普通高等教育“十一五”国家级规划教材
普通高等学校计算机教育“十三五”规划教材

C语言程序设计教程

（第5版）

THE C PROGRAMMING LANGUAGE
(5th edition)

李丽娟 ◆ 主编

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

普通高等教育“十一五”国家级规划教材

普通高等学校计算机教育“十三五”规划教材

C语言程序设计教程

（第5版）

THE C PROGRAMMING LANGUAGE
(5th edition)

李丽娟 ◆ 主编

人民邮电出版社
北京

图书在版编目（CIP）数据

C语言程序设计教程 / 李丽娟主编. -- 5版. -- 北京 : 人民邮电出版社, 2019.7
普通高等学校计算机教育“十三五”规划教材
ISBN 978-7-115-49568-6

Ⅰ. ①C… Ⅱ. ①李… Ⅲ. ①C语言－程序设计－高等学校－教材 Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字(2018)第227931号

内 容 提 要

本书以C语言的基本语法、语句为基础，深入浅出地讲述了C语言程序设计的基本概念、思想与方法。全书以程序案例为导向，采用计算思维的方法设计程序，通过程序案例，拓宽学生的思维，引导学生自主思考，逐步掌握程序设计的一般规律和方法。全书理论联系实际，突出模块化程序设计方法。

全书内容可分为3个部分，共11章。第1部分为第1~2章，介绍初学者的入门知识，主要内容有C语言程序的结构、基本数据类型及其取值范围、基本运算符、表达式及运算的优先级。第2部分为第3～5章，介绍程序设计的基本结构，主要内容有程序的简单算法设计、程序语句的基本控制结构。只要掌握了第1、第2部分的内容，学生就可以完成简单的程序设计。第3部分为第6～11章，介绍模块化程序设计的概念和实现方法，主要内容有函数、数组、指针、结构体、文件、位运算等。通过对这3个部分内容的学习，学生能够逐步认识模块化程序设计的思想，掌握模块化程序设计的方法。

全书语言简洁，通俗易懂，程序案例丰富，内容叙述由浅入深，适合作为大学本科和专科院校的教材，也可供一般工程技术人员参考。

◆ 主　　编　李丽娟
责任编辑　邹文波
责任印制　陈　犇

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路11号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市中晟雅豪印务有限公司印刷

◆ 开本：787×1092　1/16
印张：19.75　　2019年7月第5版
字数：545千字　　2019年7月河北第1次印刷

定价：52.00元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字20170147号

第 5 版前言

“C 语言程序设计”是计算机专业及理工类各专业重要的基础课程之一。理论联系实际是该课程的一个特点，怎样将理论知识应用于解决实际问题是学好这门课程的重点和难点。为适应我国计算机技术的应用和发展，培养学生解决实际问题的能力，编者根据多年的教学和实践经验，结合当前高等教育大众化的趋势，在分析国内外多种同类教材的基础上，编写了本书。

本书第 1 版于 2006 年出版，2009 年进行改版。2011 年，我们对全书内容进行分离和重组，直接从函数部分开始，加入 C++语言的入门基础，出版了《C/C++语言程序设计教程——从模块化到面向对象》。2013 年，我们根据教学的特点和要求，对第 3 版的内容进行了优化，出版了《C 语言程序设计（第 4 版）》。

本次改版时，编者在保留前几版特色的基础上，结合多年的教学经验，并特别根据近几年教学改革的实践及社会对人才培养的高标准要求，对内容做了进一步的优化、补充和完善。本书通过理论联系实际，采用计算思维的方法，引导和启发学生的思维。全书将程序案例分为两种类型：一种为了解基本知识型，主要通过程序加深学生对基础知识的理解和掌握；另一种为应用型，通过对实际案例问题的分析，逐步引导学生掌握思考问题和解决问题的方法。全书大部分案例后面都留有思考问题，鼓励学生对解决问题的方法举一反三，激发学生的创新思维。近几年的教学实践表明，在程序设计课程教学中采用计算思维的方法来解决问题，有利于培养学生的综合应用能力，对培养工程应用型人才是有益的。实践还表明，用流程图来表达算法，能使学生更好地理解结构化程序设计的思想，掌握 C 语言程序设计的核心方法。这些内容对各类普通高校本科、专科学生都是适用的。

