



全国高等院校信息管理与信息系统专业规划教材

C语言程序设计

张书云 主编 姜淑菊 朱雷 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



全国高等院校信息管理与信息系统专业规划教材

C 语言程序设计

张书云 主编

姜淑菊 朱 雷 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是为将C语言作为入门语言的程序设计课程编写的以培养学生程序设计基本能力为目标的教材。全书共分为9章, 主要内容包括C语言概述、数据类型、常量与变量、数据的运算与输入/输出、C语言3种基本结构程序设计、函数、数组、指针、结构体与共用体、文件系统。

本书体系合理、逻辑性强、文字流畅、通俗易懂。注重培养读者分析问题和程序设计的能力, 注重培养良好的程序设计风格和习惯。本书还配有辅助教材《C语言程序设计习题解答与实验指导》。

本书适合作为高等院校, 特别是财经类院校计算机程序设计课程的教学用书, 还可以供广大C语言程序设计初学者自学参考。

图书在版编目(CIP)数据

C语言程序设计/张书云主编. —北京: 中国铁道出版社, 2008.8

(全国高等院校信息管理与信息系统专业规划教材)

ISBN 978-7-113-08770-8

I. C… II. 张… 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(2008)第124729号

书 名: C语言程序设计

作 者: 张书云 主编

策划编辑: 严晓舟 秦绪好

责任编辑: 王占清

编辑部电话: (010) 63583215

特邀编辑: 薛秋沛

封面制作: 白雪

封面设计: 付巍

责任印制: 李佳

责任校对: 李 旻 陈 文

出版发行: 中国铁道出版社(北京市宣武区右安门西街8号 邮政编码: 100054)

印 刷: 化学工业出版社印刷厂

版 次: 2008年9月第1版

2008年9月第1次印刷

开 本: 787mm×1092mm 1/16 印张: 16 字数: 371千

印 数: 5 000册

书 号: ISBN 978-7-113-08770-8/TP·2801

定 价: 25.00元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签, 无标签者不得销售

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

全国高等院校信息管理与信息系统专业规划教材

编
审
委
员
会

主 任：杨小平

副主任：周山芙 严晓舟

委 员：（按姓氏音序排列）

曹淑艳 陈 光 计小宇 贾 伟 姜继忱

蒋砚章 匡 松 李桂根 李雁翎 刘瑞林

卢秉亮 毛汉书 潘 伟 孙连英 唐小毅

席宁华 谢怀军 杨 军 袁 枚 张 莉

张书云 赵 枫 赵 苹 赵宇斌 朱 雷

教育部高等学校文科计算机基础教学指导委员会在最新编写的《大学计算机教学基本要求(2006年版)》中指出：“21世纪是以工业文明为基础、信息文明为手段、生态文明为目标的高速发展的世纪，这也是人类进入了以知识经济为主导的信息时代的世纪。”作为高校基础教育，“培养德、智、体、美全面发展，具有创新精神和实践能力的专门人才，在包括文科专业在内的大学教育中继续加强计算机基础教育是十分必要的”。

全国高等院校计算机基础教育研究会编写的《中国高等院校计算机基础教育课程体系(2006)》中也提到：“高校计算机基础教育是高等教育中的重要组成部分，它面对的是占全体大学生95%以上的非计算机专业学生，它的目标是在各个专业领域中普及计算机知识和计算机应用，是使大学生成为既掌握本专业知识，又能熟练使用计算机技术的复合型人才。”同时还指出：“财经类专业的计算机基础教育与计算机专业相比，在培养目标、学生基础、专业性质和学时数量等方面都有很大差别，因此教学要求、教学内容、教学方法以及所用教材都应当有其自身的特点，应当针对各专业的实际需要来构建知识体系和课程体系。”

