



普通高等院校“十三五”规划教材



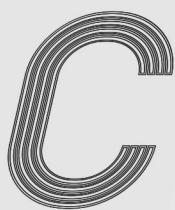
# 语言程序设计

■ 主 审 阳王东  
■ 主 编 何 骞 何建新

**CMS** 湖南教育出版社



普通高等院校“十三五”规划教材



# 语言程序设计

■ 主 审 阳王东  
■ 主 编 何 骞 何建新  
■ 副主编 谭新良 刘德峰 阳 俊  
■ 参 编 王 科 胡奇光 吴 鸣 周建存  
李正华 莫 照 李 旒 周翠红

**CMS** 湖南教育出版社

## 图书在版编目 (CIP) 数据

C 语言程序设计 / 何骞, 何建新主编. — 长沙 :  
湖南教育出版社, 2018.9  
ISBN 978-7-5539-6326-6

I. ①C… II. ①何… ②何… 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 200137 号

策划编辑	赵爱平	责任编辑	刘文婷
封面设计	袁敏	排版设计	邹满辉

---

书 名: C 语言程序设计  
CYUYAN CHENGXU SHEJI  
主 编: 何 骞 何建新  
出版发行: 湖南教育出版社  
地 址: 长沙市韶山北路 443 号  
网 址: www.bakclass.com  
发 行 部: 0731-85520531  
编 辑 部: 0731-85558352 csgaojiao@163.co m  
印 刷: 湖南贝特尔印务有限公司  
经 销: 湖南天易创图文化有限公司

---

开 本: 787×1092 1 /16  
印 张: 16.25  
字 数: 375 千字  
版 次: 2018 年 9 月第 1 版 2 018 年 9 月第 1 次印刷

---

书 号: ISBN 978-7-5539-6326-6  
定 价: 46.80 元

---

本书如有印刷、装订错误, 可向承印厂调换

# 内 容 提 要

本书共 8 章。第 1 章主要讲解了 C 语言的特点和常见开发环境的搭建。第 2~5 章主要讲解了 C 语言的基础知识，包括数据类型与运算符、流程控制语句、函数和数组等，学好这些知识对后续的学习至关重要。第 6~8 章是 C 语言的核心内容，包括指针、结构体和文件等，只有熟练掌握了这些知识，才算真正学好了 C 语言。

本教材可作为高等学校各专业学习 C 语言的正式教材，也是一本自学的好教材。本书还配有辅助教材《C 语言程序设计实践》。

# 前 言

C 语言是一门通用的计算机编程语言，是编程者迈入专业编程的第一道门槛，其地位不言自明。为了让读者能够轻松理解并快速掌握，本教材对每个知识点都进行了深入的分析，并针对每个知识点精心设计了相关案例，真正做到了知识的由浅入深、由易到难。

本书共分为 8 章，具体如下：

第 1 章主要介绍了 C 语言的特点和常见开发环境的搭建。通过本章的学习，读者应掌握常见开发工具的安装与使用，并动手实现第一个 C 语言程序。

第 2~5 章主要讲解 C 语言的基础知识。在学习这些知识时，读者应掌握三种基本结构、函数和数组等内容，灵活掌握这些基本知识有利于后面知识点的学习。

第 6~8 章是 C 语言最核心的内容，读者需要掌握指针、结构体和文件等知识点。只有熟练掌握了这些知识，才算真正学好了 C 语言。

本教材由何骞、何建新担任主编，谭新良、刘德峰和阳俊担任副主编，阳王东担任主审。其中第 1 章由胡奇光和周翠红编写，第 2 章由王科和李旒编写，第 3 章由何骞、周建存和吴鸣编写，第 4 章由何建新编写，第 5 章由谭新良编写，第 6 章由何骞和李正华编写，第 7 章由刘德峰和莫照编写，第 8 章由阳俊编写。

