



“十二五”职业教育
国家规划教材
经全国职业教育教材
审定委员会审定

C 语言程序设计

(第4版)

廖雷 主编

高等教育出版社



“十二五”职业教育国家规划教材
经全国职业教育教材审定委员会审定

C YUYAN CHENGXU SHEJI

C 语言程序设计

(第4版)

廖雷 主编

高等教育出版社·北京

内容提要

本书是“十二五”职业教育国家规划教材。

本书是在廖雷主编的普通高等教育“十一五”国家级规划教材《C语言程序设计(第3版)》的基础上修订完成的。本书在全面介绍ANSI C的语言成分和标准库函数的同时,穿插讲解了相应的程序设计技巧、常用算法和具有实用价值的程序实例。

本书力求体现概念准确、编排合理、循序渐进、深入浅出、讲解通俗、便于自学的特色,读者可以不具备其他高级语言和程序设计的基础知识便可阅读本书。本书注重技术应用性,语言与程序设计并重,经典实例和实用程序并重;强化实践环节,提供了配套的实验教材,精选了较多的习题和实习题。

本书适用于高等职业学校、高等专科学校、成人高校及本科院校举办的二级职业技术学院、继续教育学院使用,也可作为计算机培训和等级考试辅导的教学用书,还可作为程序开发人员和自学者的参考书。

图书在版编目(CIP)数据

C语言程序设计 / 廖雷主编. -- 4版. -- 北京: 高等教育出版社, 2015.4

ISBN 978-7-04-042271-9

I. ①C… II. ①廖… III. ①C语言-程序设计-高等职业教育-教材 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第039473号

策划编辑 许兴瑜
插图绘制 黄建英

责任编辑 许兴瑜
责任校对 张小镝

封面设计 张楠
责任印制 张泽业

版式设计 王艳红

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
印 刷 北京丰源印刷厂
开 本 787mm×1092mm 1/16
印 张 15
字 数 360千字
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.landaco.com>
<http://www.landaco.com.cn>
版 次 2000年7月第1版
2015年4月第4版
印 次 2015年4月第1次印刷
定 价 26.00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 42271-00

教材是教学过程的重要载体，加强教材建设是深化职业教育教学改革的有效途径，推进人才培养模式改革的重要条件，也是推动中高职协调发展的基础性工程，对促进现代职业教育体系建设，切实提高职业教育人才培养质量具有十分重要的作用。

为了认真贯彻《教育部关于“十二五”职业教育教材建设的若干意见》（教成〔2012〕9号），2012年12月，教育部职业教育与成人教育司启动了“十二五”职业教育国家规划教材（高等职业教育部分）的选题立项工作。作为全国最大的职业教育教材出版基地，我社按照“统筹规划，优化结构，锤炼精品，鼓励创新”的原则，完成了立项选题的论证遴选与申报工作。在教育部职业教育与成人教育司随后组织的选题评审中，由我社申报的1338种选题被确定为“十二五”职业教育国家规划教材立项选题。现在，这批选题相继完成了编写工作，并由全国职业教育教材审定委员会审定通过后，陆续出版。

这批规划教材中，部分为修订版，其前身多为普通高等教育“十一五”国家级规划教材（高职高专）或普通高等教育“十五”国家级规划教材（高职高专），在高等职业教育教学改革进程中不断吐故纳新，在长期的教学实践中接受检验并修改完善，是“锤炼精品”的基础与传承创新的硕果；部分为新编教材，反映了近年来高职院校教学内容与课程体系改革的成果，并对接新的职业标准和新的产业需求，反映新知识、新技术、新工艺和新方法，具有鲜明的时代特色和职教特色。无论是修订版，还是新编版，我社都将发挥自身在数字化教学资源建设方面的优势，为规划教材开发配备数字化教学资源，实现教材的一体化服务。

这批规划教材立项之时，也是国家职业教育专业教学资源库建设项目及国家精品资源共享课建设项目深入开展之际，而专业、课程、教材之间的紧密联系，无疑为融通教改项目、整合优质资源、打造精品力作奠定了基础。我社作为国家专业教学资源库平台建设和资源运营机构及国家精品开放课程项目实施单位，将建设成果以系列教材的形式成功申报立项，并在审定通过后陆续推出。这两个系列的规划教材，具有作者队伍强大、教改基础深厚、示范效应显著、配套资源丰富、纸质教材与在线资源一体化设计的鲜明特点，将是职业教育信息化条件下，扩展教学手段和范围，推动教学方式方法变革的重要媒介与典型代表。