本书将 C 语言程序设计分成以下 3 个循序渐进的部分。

第 1 部分是入门基础，由第 1～2 章组成，这部分程序的结构主要是顺序结构。这部分主要介绍 C 语言程序的结构、数据的表达方式、基本表达式语句、C 语言程序的运行方式等。这部分内容奠定了 C 语言程序设计的基础，其中 2.1 小节为预备知识，对这部分内容，读者可以根据自身情况自学或者跳过。通过对这一部分内容的学习，学生应达到的目标是“可以设计由简单表达式语句组成的按顺序执行的程序”。

第 2 部分是程序设计的基本结构，由第 3～5 章组成，这部分程序的语句结构为分支和循环结构。这部分主要介绍程序设计的简单算法表示方法，以及两种重要的程序语句结构——分支结构和循环结构。这两种结构是程序设计的应用基础。通过对这一部分内容的学习，学生应充分了解分支结构和循环结构的基本规则。结合第 1 部分的顺序结构，学生应达到的目标是“能采用计算思维的方法，设计简单的算法，并依据算法编写程序，掌握思考问题和解决问题的方法”。

第 3 部分是程序设计方法和手段的提高，由第 6～11 章组成，这部分程序的结构为模块化的形式。这部分主要介绍程序的模块化实现方法和更多的程序设计手段。这部分内容为程序设计应用核心。通过对这一部分内容的学习，学生应充分了解模块化程序设计的思想。结合前两部分的内容，学生应达到的目标是“掌握程序模块的设计方法，采用计算思维的方法对问题进行分解，灵活地使用指针、结构、文件、位运算等手段和方法编写程序，培养创新思维的能力和解决问题的能力”。

本书具有以下特色。

（1）引导学生从感性认识上升到理性认识

本书的开始部分介绍 C 语言程序的基本结构和开发环境，使学生可以从感性上认识 C 语言程序的基本组成，了解 C 语言从程序编写到程序调试、运行的基本过程。第 2 章的开始部分描述数据存储的预备知识，供有兴趣的学生自行阅读，加深对计算机数据的了解。第 3 章介绍程序的简单算法表示方法，为程序设计提供有效的依据。

（2）案例丰富，层次感强，具有较好的可扩展性

本书共精选了 100 多个程序，大部分程序都在 Visual Studio 2010 环境和 Dev C++ 5.11 环境下通过验证（个别不能在 Visual Studio 2010 环境和 Dev C++ 5.11 环境下通过的程序有特别说明），并且对程序的结构、函数的设计、变量的设置进行了恰当的注释和说明。其中大部分程序案例采用计算思维的方法给出了分析，并留有可进一步探讨的余地，给学习留下广阔的空间，可以启发学生思考，从中发现问题，寻找解决问题的方法，从而不断激发学生的学习兴趣，激发其想象力和创新思维能力。

（3）问题分析引导，算法流程图规范

本书通过对问题的分析引导，找出解决问题的关键，并给出规范的流程图，强化解决问题的科学过程和手段，培养学生严谨的思考问题和解决问题的能力。

本书每章都附有习题，以帮助学生理解基本概念，巩固所学的知识。学生通过理论联系实际，进行书面练习和上机实践，进一步熟练掌握 C 语言的基本思想和基本语句，提高程序设计能力。

与本书配套的《C 语言程序设计教程实验指导与习题解答（第 5 版）》给出了学生上机实验的内容和本书中习题的参考答案。在实验中，学生可以先编写程序，然后编译、运行，查看程序的运行结果，根据程序的运行结果验证程序的正确与否，从而逐步掌握 C 语言程序设计的基本方法和基本技能。