为了帮助读者更好地掌握 C 语言知识，做到学练结合、学以致用，建议与实践教材《C 语言程序设计实践》配套使用。

本书的作者都是多年从事教学的一线教师，试图努力讲解好每个知识点，但水平有限，书中难免有疏漏，敬请读者批评指正，以便再版时修订完善。

编 者

2018 年 4 月

# 目 录

第 1 章 C 语言概述.....	1
1.1 什么是计算机语言 .....	1
1.2 C 语言的历史及特点.....	3
1.3 C 语言程序的组成部分 .....	5
1.4 C 语言程序的开发步骤.....	9
1.5 C 语言的开发环境.....	10
1.6 第一个 C 语言程序的运行.....	15
第 2 章 数据类型与运算符.....	23
2.1 C 语言字符集、标识符与关键字.....	23
2.2 常量与变量 .....	24
2.3 运算符与表达式 .....	34
第 3 章 结构化程序设计.....	49
3.1 算法——程序的灵魂.....	49
3.2 C 语言的基本语句 .....	54
3.3 数据的输入与输出 .....	57
3.4 顺序结构程序设计 .....	65
3.5 选择结构程序设计 .....	68
3.6 循环结构程序设计 .....	78
第 4 章 函数.....	98
4.1 为什么要用函数 .....	98
4.2 函数的定义 .....	100
4.3 函数的调用 .....	102
4.4 函数的嵌套调用与递归调用.....	107
4.5 变量作用域与存储方式.....	111
4.6 内存四区 .....	120
4.7 函数应用举例 .....	122
第 5 章 数组.....	125
5.1 一维数组 .....	125
5.2 二维数组 .....	131
5.3 字符数组和字符串 .....	136
5.4 数组作为函数的参数.....	144
5.5 程序举例 .....	148

第 6 章 指针.....	153
6.1 指针与指针变量.....	153
6.2 指针与函数.....	157
6.3 指针与数组.....	163
6.4 指针与字符串.....	169
6.5 指针数组与命令行参数.....	172
6.6 动态内存分配.....	176
6.7 程序举例.....	178
第 7 章 结构体与共用体.....	183
7.1 结构体类型的定义.....	183
7.2 结构体类型变量.....	185
7.3 结构体类型数组.....	193
7.4 结构体类型指针.....	197
7.5 链表.....	202
7.6 共用体类型与共用体类型变量.....	216
7.7 枚举类型.....	221
7.8 用 typedef 声明新类型.....	222
7.9 程序举例.....	224
第 8 章 文件.....	227
8.1 文件概述.....	227
8.2 文件操作.....	231
附录 I ASCII 码字符集.....	242
附录 II 运算符的优先级和结合性一览表.....	243
附录 III C 语言常用库函数.....	245
参考文献.....	252

# 第 1 章 C 语言概述

编程，简单来说就是为了让计算机理解人想让它干什么而编写指令，即编写程序。具体来说就是为了使计算机能够理解人的意图，人就必须将需要解决的问题的思路、方法和手段通过计算机能够理解的形式告诉计算机，使得计算机能够根据人的指令一步一步去工作，完成某种特定的任务。为了实现编程，人创造了人和计算机都能识别的计算机语言。

## 1.1 什么是计算机语言

计算机语言是人与计算机之间进行交流的语言，人可以通过计算机语言来控制计算机，让计算机为人做事情。人类语言，如汉语言、英语等，都有固定的格式、词汇和语法规则。同样，计算机语言也有固定的格式、词汇和语法规则，人必须经过学习才会使用，才能控制计算机。

计算机语言有很多种，但他们的功能和作用是不尽相同的，按照人与计算机的交互程度，可以分成机器语言、汇编语言和高级语言。

### 1.1.1 机器语言

机器语言是计算机所能识别，不需要翻译直接供机器使用的计算机语言。它在形式上是构成的一串二进制代码，每种计算机都有自己的一套机器指令。用机器语言编写程序，程序员要熟记计算机的全部指令代码和代码的涵义，要自己处理每条指令和每一数据的存储分配和输入输出，还要记住编程过程中每步所使用的工作单元处在何种状态，这是一件十分繁琐的工作。

机器语言是第一代计算机语言。用机器语言编写的程序全是由“0”和“1”组成的指令代码，直观性差，很容易出错。同时，机器语言与人所习惯的语言（如自然语言、数学语言等）差别很大，难学、难记、难读，因此很难用来开发实用的计算机程序。

### 1.1.2 汇编语言

为了克服机器语言难读、难编、难记和易出错的缺点，人们采用一定的助记符来代替机器语言中的指令和数据。如用“ADD”表示加法，用“SUB”表示减法，用“MOV”表示数据传递等；同时又用变量取代各类地址，如用“A”取代地址码等。这就形成了汇编语言，它是一种用助记符表示的仍然面向机器的计算机语言。

用汇编语言编写的程序称为汇编语言源程序。这种程序必须经过翻译（称为汇编），变成机器语言才能被计算机识别和执行。汇编语言在一定程度上克服了机器语言难于辨认和记忆的缺点，但对大多数用户来说，仍然是很难理解和使用的。

机器语言和汇编语言都是面向机器的计算机语言，它们都属于低级语言。

### 1.1.3 高级语言

为了克服低级语言的缺点，人们发明了更加直观易懂的高级语言。高级语言是与人类自然语言相接近，能为计算机所接受的语意确定、规则明确、自然直观和通用易学的计算机语言。

高级语言是一种比较接近于自然语言和数学表达式的语言，具有学习容易、使用方便、通用性强、移植性好的特点，便于各类人员学习和应用。高级语言并不特指的某一种具体的语言，其品种非常多，如目前常用的 C、C++、C#、Java、PHP、Python、Basic、Fortran、Ruby、Perl 等。