教学改革无止境，精品教材永追求。我社将在今后一到两年内，集中优势力量，全力以赴，出版好、推广好这批规划教材，力促优质教材进校园、精品资源进课堂，从而更好地服务于高等职业教育教学改革，更好地服务于现代职教体系建设，更好地服务于青年成才。

高等教育出版社

2014年12月

C语言是极具影响力的一门程序设计语言，C++、Java和C#都属于C语言家族，C语言是它们的基础，因此，国内高校的很多专业都将C语言作为第一门程序设计语言课程开设。编者于2000年编写了《C语言程序设计》，由高等教育出版社出版。2009年，已修订至第3版。编写本书时充分考虑了高职高专学生的实际情况，力求使本书具备起点低、概念准确、讲解通俗、深入浅出、注重实践、强化应用、反映最新技术进展等特点。本书出版后反映良好，累计印刷20余万册，具有稳定的读者群。目前，在高等教育出版社的支持下，完成了对第3版的修订工作，第4版被列为“十二五”职业教育国家规划教材。

本书主要用于工科各专业，书中部分内容和实例可根据各专业的实际情况进行取舍。建议学时数为64学时，其中，理论教学46学时，课内上机18学时。有条件的学校最好能再安排18学时的课外上机。另外，建议计算机相关专业再安排一周的课程设计（实训）。

为方便教学，编者还编写了《C语言程序设计习题解答及上机指导（第4版）》，并将制作《C语言程序设计（第4版）》教材的电子教案，供读者选用。

本书由廖雷主编，廖雷、袁璟、陈立共同完成编写。其中，廖雷编写了第1、2、5、7、8、9章，袁璟编写了第3、4章，陈立编写了第6、10章。

感谢南京大学闵建洪教授审阅了书稿，感谢兄弟院校的老师 and 同学在研讨班、讲习班和网上交流中给作者的启迪。

衷心感谢读者选用本书，热诚欢迎广大教师和同学指出本书存在的错误，并提出修改建议。需要电子资源的读者可以发邮件至 1548103297@qq.com 索取，主编的电子邮箱是 liaolei@sina.com。

编 者

2015年2月于南京

近十年来, 计算机技术得到飞速发展, 计算机已在各行各业得到了广泛应用, 对计算机应用人员 (特别是那些非计算机专业出身的技术人员) 的技术要求已不再停留在操作员的层次, 而是随之提出了更高的要求, 即要求他们掌握一定的程序设计技术, 以解决更为复杂和专业的问题。

掌握程序设计的前提是掌握程序设计语言, 在众多的程序设计语言中, C 语言以其灵活性和实用性受到了广大计算机应用人员的喜爱。同时, C 语言是既得到美国国家标准学会 (ANSI) 标准化, 又得到工业界广泛支持的计算机语言之一, 几乎任何一种类型的计算机 (大型机、小型机、工作站、PC)、任何一种操作系统 (Windows、UNIX、Linux、Netware) 都支持 C 语言开发。C 语言在巩固其原有应用领域的同时, 又在拓展新的应用领域, 支持大型数据库开发和 Internet 应用。并且, C 语言本身也拓展了面向对象的语言成分, 发展成为目前最有影响力的面向对象的程序设计语言 C++。C/C++ 语言几乎具备现代程序设计语言的所有语言成分, 一旦掌握了 C/C++ 语言, 就可以较为轻松地学习其他任何一种程序设计语言, 但反过来就未必如此。

本书的读者对象主要是高职高专的工科各专业的学生。根据高职高专的课程设置, C 语言被推荐为工科各专业程序设计语言的必修课程, 其他专业也可酌情选学。同时, 社会上的众多计算机应用人员也对 C 语言有浓厚的兴趣, 然而, 与其他程序设计语言相比, C 语言学习难度要大一些, 因此, 要学好 C 语言, 找到一本合适的教材是使读者能顺利地掌握和运用 C 语言的一个重要因素。为此, 本书根据读者对象的性质, 力图体现以下编写特色。

