世界的三大要素。在人类社会文明的发展过程中离不开信息交流，计算机作为信息处理工具，在信息存储、处理和交流传播方面起着重要的作用。人类历史上曾经历了五次信息革命。第一次信息革命是语言的使用；第二次信息革命是文字的使用；第三次信息革命是印刷术的发明；第四次信息革命是电话、广播电视的使用。而从20世纪60年代开始的第五次信息革命产生的社会技术——信息技术，则是计算机、通信与控制技术相结合的技术，它标志着人类已经迈向信息社会。

文化是人类在社会历史发展中所创造的物质财富和精神财富的总和。可以认为，文化离不开语言。计算机技术已经创造并且还在继续创造出不同于传统自然语言的计算机语言。计算机语言已从简单的应用发展到了多种复杂的应用，并逐步发展到能像传统自然语言一样地表达和传递信息。可以说，计算机技术引起了语言的重构。同时，一个社会的文化模式是以它的记忆为基础的。数据库和网络技术的诞生使知识和信息的存储，在数量上与性质上都发生了质的变化，人们获得知识的方式也因此而发生了变化。文字的出现曾改变了人类历史的进程和文明的面貌，而数据库和网络技术的出现，则从根本上改变了静态的信息存储方式和局部的信息交换方式，人类开始进入了信息社会。

计算机技术使语言、知识及它们间的相互交流发生了根本性的变化，因此引起了思维概念和推理的改变。在1981年召开的第三次世界计算机教育会议上，第一次提出了计算机文

化 (Computer Literacy) 的术语。即为了区别传统的人类文化, 将人类具备的对自然语言的阅读和写作能力称为“第一文化”, 将人类具备的对计算机语言的阅读和编程能力称为“第二文化”, 也称为“计算机文化”。在当今社会, 掌握计算机知识, 提高应用计算机能力应当成为对人才素质最基本的要求。

计算机在信息社会中具有重要的地位。从广义上说, 在信息社会中, 计算机是一种能以高速、精确和可靠的方式进行信息处理的工具。计算机技术对信息的产生、收集、处理、存储和传播将发挥越来越重要的作用, 计算机作为一种崭新的生产力将推动信息社会更快地向前发展。

计算机是一种在事先存入的程序控制下, 能够接收数据、存储数据、处理数据和提供处理结果的电子设备。图 1.1.1 描述了计算机工作流程的简图, 其中包括: 输入、处理、存储和输出四个步骤。

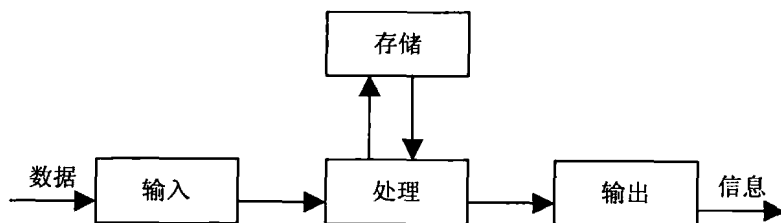


图 1.1.1 计算机工作流程

输入计算机的数据通常是指未经处理的原始数字、字符、图像或声音, 经输入设备转换为二进制形式的数据, 在计算机中预先存入的程序 (指令的集合) 控制下, 按预定方式被处理、加工成有意义或有用的数据。通常将经计算机处理形成的有用数据称为信息。信息可以按用户要求的方式存储, 或在输出设备中转换成用户可识别的报表或图表, 或在显示屏上显示等。

一台计算机由硬件系统和软件系统两大部分组成, 硬件是物质基础, 而软件是计算机的灵魂。所有的软件, 都是用计算机程序设计语言 (programming language) 编写的。程序是指计算机可以直接或间接执行的指令的集合。程序设计语言是一组用来定义程序的规则。这些规则包括:

- 1) 字母表的定义;
- 2) 词法规则: 单词符号的形成规则 (一个单词对应一条形成规则, 规定了该单词由哪些字母按照什么顺序进行排列);
- 3) 语法规则: 语法单位的形成规则 (语法单位包括: 表达式、语句、模块、程序);
- 4) 语义规则: 单词符号和语法单位的含义规则;
- 5) 语用规则: 语义规则的发展和延伸, 强调在一定的语境中使用单词和语法单位时体现出来的具体意义, 要根据上下文 (前、后内容) 明确单词和语法单位的具体意义。
- 6) 其他规则: 包括类型使用规则、参数传递规则、作用域规则等。

计算机程序设计语言的发展, 经历了从机器语言、汇编语言到高级语言的历程。

1. 机器语言

计算机只能理解二进制代码。一组有意义的二进制代码称为一条指令, 包括操作码和

地址码两个部分。二进制代码对应的程序设计语言，称为机器语言。对于指令的操作码与功能、指令格式、寻址方式、数据格式等，不同的计算机有不同的规定。多条二进制指令的集合，称为机器语言程序。

机器语言是第一代计算机语言。机器语言的字母表为 {0, 1}；单词符号就只有二进制串（但可以表示操作码、地址码）；语法单位包括机器指令和程序；语法规则规定了指令和程序的形成，语义规则规定了指令和程序的含义。

利用机器语言进行程序设计是十分繁琐的工作。机器语言程序（全由 0、1 构成）不便阅读和编写，还容易出错（查找错误也十分困难）。由于每种计算机的指令系统往往各不相同，所以，在一台计算机上执行的程序，不能够在另一种计算机上执行，必须另编程序，造成可移植性差，不可避免重复工作。但由于使用的是针对特定型号计算机的语言，故而运算效率比较高。

2. 汇编语言

为了克服机器指令难读、难编、难记和易出错的缺点，人们用与代码指令实际含义相近的英文缩写词、字母和数字等符号取代指令中的二进制代码（包括操作码和地址码），例如，用 ADD 代表加法，用 B 代表存储数据的某个单元的地址（就如电子科技大学可以代表成都市高新西区西源大道 2006 号）等，这样，人们能较容易读懂并理解程序，使得纠错及维护变得方便，这种程序设计语言称为汇编语言，即第二代计算机语言。然而计算机是不能够理解这些符号的，需要将这些符号翻译成计算机能够理解的二进制串，此翻译过程称为汇编（完成汇编的程序称为汇编程序）。

汇编语言字母表主要增加了英文字母；单词符号包括：操作码、内存符号、寄存器、数据；语法单位是汇编指令和程序。

汇编语言仍然是面向机器的语言，使用起来还是比较繁琐，通用性也差。但是，用汇编语言编写的程序，其目标程序占用内存空间少，运行速度快，有着高级语言不可替代的用途。

机器语言和汇编语言都是面向实际计算机的语言，统称为低级语言。

3. 高级语言

程序设计语言对机器的过分依赖，要求使用者必须对硬件结构及其工作原理都十分熟悉，这对非计算机专业人员是难以做到的，对于计算机的推广应用不利。计算机事业的发展促使人们寻求与人类自然语言相接近且能为计算机所接受的通用易学的计算机语言。这种与自然语言相近并被计算机接受和执行的计算机语言称为高级语言。高级语言是面向用户的语言。无论何种机型的计算机只要配备上相应的高级语言的编译或解释程序，则用该高级语言编写的程序就可以运行。

1954 年，第一个完全脱离机器硬件的高级语言 FORTRAN 问世了，近六十多年来，共有上千种高级语言出现，影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、PASCAL、C、PROLOG、Ada、C++、VC、VB、DELPHI、JAVA 等。

高级语言一般包含有四种成分：

数据成分：描述程序所涉及的数据；

运算成分：描述运算；

控制成分：表达程序的控制结构；

传输成分：表达数据的传输（输入输出、文件读写操作）。

高级语言的程序主要是描述计算机的解题过程，即描述复杂的加工处理过程，所以也称高级语言为面向过程语言。

高级程序设计语言的字母表主要包括英文字母、数字字符、运算符号、分界符号等；单词符号包括：关键字、标识符、常量、运算符和分界符；语法单位包括表达式、语句、模块和程序。具有较复杂的语法规则和语义规则。