“C 语言程序设计”课程的建议学时数为 88，其中，课堂教学学时数为 40，上机实验学时数为 48，书中有*号标注的内容可根据教学安排不讲或少讲。各章的教学学时数安排可大致如下表所示。实际教学中可以根据具体情况予以调整，适当减少或增加学时数。

章	内　　容	课堂教学学时数	上机实验学时数
1	引言	2	2
2	基本的程序语句	4	6
3	程序的简单算法设计	2	2
4	分支结构	2	4
5	循环结构	2	4
6	函数与宏定义	8	8
7	数组	4	4
8	指针	8	8
9	构造数据类型	4	4
10	文件操作	2	4
11	位运算	2	2
合计		40	48

本书可以作为普通高等院校计算机专业及理工类各专业本科、专科学生的教材，也可作为研究生入学考试和各类计算机认证考试的参考书，还可作为计算机应用工作者和工程技术人员的参考书。

本书由李丽娟担任主编，吴蓉晖、谷长龙、官骏鸣、张宇波、郝耀军、张志宏、洪跃山、李根强、杜四春担任副主编。

由于编者水平有限，书中难免存在不足与疏漏之处，敬请广大读者批评指正。

如果需要书中的源程序代码和教学用的 PPT 文件，请登录人邮教育社区（http://www.ryjiaoyu.com）下载，或者与编者联系（邮箱：jt_lljh@hnu.cn）。

李丽娟

2019 年 5 月于岳麓山

目 录

第1章 引言

1.1 C语言的发展过程

C语言的前身是ALGOL 60。1960年5月，Peter Naur的ALGOL 60报告发表。之后英国剑桥大学将ALGOL 60发展成为CPL（Combined Programming Language），1967年，英国剑桥大学的Matin Richards对CPL进行了简化，产生了BCPL。1970年，美国贝尔实验室的Ken Thompson对BCPL进行了修改，将其命名为“B语言”，并用B语言写了第一个UNIX操作系统。

1973年，美国贝尔实验室的D.M.Ritchie在B语言的基础上设计出了一种新的语言——C语言。1977年，Dennis M.Ritchie发表了不依赖于具体机器系统的C语言编译文本《可移植的C语言编译程序》。1978年，Brian W.Kernighian和Dennis M.Ritchie合作出版了著名的*The C Programming Language*一书。1983年，美国国家标准协会（American National Standards Institute，ANSI）在此基础上制定了一个C语言标准，我们通常称之为ANSI C，从而使C语言成为世界上应用最为广泛的高级程序设计语言。

1.2 C语言的特点

C语言是一种结构化的程序设计语言，它简明易懂，功能强大，适用于各种硬件平台。与常见的高级语言不一样的是，C语言兼有高级语言和低级语言的功能，既适用于系统软件的开发，也适用于应用软件的开发。

C语言的特点还表现在以下几个方面。

1. 程序设计结构化

结构化就是将程序的功能进行模块化，每个模块都具有不同的功能，程序将一些不同功能的模块有机组合在一起，通过模块之间的相互协同工作，共同完成程序所要完成的任务。这种模块化的程序设计方式使C语言程序易于调试和维护。

了解和掌握结构化程序设计的思想，有助于提高我们分析问题和解决问题的能力。

2. 运算符丰富

C语言共有34种运算符。它把括号、赋值、逗号等都作为运算符处理，因而C语言的运算

类型极为丰富，可以实现其他高级语言难以实现的一些运算。

3. 数据类型丰富

C 语言除了具有系统本身规定的一些数据类型外，还允许用户定义自己的数据类型，以满足程序设计的需要。

4. 书写灵活

只要符合 C 语言的语法规则，程序的书写格式并不会受到严格的限制。

实际编写程序时并不提倡这样做，要求根据语法规则按缩进格式书写程序。

5. 适应性广

C 语言程序生成的目标代码质量高，程序执行效率高，与汇编语言相比，用 C 语言写的程序可移植性好。

6. 关键字简洁

在 C 语言中，关键字有其特殊的意义和作用，不允许用户将其用于其他用途，所有关键字都必须是小写英文字母。ANSI C 规定，C 语言共有 32 个关键字，这 32 个关键字可以分为以下 4 类。