正是在这样的背景下，全国高等院校计算机基础教育研究会财经管理信息专业委员会和中国铁道出版社，共同组织各类高校的众多有多年教学实践的专家教授共同研究，并根据专业建设和课程设置精心策划了“全国高等院校财经管理类专业计算机规划教材”和“全国高等院校信息管理与信息系统专业规划教材”两套教材，并且成立了两套丛书的编审委员会，聘请相关学校有多年教学经验的老师根据编审委员会制定的统一方案和要求编写教材。

两套丛书都由十几本教材组成，将会陆续出版。

丛书强调学生在一定的理论基础上努力提高计算机技术的实际应用能力，希望学生能够了解基本的现代信息技术，熟练地使用各种与专业相关的计算机软件，并能结合专业培养和提高实践与创新能力。为了适应社会各界对毕业生的要求，本丛书也注意了行业、企业、专业职业技能资格认证的需要，紧密把握技能培养和实际操作能力的培养问题。丛书特点可以用面向专业、突出应用、培养能力、服务社会来概括。

当然，本丛书难免有许多不足之处，为了出好书、出精品书，也是为了向广大读者负责，我们真诚地希望广大师生和专家及时提出批评、给予指正，以便不断地随时改进我们的工作。

丛书编审委员会
2008年7月

前言

FOREWORD

程序设计是高等院校重要的计算机基础课程，它以编程语言为平台，介绍程序设计的思想和方法。通过该课程的学习，学生不仅要掌握高级程序设计语言的知识，更重要的是在实践中逐步掌握程序设计的思想和方法，培养问题求解和程序设计语言的应用能力，因此，这是一门以培养学生程序设计基本方法和技能为目标、以提高实践能力为重点的特色鲜明的课程。

C 语言是国内外广泛使用的程序设计语言之一，是国内外大学普遍开设的程序设计课程。

C 语言功能丰富，表达能力强，使用灵活方便，程序执行效率高，不但具有高级语言的功能，而且还可以实现汇编语言的许多功能。C 语言程序具有完善的模块程序结构，可移植性好，而且可以直接实现对系统硬件及外部设备接口的控制，具有较强的系统处理能力。

作为程序设计课程，其根本目的是让读者掌握基本的程序设计的概念、方法和技巧，而程序设计语言本身只是一个编程工具。本书在详细介绍、解析 C 语言语法知识的同时，通过大量的例题，由浅入深、循序渐进地充分展示了计算机解决问题的思想和方法，突出了程序设计的基本方法的阐述。

本书是作者根据多年从事程序设计语言及计算机相关课程的教学实践，在多次编写教案、教材的基础上经过精心整理、组织而成。全书讲述力求概念准确、严谨，语言简练，条理清晰，注重培养读者分析问题和程序设计的能力，注重培养良好的程序设计风格和习惯。

对于本书的学时安排，建议课堂教学 48 学时，上机实践 24 学时。

为了配合程序设计课程的教学和读者的学习，还编写了《C 语言程序设计习题解答与实验指导》，作为本书的配套参考书。

本书由张书云主编，姜淑菊、朱雷任副主编。本书第 1~4、6~9 章和附录由张书云、朱雷编写；第 5 章由姜淑菊编写，并参加了全书的修改工作；张书云负责全书书稿的修改、补充与总撰。

在本书的编写过程中，对外经济贸易大学的曹淑艳教授在百忙之中抽出时间对本书进行了审阅，王台和韩卫也对本书的编写提出了宝贵的建议，毕常振、卞云飞、乔俊丽、王建立和陈美玲参加了书稿的录入以及程序的调试工作，在此一并感谢。