计算机并不能直接地执行使用高级语言编写的源程序，源程序在输入计算机时，通过“翻译程序”翻译成机器语言形式的目标程序，计算机才能识别和执行。这种“翻译”通常有两种方式，即编译方式和解释方式。

编译方式是指将程序的源代码“翻译”成目标代码（机器指令），目标代码可以脱离其语言环境独立执行，效率较高。但源代码一旦进行修改，需要再重新编译生成新的目标文件才能执行。

解释方式是应用程序源代码一边由相应语言的解释器“翻译”成目标代码（机器语言），一边执行，因此效率比较低，而且不能生成可独立执行的可执行文件，应用程序不能脱离其解释器，但这种方式比较灵活，可以动态地调整、修改应用程序。

高级语言的发展经历了从早期语言到结构化程序设计语言，从面向过程到非过程化程序语言的过程。20 世纪 60 年代中后期，软件越来越多，规模越来越大，而软件的生产基本上是各自为阵，缺乏科学规范的系统规划与测试、评估标准，大批耗费巨资建立起来的软件系统，由于含有错误而无法使用；人们感觉软件越来越不可靠，几乎没有不出错的软件，历史上称这段时期为“软件危机”时期。人们认识到：大型程序的编制不同于编写小程序，它应该是一项新的技术，应该像处理工程一样处理软件研制的全过程。1969 年，提出了结构化程序设计方法，1970 年，第一个结构化程序设计语言 PASCAL 语言出现，标志着结构化程序设计时期的开始。20 世纪 80 年代，出现了面向对象的程序设计语言；直到今天，程序设计语言还在不断地发展。

1.1.2 计算机中信息的表示

计算机是对数据进行处理的机器。由于在计算机内部采用二进制数系统，所以无论何种类型的信息都必须以二进制数的形式在计算机中进行处理。

1. 进位计数制

在日常生活和工作中，人们在计数时使用不同的记号和命名数字的方法，从而构成各种计数制。每一种计数制都使用一组特定的数字符号，将这些符号按序排列，可以表示一个数值，每一种计数制都规定了进位的原则，这种计数方法称为进位计数制。人们最常用的是十进制。在计算机科学中除十进制外，常用的还有二进制、八进制和十六进制。

在采用进位计数的数字系统中，如果用 r 个基本符号（例如 $0, 1, 2, \dots, r-1$ ）表示数值，则称其为 r 进制数， r 称为该数制的基。例如：十进制的基 r 为 10，基本符号为 $0, 1, 2, \dots, 9$ 。若取基 $r=2$ ，则基本符号为 0 和 1，称为二进制数。不同的计数制具有的共同特点是：

- (1) 每一种计数制都有固定的符号集。如十进制数制，基本符号有 10 个： $0, 1, 2, \dots$,

9. 四进制数制, 基本符号有 4 个: 0, 1, 2, 3。

(2) 使用位置表示法。处于不同位置的符号代表的值不同; 其值与所在位置的权值有关。

例如, 十进制数 123.45 可表示为

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

各种进位计数制中的权值恰好是基数的 i 次幂, 其中 $i \in (-\infty, +\infty)$, i 的大小与该位在数中的位置有关。

对任何一种进位计数制 r 表示的数都可以写成按其位权展开的多项式。任意一个 r 进制数 N 可表示为

$$N = \sum D_m r^i$$

其中, D_m 为该数制采用的基本数符, r^i 是权值, r 是基数, 基数不同所表示的进位数制不同。计算机中常用的进位数制如表 1.1.1 所示。

表 1.1.1 计算机中常用的几种进位数制的表示

进位制	二进制	八进制	十进制	十六进制
规则	逢二进一	逢八进一	逢十进一	逢十六进一
基数 r	2	8	10	16
数符	0, 1	0, 1, ..., 7	0, 1, ..., 9	0, 1, ..., 9, A, ..., F
权	2^i	8^i	10^i	16^i
表示形式	B	O	D	H