（1）数据类型关键字 12 个。

（2）控制类型关键字 12 个。

（3）存储类型关键字 4 个。

（4）其他关键字 4 个。

C 语言的关键字及其作用如表 1-1 所示。

表 1-1　　C 语言的关键字及作用

序号	类　型	关键字					作　用
1	数据类型（9 个）	char	int	float	double	void	用于数据类型的声明
		short		long			用于声明整型数据的大小
		signed		unsigned			用于声明整型数据在正负坐标上的区间
	自定义的数据类型（3 个）	struct					用于声明结构体数据类型
		union					用于声明共用体数据类型
		enum					用于声明枚举数据类型
2	控制类型（12 个）	if	else	switch	case	default	用于分支结构
		for	while	do...while			用于循环结构
		continue					结束本次循环，进入下一轮循环
		break					直接跳出循环结构或分支结构
		goto					直接转移到指定的语句处
		return					返回到函数调用处
3	存储类型（4 个）	auto					用于声明自动变量（可缺省）
		extern					用于声明外部变量
		register					用于声明寄存器变量
		static					用于声明静态变量

续表

序号	类 型	关键字	作 用
4	其他类型（4个）	const	用于声明只读变量
		sizeof	计算数据类型长度
		typedef	给自定义数据类型取别名等
		volatile	变量在程序执行中可被隐含地改变

需要说明的是，除上述关键字外，不同的实现环境对C语言的关键字有所扩充，并且扩充的关键字会因实现环境的不同而不同，读者只需要从使用的实现环境中去了解即可，在此不多加赘述，扩充的关键字只适用于特定的实现环境。

7. 控制结构灵活

C语言的程序结构简洁高效，使用方便、灵活，程序书写自由。C语言一共有9种控制结构，可以完成复杂的计算，9种控制结构及作用如表1-2所示。

表1-2 C语言的9种控制结构及作用

关键字	作 用	关键字	作 用	关键字	作 用
goto	直接转移	for	循环语句	break	直接跳出循环结构或分支结构
if	条件分支	do…while	循环语句	continue	结束本次循环，开始下一轮循环
switch	多路分支	while	循环语句	return	返回到函数调用处

表1-2所示的关键字与表1-1所示的关键字的意义和作用是相同的，表1-2中的continue只能用于循环，而break可以用于循环或switch多路分支。

了解C语言的上述特点，对我们学习和掌握好C语言程序设计很有帮助。虽然C语言程序对书写的要求没有太多限制，只要符合语法规则就行，但在这里我们强调程序书写必须规范，特别是对初学者，这一点很重要。一个书写规范、整齐的C语言程序能够帮助程序员快速读懂程序所表达的思想，同时也能更清晰地将程序设计的意图正确地表达出来。

1.3 简单的C语言程序

为了说明C语言源程序结构的特点，先看以下几个程序。这几个程序由易到难，表现了C语言源程序在组成结构上的特点。虽然有关内容还未介绍，但从这些例子可以了解一个C语言源程序的基本组成部分和程序的书写格式。

【例1-1】编写程序，在屏幕上输出“Hello,World!”的字符串。

程序如下：

```
/* example1_1.c  在屏幕上输出字符串*/
#include <stdio.h>
void main()
{
    printf("Hello,World!\n");
}
```

程序运行结果：

```
Hello,World!
```

程序说明：

（1）#include 称为文件包含命令。其作用是把系统目录下的头文件 stdio.h 包含到本程序中，成为本程序的一部分。这里被包含的文件是由系统提供的，所以用尖括号< >来标定，其扩展名为.h，也称为头文件或首部文件。

C 语言系统提供的头文件包括各种标准库函数的函数原型，因此，凡是在程序中调用某个库函数，都必须将该函数原型所在的头文件包含进来。在这里，包含的文件是 stdio.h。该文件里的函数主要是处理数据流的标准输入/输出，此时表示在程序中要用到这个文件中的函数。“#”只是一个标志。