由于时间仓促，内容涉及面较广，疏漏之处在所难免，恳请广大专家和读者批评指正。

编者

2008 年 7 月

第 1 章 C 语言概述.....	1
1.1 程序设计语言.....	1
1.2 C 语言的发展历史.....	2
1.3 为什么学习 C 语言.....	2
1.4 使用 C 语言进行程序开发.....	3
1.4.1 程序开发过程.....	4
1.4.2 源程序的编辑、编译、连接与运行.....	4
1.4.3 一个简单的 C 语言程序.....	6
1.5 C 语言程序的组成.....	10
1.6 C 程序编程风格.....	12
1.7 综合应用例题.....	12
习题.....	13
第 2 章 数据类型、常量与变量.....	15
2.1 数据类型.....	15
2.2 标识符.....	16
2.3 常量.....	17
2.3.1 整型常量.....	17
2.3.2 实型常量.....	17
2.3.3 字符型常量.....	17
2.3.4 字符串常量.....	18
2.3.5 常量的使用与符号常量.....	18
2.4 变量.....	19
2.4.1 变量定义.....	19
2.4.2 整型变量.....	20
2.4.3 实型变量.....	21
2.4.4 字符型变量.....	21
2.4.5 变量的初始化与赋值.....	22
2.5 综合应用例题.....	23
习题.....	25
第 3 章 数据的运算与输入/输出.....	27
3.1 运算符和表达式概述.....	27
3.2 算术运算符和算术表达式.....	27
3.2.1 算术运算符.....	28
3.2.2 算术表达式.....	29
3.3 赋值运算符和赋值表达式.....	30

3.3.1	赋值运算符.....	30
3.3.2	赋值表达式.....	30
3.3.3	复合的赋值运算符.....	31
3.4	关系运算符和关系表达式.....	32
3.4.1	关系运算符.....	32
3.4.2	关系表达式.....	32
3.5	逻辑运算符和逻辑表达式.....	33
3.5.1	逻辑运算符.....	33
3.5.2	逻辑表达式.....	33
3.6	条件运算符和条件表达式.....	35
3.7	逗号运算符和逗号表达式.....	35
3.8	位运算符和位运算表达式.....	36
3.8.1	按位取反运算符.....	36
3.8.2	按位“与”、按位“或”和按位“异或”运算.....	36
3.8.3	移位运算.....	37
3.8.4	位运算符的优先级及结合方向.....	38
3.9	数据类型转换.....	38
3.9.1	自动转换规则.....	39
3.9.2	强制转换.....	39
3.10	数据的输入/输出.....	40
3.10.1	字符数据的输入/输出.....	40
3.10.2	格式输出函数.....	42
3.10.3	格式输入函数.....	44
3.11	综合应用例题.....	46
	习题.....	47
第 4 章	C 语言 3 种基本结构程序设计.....	49
4.1	C 语句.....	49
4.2	顺序结构程序设计.....	50
4.3	选择结构程序设计.....	51
4.3.1	if 语句.....	51
4.3.2	switch 语句.....	55
4.4	循环结构程序设计.....	57
4.4.1	goto 语句.....	57
4.4.2	while 循环语句.....	58
4.4.3	do...while 循环语句.....	59
4.4.4	for 循环语句.....	60
4.4.5	循环语句的比较.....	62
4.4.6	循环的嵌套.....	62
4.4.7	循环控制语句 break 和 continue.....	63

4.5 综合应用例题.....	66
习题	69
第 5 章 函数	75
5.1 概述	75
5.1.1 模块与函数.....	75
5.1.2 函数的基本概念.....	76
5.1.3 函数定义的一般形式	77
5.2 函数的调用	79
5.2.1 函数调用的一般形式	79
5.2.2 函数调用的方式.....	79
5.2.3 函数的参数.....	80
5.2.4 函数的返回值.....	81
5.2.5 对被调函数的声明.....	82
5.3 函数的嵌套调用和递归调用	84
5.3.1 嵌套调用	84
5.3.2 函数的递归调用.....	86
5.4 变量的存储属性.....	89
5.4.1 变量的作用域.....	89
5.4.2 变量的生存期.....	91
5.4.3 变量的存储类型.....	92
5.4.4 存储类型小结.....	97
5.5 内部函数和外部函数.....	98
5.5.1 内部函数	98
5.5.2 外部函数	98
5.6 编译预处理	99
5.6.1 宏定义	99
5.6.2 “文件包含”处理.....	104
习题	105
第 6 章 数组	110
6.1 一维数组	110
6.1.1 一维数组的定义.....	110
6.1.2 一维数组元素的存储形式	111
6.1.3 一维数组元素的引用	111
6.1.4 一维数组的初始化.....	112
6.1.5 一维数组应用举例.....	113
6.2 二维数组	115
6.2.1 二维数组的定义.....	115
6.2.2 二维数组元素的存储形式	116
6.2.3 二维数组元素的引用	116

