普通高等教育“十一五”国家级规划教材

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C语言程序设计教程（第2版）

The C Programming Language (2nd Edition)

李丽娟 主编

- 实例导入，案例丰富
- 培养算法设计的思想
- 同一问题提供多种解决方案

精品系列

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

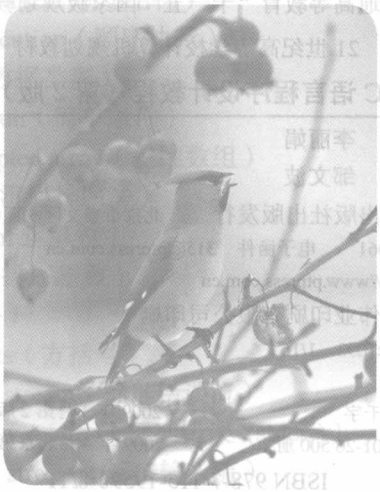
21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C语言程序设计教程（第2版）

The C Programming Language (2nd Edition)

李丽娟 主编



精品系列

人民邮电出版社

北京

图书在版编目（CIP）数据

C语言程序设计教程 / 李丽娟主编. —2版. —北京：人民邮电出版社，2009.3（2009.4 重印）
21世纪高等学校计算机规划教材
ISBN 978-7-115-19596-8

Ⅰ. C… Ⅱ. 李… 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（2009）第006830号

内 容 提 要

本书以 C 语言程序案例为导向，深入浅出地讲解了 C 语言程序设计的基本方法。全书注重算法设计与程序设计的关联性，强化模块化程序的设计方法。

全书内容可分为三部分，共 11 章。第一部分为第 1、2 章，是初学者的入门知识，简单介绍 C 语言的基础知识，主要内容有 C 语言程序的基本结构、数据类型和数据的存储方式、基本的程序表达式。第二部分为第 3 章～第 5 章，是程序设计的基础部分，主要内容有描述程序算法的方法、程序语句的基本控制结构。掌握了第一、二部分的内容，读者可以完成简单的程序设计。第三部分为第 6 章～第 11 章，是模块化程序设计的概念和实现的方法，主要内容有函数、数组、指针、结构、文件、位运算等。通过对这三部分知识单元的学习，读者可以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，内容叙述由浅入深。本书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。

普通高等教育“十一五”国家级规划教材

21 世纪高等学校计算机规划教材

C 语言程序设计教程（第 2 版）

◆ 主　　编　李丽娟
　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
　北京隆昌伟业印刷有限公司印刷

◆ 开本：787×1092　1/16
　印张：21
　字数：551 千字　　2009 年 3 月第 2 版
　印数：24 501-26 500 册　　2009 年 4 月北京第 2 次印刷

ISBN 978-7-115-19596-8/TP

定价：34.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

出版者的话

计算机应用能力已经成为社会各行业从业人员最重要的工作技能要求之一，而计算机教材质量的好坏会直接影响人才素质的培养。目前，计算机教材出版市场百花争艳，品种急剧增多，要从林林总总的教材中挑选一本适合课程设置要求、满足教学实际需要的教材，难度越来越大。

人民邮电出版社作为一家以计算机、通信、电子信息类图书与教材出版为主的科技教育类出版社，在计算机教材领域已经出版了多套计算机系列教材。在各套系列教材中涌现出了一批被广大一线授课教师选用、深受广大师生好评的优秀教材。老师们希望我社能有更多的优秀教材集中地呈现在老师和读者面前，为此我社组织了这套“21 世纪高等学校计算机规划教材——精品系列”。

本套教材具有下列特点。

（1）前期调研充分，适合实际教学需要。本套教材主要面向普通本科院校的学生编写，在内容深度、系统结构、案例选择、编写方法等方面进行了深入细致的调研，目的是在教材编写之前充分了解实际教学的需要。

（2）编写目标明确，读者对象针对性强。每一本教材在编写之前都明确了该教材的读者对象和适用范围，即明确面向的读者是计算机专业、非计算机理工类专业还是文科类专业的学生，尽量符合目前普通高等教育计算机课程的教学计划、教学大纲以及发展趋势。