（2）main 是主函数的函数名，表示这是一个主函数。1 个可执行的 C 语言源程序只允许有 1 个 main 函数。

（3）printf()函数是一个库函数，其函数原型在头文件 stdio.h 中。该函数的功能是将圆括号内的内容输出到显示器。

（4）main()函数中的内容必须放在一对花括号“{}”中。

【例 1-2】请从键盘输入一个角度的弧度值 x，计算该角度的余弦值，将计算结果输出到屏幕。

程序如下：

```
/* example1_2.c  计算角度的余弦*/
#include <stdio.h>
#include <math.h>
void main()
{
     double x,s;
     printf("Please input value of x: ");
     scanf("%lf",&x);
     s=cos(x);
     printf("cos(%lf)=%lf\n",x,s);
}
```

程序运行结果：

```
Please input value of x: 0↵
cos(0.000000)=1.000000
```

在这里要求输入的角度为弧度值，如果要计算 cos 180°，则要输入 π 的值，再次运行上面这个程序。

程序运行结果：

```
Please input value of x: 3.1415926↵
cos(3.141593)=-1.000000
```

程序说明：

（1）程序包含两个头文件：stdio.h、math.h。

（2）main()函数定义了两个双精度浮点型变量 x、s。

（3）printf("Please input value of x:"); 用于显示提示信息。

（4）scanf(" %lf ",&x); 用于从键盘获得一个实数 x。

（5）s=cos(x); 用于计算 x 的余弦，并把计算结果赋给变量 s。

（6）printf ("cos(%lf)=%lf\n",x,s); 将计算结果输出到屏幕。双引号中有两个格式字符“%lf”，分别对应着 x 和 s 两个输出变量。

（7）程序运行后会在屏幕的左上方显示提示信息，要求用户从键盘输入一个用弧度值表示的角度值 x，接下来程序会计算该角度的余弦，并将计算结果输出到屏幕。

本例使用了 3 个库函数：输入函数 scanf()、余弦函数 cos()和输出函数 printf()。余弦函数 cos()是数学函数，其函数原型在头文件 math.h 中，scanf()和 printf()是标准输入/输出函数，其函数原型在头文件 stdio.h 中。

需要说明的是，在有些编译环境中，可以省去 scanf()和 printf()这两个函数的包含命令。所以在【例 1-1】和【例 1-2】中可以省略文件包含命令#include <stdio.h>。但我们不建议这么做，很显然，任何时候，明确的表达对程序的理解是有益的，一般情况下，只要程序中用到了头文件中的函数原型，都应该将相应的头文件包含进来。这是一个良好的习惯，否则，有可能由于编译系统的不同而发生语法错误。希望读者在开始学习的时候就养成良好的书写习惯。

【例 1-3】设计一个加法器，能实现两数的相加。通过调用该加法器，计算两数的和。

```
/* example1_3 两数相加的加法器*/
#include <stdio.h>
int add(int x, int y); /*加法器的函数原型声明*/
void main()
{
     int a, b, c;
     printf ("please input value of a and b:\n");
     scanf("%d %d", &a, &b);
     c=add(a,b);
     printf ("sum=%d\n",c);
}
int add(int x, int y) /*加法器的函数定义*/
{
     return(x+y);
}
```

程序运行结果：

```
please input value of a and b:
5 9↵
sum=14
```

程序说明如下。

（1）【例 1-3】中的主函数体分为两部分，一部分为说明部分，另一部分为执行部分。说明是指变量的类型说明（【例 1-1】中未使用任何变量，因此无说明部分）。C 语言规定，源程序中所有用到的变量都必须先说明，后使用；否则会出现语法错误。这是编译型高级程序语言的一个特点，说明部分是 C 语言源程序结构中很重要的组成部分。

（2）程序语句 c=add(a,b);通过调用加法器 add()来完成（a+b）的计算，并将计算结果赋给变量 c。

（3）运行程序时，首先在显示屏幕上显示字符串 please input value of a and b:，提示用户从键盘输入 a 和 b 的值，这是由执行部分的第一行完成的。用户在提示下从键盘上键入两个数，如 5↵9↵（或 5 9↵），接着屏幕上会显示出计算结果：sum=14。