1. 起点较低, 不需具备程序设计语言基础知识。

很多 C 语言的教材都要求读者先前学过一门程序设计语言。但我们认为, 本书的主要读者是高职高专工科各专业的学生, 他们学制只有三年, 要求他们先学一门其他语言, 总课时就会占得较多, 学生学业负担就会较重。因此, 我们从程序设计的最基础知识讲起, 把一些经典算法的来龙去脉交代清楚, 此时就不一定非要要求读者有其他程序设计语言的基础。

2. 概念准确, 编排合理。

由于历史的原因, 与其他程序设计语言相比, C 语言显得不是很严谨, 因此, 在理解 C 语言时容易产生偏差。为此, 我们认真研读、消化了最具权威性的由 C 语言设计者 Brian W.Kernighan 和 Dennis M.Ritchie 按照 ANSI C 标准所著的《The C Programming Language, 2nd Edition》一书, 以及微软和 Inprise/Borland 公司最新的 C/C++ 语言产品的联机手册, 以此为基础讲解 C 语言。同时, 在内容的编排上, 注意分散难点, 便于读者循序渐进。

3. 详略得当, 重点突出。

为体现高职高专教学中的“必需、够用为度”原则, 本书主要讲解C语言最基本、最常用的内容, 有意回避C语言中出现频率很低或与语言的实现版本有关的内容, 把重点放在语言本身的难点(如指针)和程序设计技巧方面。

4. 深入浅出, 讲解通俗。

根据高职高专教学和社会读者自学的特点, 避免一般本科教材中引出概念、解释概念、举例说明的传统讲解模式, 采用根据知能结构要求提出问题、分析问题、解决问题, 最后总结出概念并推广到一般的方法。同时, 在讲解中强调通俗性。

5. 两个并重, 强化实践, 重视应用。

所谓两个并重, 即程序设计语言和程序设计技巧并重, 经典实例和实用程序并重。本书力求使读者学完C语言后, 不仅能懂C语言的语法、语义, 更重要的是具备编程解决实际问题的能力。读者不仅要了解一些经典实例和经典算法, 还应研究一些来自实际工作和工程实践的实用案例, 并加以借鉴、模仿、改写。同时, 本书重视实践环节, 有专门章节介绍上机编程和程序调试技巧。附录中有中英文出错信息对照表, 并提供了较多的习题和上机实习题。

6. 反映最新技术进展。

当前的程序设计技术, 已从结构化程序设计技术向面向对象程序设计技术过渡。具体地说, 对一个规模较大的应用程序, 总体框架是由面向对象程序设计构搭而成, 而在局部实现时仍需采用结构化程序设计技术。因此, 我们在本书最后一章从实用的角度简要介绍了面向对象的程序设计语言C++。同时我们认为, 面向对象的程序设计可分为两个层次, 第一层次是利用一些开发工具提供的类体系(如微软Visual C++中的MFC、Inprise/Borland C++ Builder中的VCL)进行开发, 第二层次是根据实际应用, 自己构造一个类体系。我们认为, 就目前的技术发展水平, 高职高专工科各专业的学生只需具备第一层次的技能即可。

自从Windows出现以来, 用C语言开发程序的技术人员比例一度比DOS时代低, 究其原因, 无论微软的Visual C++还是Inprise/Borland的Borland C++, 学习难度都较大, 因此, 一部分技术人员改用诸如Visual Basic、Delphi、PowerBuilder等快速应用程序开发工具(RAD)进行开发。而如今, 局面终于开始改观, Inprise/Borland推出的C++ Builder是一个全新的真正可视化编程工具, 它采纳了微软公司Windows GUI的许多先进特性和设计思想, 采用了标准化的面向对象程序设计语言C++。可以说, C++ Builder既有Visual Basic的易用性, 又有C/C++的高效性, 将Windows下的C/C++程序设计带入了大众化时代。因此, 本书在最后一章中简要介绍了C++ Builder 4.0。

本书由廖雷主编, 袁璟、陈立参编。其中, 主编负责拟定全书框架并对参编所写章节仔细阅读, 提出了明确和具体的修改意见。各章具体分工如下: 廖雷编写第1、2、6、8、9、10章, 以及附录2、附录3; 袁璟编写第3、4、5章, 第9章的图形处理部分内容及附录1; 陈立编写第7、11、12章。深圳职业技术学院的余苏宁副教授在百忙之中抽出时间认真细致地审阅了全书, 并提出了宝贵意见, 在此深表感谢, 同时对余教授在审稿过程中体现出的优秀的业务素质和严谨的工作作风表示由衷的钦佩。