（3）精选作者，保证质量。本套教材的作者，既有来自院校的一线授课老师，也有来自 IT 企业、科研机构等单位的资深技术人员。通过他们的合作使老师丰富的实际教学经验与技术人员丰富的实践工程经验相融合，为广大师生编写出适合目前教学实际需求、满足学校新时期人才培养模式的高质量教材。

（4）一纲多本，适应面宽。在本套教材中，我们根据目前教学的实际情况，做到“一纲多本”，即根据院校已学课程和后续课程的不同开设情况，为同一科目提供不同类型的教材。

（5）突出能力培养，适应人才市场要求。本套教材贴近市场对于计算机人才的能力要求，注重理论知识与实际应用的结合，注重实际操作和实践动手能力的培养，为学生快速适应企业实际需求做好准备。

（6）配套服务完善。对于每一本教材，我们在教材出版的同时，都将提供完备的 PPT 课件，并根据需要提供书中的源程序代码、习题答案、教学大纲等内容，部分教材还将在作者的配合下，提供疑难解答、教学交流等服务。

在本套教材的策划组织过程中，我们获得了来自清华大学、北京大学、中国人民大学、浙江大学、吉林大学、武汉大学、哈尔滨工业大学、东南大学、四川大学、上海交通大学、西安交通大学、电子科技大学、西安电子科技大学、北京邮电大学、北京林业大学等院校老师的大力支持和帮助，同时获得了来自信息产业部电信研究院、联想、华为、中兴、同方、爱立信、摩托罗拉等企业和科研单位的领导或技术人员的积极配合。在此，向他们表示衷心的感谢。

我们相信，“21 世纪高等学校计算机规划教材——精品系列”一定能够为我国高等院校计算机教学做出应有的贡献。同时，对于工作欠缺和不妥之处，欢迎老师和读者提出宝贵的意见和建议。

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。为适应我国计算机科学技术的应用和发展，进一步提高计算机程序设计课程的教学质量，作者根据多年的教学经验，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。

在 2002 年，作者曾编写出版了《C 程序设计基础教程》。随着教学要求的变化，作者在 2005 年对该书进行了一些修改，于 2006 年出版《C 语言程序设计教程》，被多所学校用作教材，前后多次印刷。

本书在继承前两种教材特色的基础上，结合作者多年的教学经验，并特别根据近几年教学改革的实践以及对人才培养的高标准要求，对其内容做了进一步的优化、补充和完善。通过近几年教学实践表明，在教学中较早引入算法的概念和设计算法的基本方法，有利于培养学生的综合应用能力，对培养工程应用型人才也是有益的。实践还表明，通过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。这些内容对于各类普通高校本科、专科学生也是适用的。

本书将 C 语言程序设计分成以下 3 个循序渐进的部分。

第一部分是入门基础，由第 1、2 章组成，主要介绍 C 语言程序的基本结构、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。这部分的内容奠定了 C 语言程序设计的基础，通过学习，读者可以设计由表达式语句组成的简单程序。这部分程序的结构主要是顺序结构。

第二部分是程序设计的基本结构，由第 3 章～第 5 章组成，主要介绍简单程序设计方法，简单算法的设计和算法的表示方法，以及两种常用的程序语句结构：分支结构和循环结构。通过学习，读者可以了解 C 语言程序结构主要由顺序结构、分支结构和循环结构 3 种基本结构组成，学会制订简单的算法并能依据算法思想编写程序，设计出语句结构较为复杂的程序，掌握用 C 语言程序结构的方式来思考问题和解决问题。

第三部分是模块化程序设计的结构，由第 6 章～第 11 章组成，主要介绍 C 语言程序的模块化功能。通过学习，读者可以了解模块化程序设计的思想，掌握程序模块的设计方法，并进一步通过指针、结构、文件、位运算等全面了解和掌握 C 语言程序设计的手段和方法，培养和锻炼开放性思考问题和解决问题的能力。

本书具有以下特色。

1. 实例导入，以程序为核心，由浅入深。

如在第 1 章中，通过对 C 语言程序基本结构和开发环境的介绍，读者可以认识 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，加强对 C 语言程序的感性认识。