6.2.4	二维数组的初始化	117
6.3	字符数组和字符串	119
6.3.1	字符数组的定义	119
6.3.2	字符串和字符串的存储方法	119
6.3.3	字符数组的初始化	121
6.3.4	字符数组的引用	121
6.3.5	字符串的输入/输出	122
6.3.6	字符串处理函数	123
6.4	函数与数组	126
6.4.1	数组元素作函数实参	126
6.4.2	一维数组名作函数参数	126
6.4.3	多维数组名作函数参数	129
6.5	综合应用例题	130
	习题	132
第 7 章	指针	135
7.1	指针的基本概念	135
7.1.1	地址与指针	135
7.1.2	指针变量的定义	136
7.1.3	指针变量的引用	137
7.1.4	指针变量作函数参数	138
7.2	指针的运算	142
7.2.1	指针的赋值运算	142
7.2.2	指针的算术运算	143
7.2.3	指针的关系运算	143
7.3	数组的指针和指向数组的指针变量	144
7.3.1	指向数组元素的指针变量的定义与赋值	144
7.3.2	通过指针引用数组元素	144
7.3.3	数组名作函数参数	146
7.3.4	多维数组的指针	150
7.4	指针与字符串	155
7.4.1	指向字符串的指针	156
7.4.2	字符串指针作函数参数	157
7.4.3	字符指针变量和字符数组的异同	160
7.5	函数的指针和指向函数的指针变量	161
7.5.1	函数指针变量的定义与使用	161
7.5.2	用函数指针变量作函数参数	162
7.6	指针型函数	164
7.7	指针数组和指向指针的指针	165
7.7.1	指针数组的概念	165

7.7.2	用指针数组处理多个字符串	166
7.7.3	命令行参数	168
7.7.4	指向指针的指针	170
	习题	171
第 8 章	结构体与共用体	176
8.1	结构体类型	176
8.1.1	结构体类型的定义	176
8.1.2	结构体类型变量的定义	177
8.1.3	结构体变量的初始化	179
8.1.4	结构体变量的引用	179
8.1.5	结构体数组	181
8.2	结构体与指针	185
8.2.1	指向结构体变量的指针	185
8.2.2	指向结构体数组的指针	186
8.2.3	结构体与函数	188
8.3	共用体	191
8.3.1	共用体的概念	191
8.3.2	共用体变量的引用	193
8.4	枚举类型	194
8.5	用 typedef 定义数据类型	196
8.6	链表	197
8.6.1	链表概述	197
8.6.2	动态链表的建立	199
8.6.3	链表的遍历	201
8.6.4	链表的插入和删除	202
8.6.5	链表综合应用例题	206
	习题	208
第 9 章	文件系统	211
9.1	文件概述	211
9.1.1	什么是文件	211
9.1.2	C 文件分类	211
9.2	文件指针	213
9.3	文件的打开与关闭	214
9.3.1	文件打开函数	214
9.3.2	文件关闭函数	216
9.4	文件的读写	216
9.4.1	字符文件的读写	217
9.4.2	字符串读写函数	220
9.4.3	数据块读写函数	221