按照编程思想的不同，高级语言编程可分为面向过程和面向对象两大类。

#### 1. 面向过程编程

面向过程编程是一种以过程为中心的编程思想，分析出解决问题的步骤，然后用函数把这些步骤一步一步实现，代表性编程语言有 C 和 Fortran。面向过程倾向于做一件事的流程，先做什么，然后做什么，最后做什么。在面向过程编程中，数据和对数据的操作是分离的。

#### 2. 面向对象编程

面向对象编程是将事物对象化，通过对象通信来解决问题，代表性编程语言有 Java、C#。面向对象倾向于建立一个对象模型，它能够近似地反映实体与实体之间的关系，它具有封装、继承、多态三大特点。在面向对象编程中，数据和对数据的操作是绑定在一起的。

通俗讲，面向过程的思想是把一个项目按照一定的顺序，从头到尾一步一步地做下去，先做什么，后做什么，一直到结束，这其实是一个人做事的方法。面向对象的思想是把一个项目分成更小的项目，每一小项目负责一方面的功能，最后再由这些小项目组合成一个整体，这其实是多人分工合作做事的方法。

### 1.1.4 语言处理程序

用汇编语言和高级语言编写的程序称为源程序，不能被计算机直接执行，必须把它们翻译成机器语言程序，计算机才能识别和执行。这种翻译也是由程序实现的，不同的语言有不同的翻译程序，统称为语言处理程序。常用的翻译方式有解释方式和编译方式两种。

#### 1. 解释方式

在高级语言源程序输入计算机后，启动解释程序，翻译一句执行一句，直到程序执行完为止。如 Basic、Ruby、Perl、Python 等。采用解释方式实现翻译的语言称为解释型语言。

## 2. 编译方式

将高级语言源程序输入计算机后，调用编译程序将其整个翻译成用机器指令表示的目标程序，然后执行目标程序，得到计算结果。如 C、C++、C#、Java、Fortran 等。采用编译方式实现翻译的语言称为编译型语言。

如图 1.1 所示，表明了两种翻译方式的工作方式。

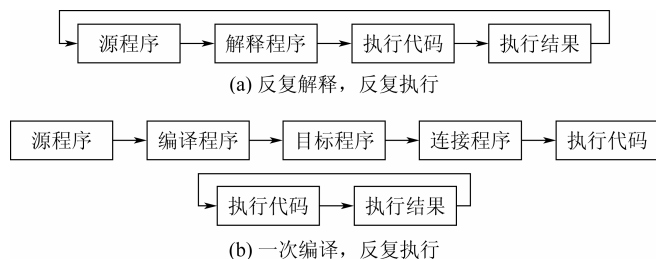


图 1.1 语言处理程序的两种方式

### 1.1.5 学习编程为什么首选 C 语言

目前，对于大部分程序员，C 语言是学习编程的第一门语言。C 语言能让我们了解编程的相关概念，带我们走进编程的大门。C 语言还能让我们明白程序的运行原理，比如，计算机的各个部件是如何交互的，程序在内存中是一种怎样的状态，操作系统和用户程序之间有着怎样的关系等。

C 语言概念少，词汇少，包含了基本的编程元素，后来的很多语言（如 C++、Java 等）都参考了 C 语言，可以说 C 语言是现代编程语言的基础。如果我们学习过 C 语言，就能更快地学习现在的其他高级编程语言。C 语言已经存在很多年了，它有广泛的使用团体和大量的现存代码可以利用。这就使我们能在过去程序的基础上，快速和高效地编写新的算法和函数。

C 语言属高级语言范畴，但它既具有高级语言的特点，又具有汇编语言的特点，可以进行较为底层的编程，能够直接操作计算机硬件，让我们更好地了解计算机。虽然 C 语言是一种较老的计算机语言，但它仍然被广泛应用于系统编程，编写其他语言及嵌入式系统之中。C 语言是唯一一个向我们阐述指针的本质的计算机语言，而对指针的掌握，是高级程序员的必要修为。

## 1.2 C 语言的历史及特点

### 1.2.1 C 语言的历史

1970 年，美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础设计出 B 语言，并且用 B 语言写了第一个 UNIX 操作系统。

1972 年，美国贝尔实验室的 Dennis M.Ritchie 在 B 语言的基础上设计出了 C 语言。C

语言保持了 B 语言精练、接近硬件的优点，又克服了 B 语言过于简单，数据无类型的缺点。

1973 年，Ken Thompson 和 Dennis M.Ritchie 用 C 语言完全重写了 UNIX。从此，C 语言成为编写操作系统的主要语言。随着 UNIX 的发展，C 语言开始流传开来。