2. 案例丰富，启发性强。

本书精选了 170 多个程序，大部分程序都在 Visual C++ 6.0 环境下通过验证（个别不在 Visual C++ 6.0 环境下的程序有特别说明），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。其中大量的程序案例留有可进一步探讨的余地，给教师的教学留下了广阔的空间，可以启发读者思考，从中发现问题，寻找解决问题的方法。从而不断激发读者的学习兴趣，激发想象力和创新思维能力。

3. 问题分析引导，算法流程图规范。

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，培养读者独立思考和解决问题的能力。

为了巩固所学的理论知识，本书每章都附有习题，以帮助读者理解基本概念，通过理论联系实际地进行书面练习和上机编写程序，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 语言程序设计教程习题解答与实验指导（第 2 版）》给出了本书中习题的全部参考答案和学生上机实验的内容。在实验中，读者可以通过编写程序，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，其中课堂教学时数为 40，上机实验学时为 48，各章的教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	课堂教学时数	上机实验时数
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	分支语句	2	4
5	循环语句	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、专升本的教材，也可作为研究生入学考试和各类认证证书考试的复习参考书，以及作为计算机应用工作者和工程技术人员的参考书。

本书由李丽娟任主编，吴蓉晖、杨小林、洪跃山、李根强、杜四春任副主编。感谢湖南大学计算机与通信学院李仁发教授对本书的支持和关心。

由于作者水平有限，书中难免存在不妥与疏漏之处，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，请登录人民邮电出版社教学服务与资源网：http://www.ptpedu.com.cn，从网上下载；或者与作者联系：jt_lljh@hnu.cn。

李丽娟

2008 年 12 月于岳麓山

目　录

第1章 引言

1.1 C语言的发展过程

C语言在20世纪70年代初问世。1978年美国电话电报公司（AT&T）贝尔实验室正式发表了C语言。同时B.W.Kernighan和D.M.Ritchie合著了著名的“*THE C PROGRAMMING LANGUAGE*”一书，通常简称为“K&R”，也有人称之为“K&R”标准。但是，在“K&R”中并没有定义一个完整的标准C语言，1983年美国国家标准协会（American National Standards Institute，ANSI）在此基础上开始设计C语言标准，并于1989年12月通过该标准，1990年，国际标准化组织（ISO）接受了ANSI提出的标准，这个C语言标准的C版本被称为C89，也被称之为ANSI C。

1.2 C语言的特点

C是一种结构化的程序设计语言，它简明易懂，功能强大，可使程序员不必关注程序在何种机器上运行，而致力于问题本身的处理。C语言集高级语言和低级语言的功能于一体，适合于各种硬件平台，既可用于系统软件的开发，也适合于应用软件的开发。

C语言具有丰富的运算符和数据类型，便于实现各种复杂类型的数据结构；它可以直接访问内存的物理地址，直接对硬件的底层操作，能实现汇编语言的大部分功能，因此，也有人把C语言称为中级语言；C语言还可进行位（bit）的运算，实现对数据的“位”操作。另外，C语言还具有效率高、可移植性强等特点。

1. 程序设计结构化

结构化的程序语言（或称为模块化语言）将程序的功能进行模块化，每一个模块具有不同的功能，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。C语言程序将一些不同功能的模块有机的组合在一起，这种模块化的程序设计方式使得C语言程序易于调试和维护。

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，从而使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. 数据结构类型丰富

C语言除了具有自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，书写程序时所受的限制并不严格（注意：编写程序时并不提倡这样做）。

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

关键字是已被程序语言本身使用的标识符，具有特定的意义，不能用作变量名、函数名等其他用途。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 大类：