9.4.4 格式化读写函数.....	223
9.5 文件的定位	224
9.5.1 rewind 函数.....	224
9.5.2 fseek 函数.....	225
9.6 检测文件状态函数.....	226
9.7 综合应用例题.....	227
习题	229
附录 A 常用字符与 ASCII 码对照表.....	232
附录 B C 语言的关键字	234
附录 C C 语言运算符的优先级与结合性.....	235
附录 D Turbo C 常用库函数.....	237
参考文献	244

第 1 章 C 语言概述

随着科学技术的飞速发展，计算机技术日新月异，计算机程序设计语言也不断更新换代。什么是程序设计语言？应该学习哪一种程序设计语言？如何进行程序设计？这些都是程序设计初学者首先遇到的问题。本章将介绍程序设计语言的发展，C 语言的发展历史，C 语言的特点，C 程序开发步骤及 C 程序的组成。

1.1 程序设计语言

什么是程序设计语言？为什么要使用程序设计语言？提到语言这个词，人们自然会想到英语、汉语等自然语言，因为语言是人与人之间交流信息的工具。当今计算机遍布生活的每一个角落，除了人与人之间的相互交流之外，人还必须和计算机交流。用什么样的方式和计算机直接交流呢？人们自然想到的是语言。人与人交流用的是双方都能听懂的自然语言，同样，人与计算机交流也要用人和计算机都能接受且理解的语言，这就是计算机语言，也被称为程序设计语言。

从计算机诞生到今天，程序设计语言也在伴随着计算机技术的进步不断升级换代，其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

一种 CPU 的指令系统，也称为 CPU 的机器语言，它是 CPU 可以识别的一组由 0 和 1 序列构成的指令码。例如，10000000 和 10010000 分别表示某 CPU 指令系统中的“加”和“减”指令。

用机器语言编写的程序能被计算机直接识别和执行，但机器语言程序不便于记忆、阅读和书写，也不容易查错。机器语言是第一代计算机程序设计语言。

2. 汇编语言

为了克服机器语言的缺点，20 世纪 50 年代中期，人们开始用一些“助记符号”来代替 0、1 码编程。如前面表示“加”和“减”的二进制指令可分别用 ADD 和 SUB 代替。这种用助记符号描述的指令系统，称为符号语言或汇编语言。

用汇编语言编写的程序较机器语言编写的程序可读性大大提高，程序维护变得更加方便。但是汇编语言指令是机器不能直接识别和执行的，必须由“汇编程序”将这些符号翻译成机器语言才能运行。这种“汇编程序”就是汇编语言的翻译程序。

汇编语言和机器语言都是依 CPU 的不同而异，因此它们都被称为面向机器的语言（一般被称为低级语言）。用面向机器的语言，可以编写出效率很高的程序。但是程序员用它们编程时，不仅要考虑解题思路，还要熟悉机器的内部结构。汇编语言被称为第二代程序设计语言。

3. 面向过程的语言

汇编语言和机器语言是面向机器的，因机器而异。1954 年出现的 FORTRAN 语言以及随后相继出现的其他语言（被称为高级语言），这些语言使人们开始摆脱进行程序设计时必须熟悉计算机硬件的桎梏，以使用户把精力集中于解题思路和方法上，使程序设计语言开始与解题方法相结合。其中一种方法是把解题过程看做数据被加工的过程，基于这种方法的程序设计语言被称为面向过程的程序设计语言。

面向过程的程序设计语言有 FORTRAN、BASIC、Pascal、C 等，这种语言被称为第三代程序设计语言。

4. 面向对象的程序设计语言

面向对象的程序设计把现实世界看成是由许多对象（object）所组成的，对象之间通过互相发送和接收消息进行联系；消息激发对象本身的运动，形成对象状态的变化。从程序结构的角度看，每个对象都是一个数据和方法的封装体——抽象数据类型。