本书得以出版，与教育部高教司高职高专处、高等教育出版社的指导和支持，以及作者所在学校及教务处、系、教研室领导和同事的关心及帮助是分不开的。同时，作者在与北京联合大学、深圳职业技术学院、上海第二工业大学、南京动力高等专科学校、承德石油高等专科学校的有关专家的交流中受益匪浅，在此一并致谢。

虽然作者从1988年起使用C语言进行科研和科技开发工作，且有10年的C语言教学经验，但由于水平有限，加之时间仓促，错漏之处在所难免，热诚欢迎广大读者把自己的意见、建议和要求反馈给我们，以便进一步地完善，作者的电子邮箱是 liaolei@sina.com。

作 者

2000年3月于南京

第 1 章 C 语言概述 1	2.4.5 条件表达式和逗号表达式..... 17
1.1 为什么要学习 C 语言 2	2.4.6 类型转换 17
1.1.1 C 语言的历史 2	2.5 变量初始化和赋值语句..... 18
1.1.2 C 语言的特点 3	2.5.1 变量的初始化 18
1.2 C 语言的一个简单实例..... 3	2.5.2 赋值语句 18
1.3 编辑、编译、连接、运行一个 C 语言程序..... 4	2.6 数据输出 18
1.4 学习 C 语言的必备知识..... 4	2.7 数据输入 22
1.4.1 数制 5	2.8 程序实例 24
1.4.2 数制之间的转换..... 5	习题 27
1.4.3 整数的原码、补码、反码表示..... 6	上机实训题 28
习题 7	第 3 章 分支结构 31
第 2 章 数据类型、运算符、表达式、 赋值语句、输入和输出 9	3.1 语句概述 32
2.1 C 语言的词法记号 10	3.2 问题的引出 33
2.1.1 关键词 10	3.3 if 语句 34
2.1.2 标识符 11	3.3.1 if 语句的一般形式 34
2.1.3 分隔符 11	3.3.2 if-else 中的复合语句 35
2.2 数据类型 11	3.3.3 if 语句的省略形式 37
2.2.1 整型 11	3.3.4 较复杂的条件表达式 38
2.2.2 浮点型 12	3.3.5 非关系/逻辑表达式构成的条件 表达式 38
2.2.3 字符型 12	3.3.6 同一个条件的多种表达方式 39
2.3 常量和变量 12	3.4 if 语句的嵌套 39
2.3.1 常量 12	3.4.1 嵌套的引出 39
2.3.2 变量 13	3.4.2 流程图 40
2.4 运算符和表达式 14	3.4.3 if 与 else 的配对规则 41
2.4.1 算术运算符和算术表达式 14	3.4.4 嵌套举例 42
2.4.2 关系运算符和关系表达式 15	3.5 switch 语句 45
2.4.3 逻辑运算符和逻辑表达式 15	习题 51
2.4.4 赋值运算符和赋值表达式 16	上机实训题 54
	第 4 章 循环结构 55