（1）数据类型关键字 12 个。

（2）控制语句关键字 12 个。

（3）存储类型关键字 4 个。

（4）其他关键字 4 个。

这些关键字及作用如表 1-1 所示。

表 1-1　　C 语言的关键字及作用

类型	关键字	用途	关键字	用途	关键字	用途	关键字	用途
数据类型（12 个）	char	声明字符型变量或函数	float	声明浮点型变量或函数	short	声明短整型变量或函数	union	声明联合数据类型
	double	声明双精度变量或函数	int	声明整型变量或函数	signed	声明有符号类型变量或函数	unsigned	声明无符号类型变量或函数
	enum	声明枚举类型	long	声明长整型变量或函数	struct	声明结构体变量或函数	void	声明函数无返回值或无参数，声明无类型指针
控制类型（12 个）	for	循环语句	break	跳出当前循环或分支	else	条件语句否定分支（与 if 连用）	case	开关语句分支
	do	循环语句	continue	结束当前循环，开始下一轮循环	goto	无条件转移语句	default	开关语句中的“其他”分支
	while	循环语句	if	条件分支	switch	开关语句	return	函数返回语句
存储类型（4个）	auto	声明自动变量（可不用）	extern	声明外部变量	register	声明寄存器变量	static	声明静态变量
其他类型（4个）	const	声明只读变量	sizeof	计算数据类型长度	typedef	给数据类型取别名等	volatile	变量在程序执行中可被隐含地改变

不同的实现环境除了包含上述的关键字以外，还对 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，在此不多加叙述，这些扩充的关键字是不可以在不同的实现环境中通用的。

一般而言，C 语言的关键字不允许用户将其用作其他用途，在 C 语言中，所有关键字都是英文小写的。

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	无条件转移	for	循环语句	break	跳出循环或分支
if	条件分支	do	循环语句	continue	结束当前循环，开始下一轮循环
switch	多路分支	while	循环语句	return	返回

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 语言上述的特点，对学习和掌握好 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的限制，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，特别对于初学者，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，同时也便于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，但可从这些例子了解到组成一个 C 语言源程序的基本部分和书写格式。

【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>
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，其意义是把尖括号（〈 〉）内指定的文件包含到本程序中，成为本程序的一部分。这里被包含的文件是由系统提供的，其扩展名为.h，也称为头文件或首部文件。C 语言的头文件中包括了各种标准库函数的函数原型，因此，凡是在程序中调用一个库函数时，都必须包含该函数的原型所在的头文件。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，在此表示在程序中要用到这个文件中的函数。“#”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 C 语言源程序只允许有 1 个 main 函数。

3. printf 是函数调用语句，printf 函数的功能是把要输出的内容送到显示器上显示。printf 函数是一个由系统定义的标准函数，在 stdio.h 库函数中，可在程序中直接调用。

4. main()函数中的内容必须放在一对花括号“{}”中。

5. 程序运行后会在屏幕的左上方显示程序的运行结果。

【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>
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
```

请注意，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. 程序包含了两个头文件：stdio.h、math.h。
2. 在 main 函数中定义了两个双精度实数型变量 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 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。

在本例中，使用了 3 个库函数：输入函数 scanf，余弦函数 cos，输出函数 printf。余弦函数 cos 是数学函数，其头文件为 math.h，因此，在程序的主函数前用 include 命令包含了 math.h 文件。scanf 和 printf 是标准输入/输出函数，其头文件为 stdio.h，在主函数前也用 include 命令包含了 stdio.h 文件。

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，任何时候明确清楚的表达对程序的理解是有益的，一般情况下，C 语言程序都要求写上文件包含：#include<stdio.h>，这是一个良好的习惯，否则，会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);
main()
{
     int a, b, c;
     printf ("please input value of a and b:\n");
     scanf("%d %d", &a, &b);
```

```
    c=add(a,b);
    printf ("max=%d\n",c);
}
int add(int x, int y)
{
    return(x+y);
}
```

程序运行结果：

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

程序说明：

1. 在例 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设计语言的一个特点，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)；是通过调用加法器 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↵），接着在屏幕上会显示出计算结果：max = 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，但却包含了 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. 自定义函数部分。

对于上面所述 C 语言程序的 6 个部分，必须说明以下几点。

1. 并不是所有的 C 语言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的包含部分和主函数部分。

2. 每一个 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所示：

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

3. 每一个 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：

```
<自定义的函数名>(<参数列表>)
{
    变量说明部分
```

```
    程序语句部分
}
```

4. 每一个 C 语言程序的语句由分号结束。

对于 C 语言程序中各部分的作用、使用方法和采用的什么语句来完成，可以在后续章节中通过对基本表达式、结构控制语句的学习进一步掌握，并通过了解模块化设计等方面的内容，掌握 C 语言程序设计的思想。