从分类学的观点看，客观世界中的对象都是可以分类的。也就是说，所有的对象都属于特定的“类”（class），或者说每个对象都是类的一个实例。因而，面向对象的程序设计的一个关键是定义“类”，并由此“类”生成“对象”。

面向对象的程序设计语言有 Java、C++、Visual Basic、Visual FoxPro 等。面向对象的程序设计语言通常被认为是第四代程序设计语言。

面向对象的程序比面向过程的程序更清晰易懂，更适宜编写大规模的程序，正在成为当代程序设计的主流。但是，面向过程的程序设计语言更适合初学者理解与掌握。

1.2 C 语言的发展历史

C 语言诞生于 20 世纪 70 年代初期，它是在 B 语言的基础上发展起来的。而 B 语言是由美国贝尔实验室的 K.Thompson 在 1970 年对当时的 BCPL 语言进行了修改后命名的。1972 年，美国贝尔实验室的 D.M.Ritchie 和 B.W.Kerninghan 在设计 UNIX 操作系统时，发现需要一种比 B 语言更加灵活的语言；因此他们对 B 语言作了进一步的完善和发展，提出了一种新型的程序设计语言——C 语言，并使用该语言改写了 UNIX 操作系统。

由于 C 语言功能强大而灵活，因此很快在世界范围内被广泛采用。然而，随着 C 语言在不同组织之间的使用，出现了一些不一致的地方。为了改变这种情况，1983 年，美国国家标准化协会（ANSI）根据 C 语言问世以来各种版本对 C 语言的发展和扩充，制定了新的标准，称为 ANSI C。最近一次修改是在 1999 年通过的标准。该标准增加了一些新的特性，简称 ANSIC-99。

目前被广泛采用的各种版本 C 语言编译系统，在微型计算机上的一些编译软件，如 C++ Builder X、DEV-C++、Intel C++ Compile、Turbo C++ 3.0 等，基本上都以 ANSIC-99 为基础，并在此基础上各自作了一些扩充，使之更加方便、完美。

1.3 为什么学习 C 语言

自从 C 语言问世以来，至今已经有 30 多年的历史，从最初为设计 UNIX 操作系统而开发的语言，到现在已经发展成为被广泛采用的系统描述语言和通用的程序设计语言。虽然在当前的计算

机编程领域中，有大量的高级语言可供选择，但很多计算机专业人员认为 C 语言是很好的程序设计语言，这主要是因为 C 语言具有以下特点：

1. 功能强大且灵活

C 语言把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以同汇编语言一样对位、字节和地址进行操作。因此既具有高级语言的功能，又具有低级语言的功能，可以用来编写各种类型的软件。

2. 比较简洁

C 语言比较简单，是一种比较精练的语言，它共有 37 个关键字，9 种控制语句。语言中的概念同面向对象的语言相比更容易理解，程序书写自由，往往可以使用比其他语言更少的代码来实现相同的功能。

3. 程序执行效率较高

使用 C 语言开发的程序在生成目标代码后，一般只比汇编程序生成的目标代码效率低 10%~20%，也就是说，使用 C 语言书写的程序运行起来非常快。

4. 可以移植

C 语言的另外一个突出优点是在各种操作系统上都可以使用 C 语言来书写程序，同时在一种操作系统上书写的程序，也可以很容易地在其他操作系统上运行。例如：在 Windows 平台上书写的 C 语言程序可以在几乎不作修改的情况下，在 Linux、UNIX 等其他操作系统下编译、连接、运行。

5. 结构式语言

C 语言是结构式语言，因此使用 C 语言可以在一个结构内部完成许多特定的功能，这些特定的功能组合成为函数，可以提供给其他程序使用。因此可以创建很多简短的、可重用的代码。