4.1 while 语句	56	上机实训题	105
4.2 do-while 语句	59	第 6 章 数组	107
4.3 for 语句	60	6.1 问题的引出	108
4.3.1 for 语句的一般形式	60	6.2 一维数组	108
4.3.2 for 语句形式的多样性	61	6.2.1 一维数组的说明、引用和存储	108
4.3.3 循环次数确定的情况	62	6.2.2 一维数组的初始化	110
4.3.4 循环次数不确定的情况	64	6.2.3 一维数组的经典实例	111
4.4 循环的嵌套	66	6.3 二维数组	115
4.5 几种循环的比较	67	6.3.1 二维数组的说明、引用和存储	115
4.6 break、continue、goto 语句	68	6.3.2 二维数组的初始化	117
4.6.1 break 语句	68	6.3.3 二维数组的经典实例	117
4.6.2 continue 语句	69	6.4 字符数组与字符串	119
4.6.3 goto 语句	69	6.4.1 字符数组的说明与初始化	119
4.7 程序实例	71	6.4.2 字符串	120
4.8 结构化程序设计	73	6.4.3 字符数组的经典实例	121
习题	74	6.5 数组应用实例	122
上机实训题	78	6.5.1 直接插入排序	122
第 5 章 函数、存储类和预处理程序	79	6.5.2 二分查找	124
5.1 函数	80	6.5.3 用数组求 Fibonacci 数列	126
5.1.1 引言	80	6.5.4 一个数模求解实例	127
5.1.2 函数的定义	81	习题	132
5.1.3 函数调用和参数传递	83	上机实训题	134
5.1.4 函数的说明	85	第 7 章 指针	135
5.1.5 函数的嵌套调用与递归调用	86	7.1 地址和指针	136
5.2 变量的作用域和生命期	89	7.1.1 地址	136
5.2.1 问题的提出	89	7.1.2 指针	136
5.2.2 auto 变量	90	7.2 指针变量	137
5.2.3 extern 变量	92	7.2.1 指针变量的说明	137
5.2.4 static 变量	94	7.2.2 指针变量的运算	138
5.2.5 register 变量	95	7.3 指针作为函数参数	139
5.2.6 存储类别小结	96	7.3.1 问题的提出	139
5.3 预处理程序	98	7.3.2 解决方法	140
5.3.1 文件包含	98	7.3.3 程序实例	141
5.3.2 宏替换	99	7.4 指针与数组	142
5.3.3 条件编译	100	7.4.1 指针与数组的关系	142
习题	101	7.4.2 用指针形参对应数组名实参	144
		7.4.3 程序实例	145

7.5 指针的运算	147	9.2.1 指向结构变量的指针	187
7.6 字符指针	148	9.2.2 用结构指针作为函数参数	188
7.6.1 用字符指针处理字符串	148	9.2.3 链表	189
7.6.2 字符串处理函数	150	9.3 杂类	192
7.6.3 使用字符数组和字符指针处理字符串的区别	152	9.3.1 位运算	192
7.7 指针数组和指向指针的指针	154	9.3.2 位段	194
7.7.1 指针数组	154	9.3.3 联合	194
7.7.2 指向指针的指针	155	9.3.4 枚举	195
7.7.3 指向指针的指针的应用	155	9.3.5 类型定义	196
7.7.4 带形参的 main 函数	158	9.3.6 多文件	196
7.8 指向函数的指针	159	习题	198
7.9 程序实例	161	上机实训题	200
习题	165	第 10 章 文件	203
上机实训题	167	10.1 文件概述	204
第 8 章 常用库函数和软件		10.1.1 文件的概念	204
开发概述	169	10.1.2 文件的分类	204
8.1 常用库函数	170	10.1.3 缓冲文件系统和非缓冲文件系统	205
8.1.1 输入/输出函数	170	10.2 缓冲文件系统	205
8.1.2 数学计算	170	10.2.1 缓冲文件系统的基本概念	205
8.1.3 数据类型测试和转换	171	10.2.2 缓冲文件的打开和关闭	206
8.1.4 其他	173	10.2.3 缓冲文件的读/写	209
8.2 软件开发概述	174	10.2.4 缓冲文件的定位	213
8.2.1 中小规模软件开发步骤	174	10.2.5 综合实例	216
8.2.2 衡量软件质量的几个主要准则	174	10.3 非缓冲文件系统	217
8.3 使用 C 语言时常犯的错误	175	10.3.1 非缓冲文件系统的基本概念	217
8.3.1 编译程序能查出的错误	175	10.3.2 非缓冲文件的建立	218
8.3.2 编译程序不能发现的错误	177	10.3.3 非缓冲文件的打开和关闭	218
8.4 程序实例	178	10.3.4 非缓冲文件的读/写	218
习题	180	10.3.5 非缓冲文件的定位	219
第 9 章 结构和杂类	181	习题	219
9.1 结构	182	上机实训题	220
9.1.1 结构类型的定义	182	附录	221
9.1.2 结构变量的说明和引用	182	附录 1 ASCII 码表	222
9.1.3 结构数组	183	附录 2 C 语言关键词	223
9.1.4 结构与函数	186	附录 3 C 语言运算符	223
9.2 指针在结构中的应用	187	参考文献	224