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，计算机才能理解程序。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，就可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 语言源程序的后缀名为.c。它是不能直接在计算机上运行的，必须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），才能在计算机上运行。这种把源程序翻译成目标代码的程序被称之为编译器或翻译器。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 1 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，它由词法分析器、语法分析器和代码生成器 3 部分组成。

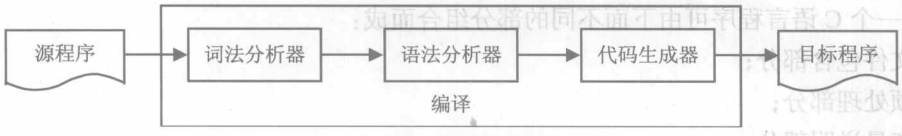


图 1-1 程序语言的翻译过程

1. 词法分析器（Lexical Analyzer）

词法分析器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（Parser）

语法分析器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（Code Generator）

代码生成器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，可以采用下面这样的方式对 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 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，但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）来完成。链接器的任务就是将目标程序链接成可执行的程序（或称载入模块），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，假设已对源程序 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，将用下面这样的方式来链接程序：

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

如果不发生错误，就会生成一个名为 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入内存，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有不同的函数或过程，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
```

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
```

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. 并不是任何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中，只允许在一个程序中有且仅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的编译。

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，完成一个 C 语言程序的完整过程主要包括 4 个部分：编辑、编译、链接和加载，如图 1-2 所示。

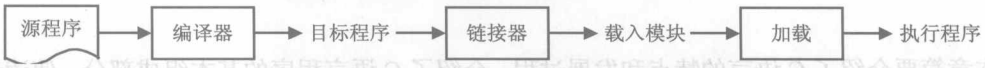


图 1-2　完整的程序生成过程

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、链接以及跟踪程序的执行过程，以便寻找程序中的问题。

适合 C 语言的集成开发工具有许多，如 Turbo C、Microsoft C、Visual C++、Dev C++、Borland C++、 C++ Builder、Gcc 等。这些集成开发工具各有特点，分别适合于 DOS 环境、Windows 环境和 Linux 环境，几种常用的 C 语言开发工具的基本特点和所适合的环境如表 1-3 所示。

表 1-3　　几种常用的 C 语言开发工具

开 发 工 具	运 行 环 境	各工具的差异	基 本 特 点
Turbo C	DOS	不能开发 C++语言程序	（1）符合标准 C （2）各系统具有一些扩充内容 （3）能开发 C 语言程序（集程序编辑、编译、链接、调试、运行于一体）
Borland C	DOS		
Micrsoft C	DOS		
Visual C++	Windows	能开发 C++语言程序（集程序编辑、编译、链接、调试、运行于一体）	
Dev C++	Windows		
Borland C++	DOS、Windows		
C++ Builder	Windows		
Gcc	Linux		

从表 1-3 中可以看出，有些集成开发工具不仅仅适合于开发 C 语言程序，还适合开发 C++语言程序。这些既适合于 C 语言又适合于 C++语言的开发工具，一开始并不是为 C 语言而写的，而是为 C++语言设计的集成开发工具，但因为 C++语言是建立在 C 语言的基础之上，C 语言的基本表达式、基本结构和基本语法等方面同样适合 C++语言，因此，这些集成开发工具也能开发 C 语言程序。

事实上，表 1-3 所示的几种适合于 DOS 环境的集成开发工具，也有适合于 Windows 环境的版本，但在计算机的图形用户界面还没有像现在这样普遍和成熟的时候，人们大多都是采用 DOS 环境下集成开发工具，如 Turbo C 2.0，它既能作为初学者的一个实验工具，同时还是很多专业人员进行程序开发的首选工具。

另外，还有一些小型的集成开发工具比较适合于学习者使用，如 Turbo C/C++ for Windows，它支持 C、C++、Windows C 等程序的编辑、编译、调试、运行，为 C 语言的学习提供了方便。

1.6 本 章 小 结

本章简要介绍了 C 语言的特点和发展过程，介绍了 C 语言程序的基本组成部分，使读者对 C 语言程序的组成部分有一个初步的了解；还介绍了 C 语言程序的开发过程：

1. 编辑源程序——主要是编写源程序；