1.4 C 语言程序的结构

上面所述的 3 个 C 语言程序范例虽然功能简单，却包含了 C 语言程序的基本组成部分，概括来说，一个 C 语言程序可由下面几个不同的部分组合而成。

（1）文件包含部分。

（2）预处理部分。

（3）变量说明部分。

（4）函数原型声明部分。

（5）主函数部分。

（6）函数定义部分。

对上面所述 C 语言程序的 6 个部分，有以下几点需要说明。

（1）并不是每个程序都必须包含上面的 6 个部分，一个最简单的 C 语言程序可以只包含文件包含部分和主函数部分。

（2）每个 C 语言程序都必须有且仅有一个主函数，主函数的组成形式如下。

```
main()
{
    变量说明部分
    程序语句部分
}
```

（3）每个 C 语言程序可以有 0 个或多个自定义的函数，自定义函数的形式同主函数形式一样。

```
<函数的返回值类型> <函数名>(<参数列表>)
{
    变量说明部分
    程序语句部分
}
```

（4）每条语句均由分号结束。

通过后续章节的学习，读者可以了解 C 语言程序基本表达式、控制结构语句的作用，并通过了解模块化程序设计的思想和方法，掌握基本的 C 语言程序设计方法。

1.5 C 语言程序的执行

用程序语言编写的程序称为源程序（Source Program），实际上计算机本身并不能直接理解这样的语言，必须将程序语言翻译成机器语言。将源程序翻译成机器语言的过程称为编译，编译的结果是得到源程序的目标代码（Object Program）；最后还要将目标代码与系统提供的函数和自定义的过程（或函数）链接起来，以得到机器可执行的程序。机器可执行的程序称为可执行程序或执行文件。

1.5.1 源程序翻译

C 语言源程序的扩展名为.c。它是不能直接在计算机上运行的，必须先通过机器翻译成目标代码，再将目标代码链接成可加载模块（可执行文件）。这种把源程序翻译成目标代码的程序被称为编译器或翻译器。适合 C 语言的编译器不止一种，不同的机器、不同的操作系统可能会有一种或多种不同的编译器。C 语言源程序的翻译过程如图 1-1 所示。

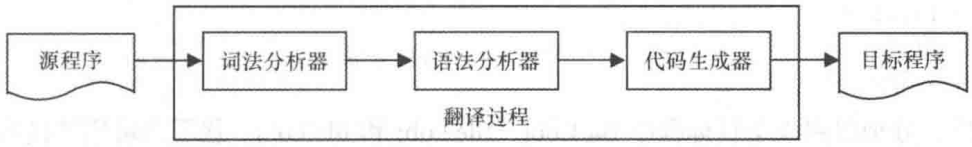

图 1-1 C 语言源程序的翻译过程

1. 词法分析器

词法分析器（Lexical Analyzer）主要是对源程序进行词法分析，它是按单个字符的方式阅读源程序，并且识别出哪些符号的组合可以代表单一的单元，并根据它们是否是数字值、单词（标识符）、运算符等，将这些单词分类。词法分析器将词法分析结果保存在一个结构单元里，这个结构单元称为记号（Token），并将这个记号交给语法分析器。词法分析会忽略源程序中的所有注释。

2. 语法分析器

语法分析器（Parser）直接对记号进行分析，并识别每个成分所扮演的角色。这些语法规则也就是程序设计语言的语法规则。

3. 代码生成器

代码生成器（Code Generator）将经过语法分析后没有语法错误的程序指令转换成机器语言指令。

例如，假定编写了一个名为 mytest 的程序，源程序的全名为 mytest.c，用 Microsoft C 编译器，在命令方式下，可以采用下面的方式对 mytest.c 进行编译。

```
cl -c mytest.c↵
```