第1章 C语言概述

第1章 C语言概述

1

第1章

C语言概述

1978年，由贝尔实验室的Bjarne Stroustrup和DM Ritchie（伯克利K&R，即《The C Programming Language》一书）编写了最初的C语言。K&R语言，即C语言，是C语言的前身。C语言是在B语言的基础上，吸收了ALGOL 68和ALGOL W的一些特点而发展起来的。C语言在20世纪80年代中期得到了广泛的应用，成为当时最流行的编程语言之一。C语言在20世纪90年代得到了进一步的发展，出现了C++和C#等新的编程语言。C++是在C语言的基础上，增加了面向对象编程的特性而发展起来的。C#是在C++的基础上，增加了垃圾回收和托管代码等特性而发展起来的。C语言在2001年，微软公司推出了C#语言，这标志着C语言在21世纪得到了进一步的发展。

1.1 为什么要学习C语言

1.1.1 C语言的历史

FORTRAN 语言是历史上的第一门计算机高级语言，它主要用于科学计算。随着 FORTRAN 的出现，越来越多的计算机专家和工程技术人员对高级语言的研究、设计及使用产生了浓厚的兴趣。诞生于 20 世纪 60 年代的 ALGOL 语言是一门结构良好、逻辑严谨、简明易学的算法语言，但由于它的应用面较窄和人们对 FORTRAN 的依恋，使其没有得到推广。20 世纪 70 年代初期出现的 Pascal 语言是第一门反映了结构化程序设计思想的高级语言，它在大学和研究所中流传较广，一度成为国内外计算机（应用）专业学生的入门语言。

在 Pascal 语言诞生的同时，C 语言在美国著名的贝尔实验室中酝酿并诞生了。与 FORTRAN、ALGOL 和 Pascal 语言不同，C 语言诞生之时并没有什么研制报告和语言报告，而是在设计 UNIX 操作系统时不断地得到更新和完善。因此，人们把 C 语言称为程序员设计的语言，而把 FORTRAN、ALGOL 和 Pascal 语言称为计算机科学家设计的语言。

UNIX 的早期版本是用汇编语言编写的，而用 C 语言编写的 UNIX 比原先的版本更易于理解、修改和扩充，更重要的是具有良好的可移植性。作为一个优秀的操作系统，UNIX 在世界范围内得到了广泛的应用，它的设计者因此于 1983 年获得了计算机科学的最高奖——图灵奖。要使用 UNIX，就必须掌握 C 语言，因此，C 语言也为越来越多的人所熟知，人们进一步认识到 C 语言是一门极有生命力的程序设计语言。渐渐地，它已不完全依赖 UNIX，成为程序员的首选语言之一。在国内，很多高校将 C 语言作为第一门程序设计语言课程开设。

1978 年，贝尔实验室的 B.W Kernighan 和 D.M Ritchie（简称 K&R）出版了《The C Programming Language》一书，建立了所谓的 C 语言的 K&R 标准，它一度成为 C 语言的事实标准。目前，美国标准化协会已对 C 语言标准化，称为 ANSI C 标准。

C 语言本身也在发展，20 世纪 80 年代中期，出现了面向对象程序设计的概念。贝尔实验室的 B. Stroustrup 博士借鉴了 Simula 67 中的类的概念，将面向对象的语言成分引入到 C 语言中，设计出了 C++ 语言。C++ 语言赢得了广大程序员的喜爱，不同的机器、不同的操作系统几乎都支持 C++ 语言，如在 PC（个人计算机）上，微软公司先后推出了 MS C++、Visual C++ 等产品；Borland 公司先后推出了 Turbo C++、Borland C++、Borland C++ Builder 等产品。同时，C++ 语言也得到了国际标准化组织（ISO）的认可，为此，国际标准化组织已对 C++ 语言实现标准化。C/C++ 语言对新语言的形成也有较大的影响力。20 世纪 90 年代中期以来，Internet 日益普及，用于 Internet 开发的 Java 语言日益为人们所熟知。事实上，Java 语言与 C++ 语言极为相似，熟悉 C++ 语言的程序员在很短的时间内就能掌握 Java。2002 年，微软公司正式推出了 C# 语言，该语言与 C/C++ 也有密切的联系，它已成为 .NET 环境的重要编程语言。

1.1.2 C 语言的特点

C 语言之所以能成为程序员的首选语言之一，是因为具有如下特点。

(1) 具有现代化程序设计语言的特征

C 语言具有丰富的数据类型、众多的运算符、体现结构化程序设计的优良的控制结构、具备抽象功能及体现信息隐蔽思想的函数。

(2) 用途广泛

C 语言的应用几乎遍及了程序设计的各个领域，如科学计算、系统程序设计、字处理软件和电子表格软件的开发、信息管理、计算机辅助设计、图形图像处理、数据采集、实时控制、嵌入式系统开发、网络通信、Internet 应用、人工智能等方面。