近年来，面向对象的语言逐渐成为编写各种应用程序的主流。在这些语言中，C++、Java 语言占较大的比例。那么是否应当直接学习 C++ 而不是 C 语言呢？实际上 C++ 语言是由贝尔实验室的 Bjarne Stroustrup 于 1983 年设计出来的，该语言在 C 语言的基础上进行了扩充和完善，同时增加了面向对象编程方面的内容，因而也增加了整个语言的复杂性，对于程序设计初学者来说掌握起来有一定难度。当学习完 C 语言后再学习 C++ 语言时，会发现几乎有关 C 语言的所有知识都适用。

另外两种流行的面向对象的语言 Java 和 C#，同 C 语言也有着非常相似的部分。如果以后决定学习 Java 语言或者 C# 语言，同样会发现，很多有关 C 语言的知识也适用于 Java 或者 C# 语言编程。

1.4 使用 C 语言进行程序开发

在开始学习使用 C 语言进行程序设计开发时，初学者可能最先遇到的问题是：“什么是程序？什么是程序设计？”。程序可以看做对一系列操作过程的描述。日常生活中也可以找到许多“程序”实例。例如，打开电视机的操作过程可以描述如下：

- (1) 插上电源。
- (2) 打开电视机开关。
- (3) 使用遥控器打开电视。

以上就是完成打开电视机任务的“程序”，如果按照这个“程序”实施这些步骤，就可以完成打开电视机这项任务。

计算机中的程序执行与日常生活中程序性活动的情况很相似。计算机的基本功能是可以执行一组基本操作，每个操作完成一件很简单的计算工作，例如整数的加减乘除运算等。为使计算机能按照人的指挥工作，每种计算机都提供了一套指令，其中的每一种指令对应着计算机能执行的一个基本动作。人们编制这些指令的工作被称为程序设计或者编程，这种工作的产品就是程序。

一连串的指令集合可以指挥计算机工作，但是人们不容易理解。为了更好地与计算机交流，需要一种意义清晰、用起来比较方便、计算机也能处理的描述方式。这一套描述方式逐渐形成了程序设计语言。

1.4.1 程序开发过程

在整个程序开发的过程中，通常分为 3 个阶段：首先需要确定程序设计所要达到的预期目标；其次确定要使用什么样的方法来编写程序；最后创建程序，运行程序来获取结果。

第一阶段是确定程序的目标，也就是程序设计的目的。例如，编写一个网络游戏或者仅仅是将程序设计者的姓名显示到屏幕上等。如果没有目标，将无从编写程序。

第二阶段是确定要使用什么样的方法来编写程序。需要使用哪些信息，通过什么样的步骤和次序，使用什么样的公式来达到程序目标的要求。例如，编写一个计算圆面积的程序。第一阶段已经完成，因为目标已经明确，即计算圆的面积。第二阶段是确定要计算圆的面积，需要知道哪些信息。这里假设用户提供圆的半径 r ，知道这些信息后，就可以使用公式 $\pi \times r^2$ 获得答案。

第三阶段是创建程序，运行程序来获取结果。前两个阶段不需要计算机设备的参与，第三阶段则需要在计算机上书写程序。

第三阶段的具体工作包含如下 4 个步骤：

- (1) 使用编辑器书写设计好的源程序，并将其保存为磁盘文件。
- (2) 编译源程序以创建目标文件。
- (3) 连接编译后的程序，创建一个可执行文件。
- (4) 运行程序，看其运行方式是否与需要达到的目标相符。

1.4.2 源程序的编辑、编译、连接与运行

1. 使用编辑器书写源程序

源程序是一系列的语句或命令，用于指示计算机执行编程者期望的任务。例如，下面是一行 C 语言源代码：

```
printf("Hello,World!\n");
```

上述语句命令计算机将文字“Hello,World!”显示到计算机屏幕上。为了将上述代码输入计算机，可以使用各种各样的文本编辑器。只不过现有的很多程序语言编译器都自带了文本编辑器，这些编辑器可以很方便地用来书写程序代码，可以直接在其中书写代码。写好源程序以后，可以将源程序进行保存，文件的扩展名为.c。