如果源程序没有错误，就会生成一个名为 mytest.obj 的目标代码程序。其他程序语言也会有类似的命令将源程序翻译成目标代码，具体的命令与每种程序语言的编译器有关。

1.5.2 链接目标程序

通过翻译产生的目标代码程序尽管是机器语言的形式，却不是机器可以执行的方式，这是因为为了支持软件的模块化，允许程序语言在不同的时期开发出具有独立功能的软件模块作为一个单元，一个可执行的程序中有可能包含一个或多个这样的程序单元，这样可以降低程序开发的低水平重复所带来的低效率。因此，目标程序只是一些松散的机器语言，要获得可执行的程序，还需将它们链接起来。

程序的链接工作由链接器（Linker）完成。链接器的任务就是将目标程序链接成可执行的程序（又称载入模块），这种可执行的程序是一种可存储在磁盘存储器上的文件。

例如，假设对源程序 mytest.c 进行编译后生成了目标代码程序 mytest.obj，我们可以利用链接器生成可执行代码。在命令方式下，可用下面的方式来链接程序。

```
link /out:mytest.exe mytest.obj↵
```

如果不发生错误，将会生成一个名为 mytest.exe 的加载模块，也就是可执行的代码程序。最后，可以通过操作系统将这个加载模块加载到内存，执行程序的进程。

上面对程序进行编译、链接都只针对了一个源程序文件，实际上，可以将多个源程序文件通过编译、链接成一个可执行文件。

例如，假定有 3 个源程序 file1.c、file2.c 和 file3.c，每一个源程序都包含不同的函数或过程，在命令方式下可先用编译器对 3 个源程序进行编译。

```
cl  -c file1.c↵
cl  -c file2.c↵
cl  -c file3.c↵
```

编译后，分别得到 3 个目标程序 file1.obj、file2.obj 和 file3.obj。接下来可用链接器将 3 个目标程序进行链接。

```
Link /out:mytest.exe file1.obj file2.obj file3.obj↵
```

链接后，可得到一个可执行的程序 mytest.exe。

对于程序的编译、链接，有必要强调以下几点。

（1）并不是每一个目标程序都可以链接成可执行程序。

（2）被链接成可执行程序的目标程序，只允许在一个程序中有且只有一个可被加载的入口点，即只允许在一个源程序中包含一个 main()函数。在上面的范例中，这个可被加载的入口点在源程序 file1.c 中。

（3）对于具体的程序语言，编译、链接程序的方法会有所不同。针对某一种程序语言的编译器，不可以用于对另一种源程序语言进行编译。

（4）上面对 C 语言进行编译、链接的方式并不是唯一的，它允许有一些其他的变化，具体可参考各编译器的使用说明。

总之，完整的 C 语言程序生成过程主要包括 4 个部分——编辑、编译、链接和加载，如图 1-2 所示。

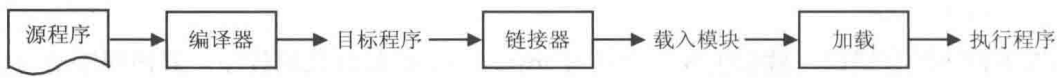

图 1-2　完整的 C 语言程序生成过程

一旦生成了可执行程序，就可以反复地被加载执行，而不需要重新编译、链接。如果修改了源程序，也不会影响到已生成的可执行程序，除非对修改后的源程序进行重新编译和链接，生成一个新的可执行程序。

1.5.3　集成开发工具

显然，用命令方式来编译、链接生成可执行的程序，并不是很方便，尤其是源程序的编辑。一般地，纯文本编辑器都可以输入源程序，如果编译时有错误，就必须回到编辑器修改程序。如此反复，源程序的编辑不是很方便，而且程序开发效率不高，目前我们已很少采用这种方法来编辑 C 语言源程序，大多采用集成开发工具来开发 C 语言程序。

程序的集成开发工具是一个经过整合的软件系统，它将编辑器、编译器、链接器和其他软件单元集合在一起。在这个工具里，程序员可以很方便地对程序进行编辑、编译、链接及跟踪程序