(3) 语言简洁，具备底层处理功能，可执行代码质量高

C 语言简洁，使用其为实现某一功能所写的源程序代码往往比用其他语言写的短，使得程序输入工作量减少。C 语言能直接访问物理地址和端口，并能进行位操作，因此能实现汇编语言的大部分功能。另一方面，由 C 语言生成的可执行代码内存容量少，执行效率高，因此，C 语言有可移植的汇编语言的美称。

(4) 可移植性好

若程序员在书写程序时严格遵循 ANSI C 标准，则其源代码可不做修改，即可用于各种型号的计算机和各种操作系统，因此，C 语言具备良好的可移植性。

1.2 C 语言的一个简单实例

下面是 C 语言的一个简单实例，其功能是从键盘上读入两个整数，并计算这两个数与 6 三者之和，最后将结果输出。

【例 1.1】 C 语言的一个简单实例。

```
#include <stdio.h>
main()
{
    int a,b,sum;
    printf("Enter Two Numbers:");
    scanf("%d%d",&a,&b);
    sum=a+b+6;
    printf("The sum is %d\n",sum);
}
```

程序运行结果如图 1-2-1 所示。

```
Enter Two Numbers:64 28
The sum is 98
```

图 1-2-1 程序运行结果

一个 C 语言程序的执行总是从被称为 main 的主函数处开始的,在例 1.1 中,main 函数中对变量 a、b、sum 做了说明,它们的类型是整型。printf 是一个标准输出函数,因此,main 函数中的第一个 printf 函数输出一行提示信息,即 Enter Two Numbers:,要求用户输入两个整数。scanf 是一个标准输入函数,它完成 a、b 两个变量的输入工作,即从键盘上输入两个数,使得 a、b 分别取值 64 和 28。语句“sum=a+b+6;”计算 a+b+6 的值并将它赋给 sum 变量。第二个 printf 函数将文字 The sum is 和运算结果 98 一起输出。

举这个例子,主要是为了使读者对 C 语言有一个大致的了解,具体细节不必完全弄懂。

1.3 编辑、编译、连接、运行一个 C 语言程序

用 C 语言书写的程序又称为 C 的源程序,它是不能直接运行的。为此,必须生成与之对应的可执行程序。具体过程如下。

① 编辑源程序,完成后将源程序以扩展名.c 存盘。

② 对源程序进行编译,即将源程序转换为扩展名为.obj 的二进制代码,此二进制代码仍不能运行。若源程序有错,则必须予以修改,然后重新编译。

③ 对编译生成的.obj 文件进行连接,即加入库函数和其他二进制代码来生成可执行程序。连接过程中,可能出现未定义的函数等错误,为此,必须修改源程序,然后重新编译和连接。

④ 执行生成的可执行代码,若不能得到正确的结果,必须修改源程序,然后重新编译和连接;若能得到正确结果,则整个编辑、编译、连接、运行过程顺利结束。

在 Turbo C 2.0 中提供了一个集成开发环境,上述过程步骤均能在 Turbo C 2.0 集成环境中完成。有关集成环境的具体使用详见后续章节。

1.4 学习 C 语言的必备知识

学习 C 语言,除学习理论知识外,还必须通过上机编制和调试程序。事实上,不同的机器在不同的操作系统下有不同的 C 语言编译器,本书主要用 Borland 公司的 Turbo C 2.0 讲解 C 语言的实际编程。Turbo C 2.0 是 DOS 下的一个集成开发环境,在 Windows 下需进入 DOS 窗口运行。因此,熟悉常用的 DOS 命令或 Windows 基本操作是学习本课程的前提条件。