计算机采用二进制, 而不采用十进制的主要原因是:

(1) 二进制只使用数字符号 0 和 1, 可用自然界存在的两种对立的物理状态表示。例如, 晶体管导通为 1, 截止为 0; 高电压为 1, 低电压为 0; 灯亮为 1, 灯熄为 0; 磁性器件磁化在一个方向为 1, 另一个方向为 0; 等等。计算机采用具有两种不同稳定状态的电子或磁性器件表示 0 和 1。二进制状态简单, 比十进制容易实现, 数据传送也不易出错。

(2) 二进制的运算比十进制数简单。二进制两个整数的“加”的运算规则只有三条:

加法 $0 + 0 = 0$

$0 + 1 = 1$ $1 + 0 = 1$

$1 + 1 = 10$

这样大大简化了计算机中实现运算的线路。实际上, 计算机的减法、乘法及除法运算都可转化为加法这种最基本的运算来完成。

(3) 采用二进制可以进行逻辑运算, 使逻辑代数、电路成为计算机电路设计的基础。

2. 不同进制之间的转换

(1) r 进制转换为十进制: 采用权展开公式实现:

$$N = \sum D_m r^i$$

将 r 进制数转换为十进制数的方法是: 将 r 进制数的 (非 0) 数符与相应位置的权值相乘, 然后相加即可。例如, 将二进制数转换为相应的十进制数, 只要将二进制数中出现 1

的位权相加即可。

例 1 将二进制数 100110.101 转换成相应的十进制数。

$$(100110.101)_B = 1 \times 2^5 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^{-1} + 1 \times 2^{-3} = (38.625)_D$$

例 2 将八进制数 103.2 转换成相应十进制数。

$$(103.2)_O = 1 \times 8^2 + 3 \times 8^0 + 2 \times 8^{-1} = (67.25)_D$$

(2) 十进制数转换为 r 进制

① 十进制数转换为 r 进制整数。将一个十进制整数反复除以基数 r，直到商为 0 为止，并记下每次所得余数（注：余数始终是介于 0 到 r-1 之间的数，包括 0 和 r-1），将最后一个余数到第一余数按从左到右的次序连起来，它们所组成的数字串即为相应的 r 进制整数。这种方法称为除 r 取余法。例如，将十进制整数转换成相应的二进制整数，只需将十进制整数反复除以 2，直到商为 0 为止。

例如，将十进制数 14 转换成二进制数：

2		14		余数
2		7		0 第一个
2		3		1
2		1		1
		0		1 最后一个

即 $(14)_D = (1110)_B$

② 十进制小数转换为 r 进制小数。将十进制小数转换成 r 进制小数时，首先将十进制小数反复乘以基数 r，并取其整数部分（注：整数部分始终是介于 0 到 r-1 之间的数，包括 0 和 r-1）。将从第一个整数到最后一个整数按从左到右的顺序连起来，它们所组成的数字串即为相应的 r 进制小数，这种方法称为乘 r 取整法。

例如，将十进制数 0.375 转换成相应的二进制数。

$0.375 \times 2 = 0.75$	<u>0</u>	第一个整数
$0.75 \times 2 = 1.5$	<u>1</u>	
$0.5 \times 2 = 1.0$	<u>1</u>	最后一个整数

即 $(0.375)_D = (0.011)_B$

如果十进制数包含整数和小数两部分，则必须将十进制数整数部分和小数部分分别按除 r 取余数和乘 r 取整数进行转换，然后，再将 r 进制整数和小数部分组合在一起。

例如，将十进制数 14.375 转换成二进制数，只要将上例的整数部分和小数部分组合在一起即可，即 1110.011。

(3) 非十进制数间的转换

两个非十进制数之间的转换方法一般结合上述两种方法进行转换，即先将待转换数据转换为相应的十进制数，然后再将十进制数转换为其他进制数。由于二进制、八进制和十六进制之间存在特殊关系，即 $8=2^3$ ， $16=2^4$ ，因此转换方法就比较容易，如表 1.1.2 所示。