2. 编译源程序

为了更好地在计算机同人之间进行交互，人们使用了程序语言来代替计算机的指令。程序设计人员可以很容易地理解上述 C 语言源代码，但是计算机却不能。计算机只能识别或被称为机器

语言的二进制指令。为了使计算机进行工作，需要将设计好的程序转换成为计算机可以识别的机器语言，计算机才可以按照设计人员的指令来工作。这种转换工作并不需要手动来完成，它是由一个被称做编译器的程序完成的。编译器将源程序文件作为输入，经过编译以后生成一个磁盘文件，该文件中就包含了与源程序语句对应的机器语言的二进制指令。编译器创建的机器语言指令被称为目标代码，而包含它们的磁盘文件被称为目标文件。通常使用.obj作为文件的扩展名。

3. 程序连接

目标文件包含了程序设计人员设计的程序指令，但是仍然不能够直接运行，还需要完成程序连接部分的工作。进行程序连接的主要原因是：为了简化程序设计人员在程序设计时候的工作，编译器中提供了很多使计算机能够正常工作所必须使用的一些通用代码，这样可以使程序员在设计每一个程序的时候，可以把更多的精力放在所需要解决的问题上。连接部分的主要工作是把这些编译器中提供的程序（通常以库文件的方式）同设计好的程序的目标文件连接起来，最终生成一个可以被计算机执行（程序可以在计算机上运行）的完整的二进制文件。这个文件也称做可执行程序，文件的扩展名为.exe。

从上面的步骤可以看出：编译、连接步骤都是通过编译器来完成的。设计人员仅需按照C语言的语法规则书写源代码，其他工作由编译器来完成。大多数开发环境提供了一个选项，可以设置是分为两个步骤完成编译和连接任务还是一步完成。不管编译和连接工作是如何完成的，这两个过程都是独立的操作。图1-1说明了从源代码到目标代码，再到可执行程序的过程。

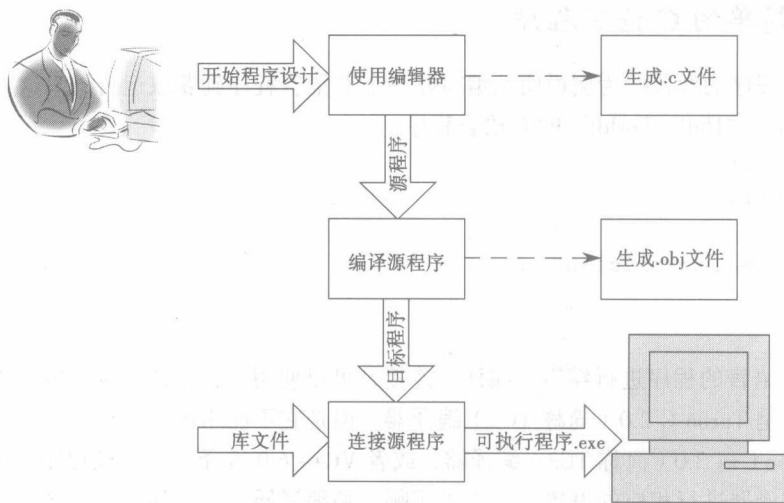


图 1-1 程序的执行过程

4. 运行程序

程序经过编译和连接并创建出可执行文件后，便可以在系统提示符下输入其名称（或像运行Windows操作系统下的其他程序那样，使用鼠标双击文件图标）运行它，并观察其运行方式和运行结果是否与设计目标相符合。如果运行程序时得到的结果与期望的结果不一致，则需要回到程序开发步骤的第2步，即确定要使用什么样的方法来编写程序，来找出导致出现不一致的原因，如果没有发现第2步中的错误，则需要在源程序中查找错误，并最终在源程序中进行更正。修改源程序后，需要重新编译、连接程序，并得到新的可执行文件，然后再次运行这个程序。这样的