学习 C 语言,还必须弄清数据在内存中的表示方法。数据在计算机内部是以二进制来表

示的, 而为了便于问题描述, 还常常用到十六进制和八进制。实际上, 它们都可以认为是二进制的缩写方式。

1.4.1 数制

1. 十进制数

十进制是日常使用的进制, 由 0~9 这 10 个数字组成, 运算规则是逢十进一。

2. 二进制数

二进制数在日常生活中也会出现, 如两只鞋子称为一双, 这就是二进制。二进制的基数是 2, 因此, 在二进制中出现的数字只有两个, 即 0 和 1。

二进制的运算规则是逢二进一, 因此有:

$$0+0=0 \quad 0+1=1 \quad 1+0=1 \quad 1+1=10$$

$$0 \times 0=0 \quad 0 \times 1=0 \quad 1 \times 0=0 \quad 1 \times 1=1$$

3. 八进制数

八进制的基数是 8, 所使用的数字为 0、1、2、3、4、5、6、7。其运算规则是逢八进一。

4. 十六进制数

十六进制的基数是 16, 它使用的数字与字符为 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。其中, A~F 分别代表十进制数的 10~15。其运算规则是逢十六进一。

5. 数制标记方法

为了区分不同进制的数据, 可以用圆括号将数据括起来, 在括号的右下角以数字 2、8、10、16 表示该数代表的进制。例如, $(100)_2$ 、 $(261)_8$ 、 $(192)_{10}$ 、 $(1FA)_{16}$ 分别代表的是二进制的 100、八进制的 261、十进制的 192、十六进制的 1FA。

1.4.2 数制之间的转换

下面通过一些实例来说明不同进制数相互转换的方法。

1. 二进制数转换成十进制数

$$(111.011)_2=(1 \times 2^2+1 \times 2^1+1 \times 2^0+0 \times 2^{-1}+1 \times 2^{-2}+1 \times 2^{-3})_{10}=(7.375)_{10}$$

2. 十六进制数转换成十进制数

$$(2AB.C)_{16}=(2 \times 16^2+10 \times 16^1+11 \times 16^0+12 \times 16^{-1})_{10}=(683.75)_{10}$$

3. 十进制整数转换成二进制数

十进制整数转换成二进制数可采用“除 2 取余法”, 直至商为 0。如将十进制整数 28 转换成二进制数, 如图 1-4-1 所示。

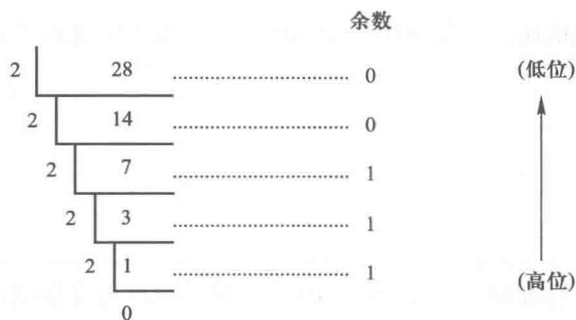


图 1-4-1 十进制整数转换成二进制数

注意，最后得到的余数是高位数，因此转换结果如下：

$$(28)_{10} = (11100)_2$$

由以上实例可知，可以实现任何数制之间的转换。例如，要将二进制数转换成十六进制数，可先将二进制数转换成十进制数，再将十进制数用“除 16 取余法”得到十六进制数。

事实上，二进制、八进制、十六进制数之间的相互转换有许多简便的方法，读者可自行查阅有关书籍。

1.4.3 整数的原码、补码、反码表示

一般，计算机中用 16 位或 32 位来表示整数，位数越多，能表示数的范围就越大。

整数有正负之分，为此，可以用一个二进制位作为符号位，一般总是最高位，当符号位为 0 时表示正数，符号位为 1 时表示负数。例如，当用 16 位来表示一个整数时，可表示如下：

$$(0000000000101011)_2 = (+43)_{10}$$

$$(1000000000101011)_2 = (-43)_{10}$$

上述表示法，称为整数的原码表示法。

整数也可采用反码表示法。对于负整数来说，符号位为 1，但绝对值部分正好与原码相反（即 0 变为 1，1 变为 0）。如：

$$(-43)_{\text{原}} = 1000000000101011$$

$$(-43)_{\text{反}} = 11111111111010100$$

实际上，整数在机器内大多用补码表示。对负整数而言，符号位仍为 1，但绝对值部分却是反码的最低位加 1 得到的结果。如：

$$(-43)_{\text{补}} = 11111111111010101$$

注意，对正整数而言，其原码、反码、补码相同。

关于补码的知识，是学习 C 语言的必备知识。至于实数在计算机内的表示法，可查相关资料。