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》，极大地推动了 C 语言的发展。

1983 年，美国国家标准学会（American National Standards Institute, ANSI）成立了一个委员会，来制定 C 语言标准。

1989 年，ANSI 发布了第一个完整的 C 语言标准——ANSI X3.159-1989，简称“C89”，遵循这个标准的 C 语言通常被称为 ANSI C。

1990 年，ANSI C 标准被国际化标准组织（ISO）采纳，形成 ISO9899-1990 标准，简称“C90”，通常也被称为 ISO C。目前，这些标准被绝大多数编译器支持。

1999 年，在做了一些必要的修正和完善后，ISO 发布了新的 C 语言标准，简称“C99”。2011 年，ISO 又发布了新的标准，简称为“C11”。

C 语言从出现到现在，在不断地完善，直到今天，已经成为最重要的计算机语言之一。

## 1.2.2 C 语言的特点

### 1. C 语言功能强大

C 语言功能强大，是专业程序员的首选语言。大多数软件开发商都是首选 C 语言开发操作系统，同时，C 语言也可用于开发字处理器、图形、电子表格等应用程序，其他语言（如 Python、FORTRAN、Perl 等）的许多编译器和解释器也是由 C 语言开发的。

### 2. C 语言可移植

C 语言移植性好，可适用于不同系统平台，代码移植时只需做较少修改或不做修改。因此，在使用 Windows 操作系统的计算机上编写的 C 程序，只需做少量的修改或不修改，就可以被移植到使用 Linux 操作系统的计算机上。

### 3. C 语言具有低级语言优势

C 语言具有低级语言的特点，具有与汇编语言相近的功能和描述方法，能直接访问计算机硬件（如地址运算、二进制数位运算等），对硬件端口等资源直接操作，可充分使用计算机资源。因此，C 语言既具有高级语言便于学习和掌握的特点，又具有机器语言或汇编语言对硬件操作的能力。

### 4. C 语言使用灵活

C 语言简洁紧凑，运算符丰富，数据类型丰富，表达方式灵活实用。C 语言中还提供了大量的函数，其中包括系统生成的库函数和用户自定义的自定义函数。这些，使得用 C 语言编写程序，高效、快速并易于理解。

### 5. C 语言采用模块化设计

C 语言是一种融合了控制特性的现代语言，这种设计方式使得用户可以自然地采用自

顶向下的规划，结构化的编程以及模块化的设计。这种做法使得编写出的程序更可靠，更易懂。模块化结构，使程序调试、测试和维护变得更加容易。

## 1.3 C 语言程序的组成部分

### 1.3.1 一个最简单的 C 语言程序

所有的 C 语言程序都是由多个部分组合而成。为了帮助说明 C 语言程序的整体情况，本节首先介绍一个最简短（但完整）的 C 语言程序并说明其各个组成部分。

**例 1.1** 在屏幕上显示“hello world!”。

```
#include <stdio.h>
int main()
{
    printf("hello world!");
    return 0;
}
```

运行结果为：

hello world!

**说明：**

（1）#include 编译指令和 stdio.h 头文件（第 1 行）。

第 1 行：#include <stdio.h>

字符#是预处理标志，#后面的 include 是预处理指令。编译指令#include 用来命令编译器，在编译时将指定的头文件添加到程序中。#include 的后面跟着一对尖括号，尖括号内的 stdio.h 就是指定的头文件的名称。头文件 stdio.h 中声明了第 4 行代码中的标准输出函数 printf。

预处理标志表示该行代码要最先进行处理，它总是在编译器编译代码之前运行。

头文件 stdio.h 是 standard input/output header（标准输入输出头文件）的缩写，它包含了有关输入输出函数的信息以供编译器使用，printf 函数就是其中的一个。在 C 语言程序中，总是要包含一个或多个头文件。

（2）main 函数（第 2~6 行）。

第 2 行：int main()

这行代码声明了一个 main 函数，int 指明了 main 函数的返回类型。main 函数的返回值将返回给操作系统。int 注明了 main 函数返回值的类型是整数。main 是函数的名字，其后面的一对圆括号“( )”用来说明 main 是个函数。圆括号内为空，表示没有给 main 函数传递任何信息。

在最简单的情况下，main 函数由函数名 main，紧跟 main 后面的一对圆括号“( )”和

一对花括号“{}”组成。

main 函数又称为主函数，C 语言程序总是从 main 函数开始执行。一个 C 语言程序有且只有一个 main 函数。

(3) 左花括号“{” (第 3 行)。

第 3 行: {

左花括号“{”表示 main 函数的开始，它和第 6 行的右花括号“}”对应。这对花括号划定了 main 函数的界线，花括号内的语句组成了 main 函数的主体。通常，所有 C 语言的函数都使用花括号“{}”来表示函数体的开始与结束，仅有花括号“{}”能起到这种作用，圆括号“()”和中括号“[]”都不行。

(4) printf()语句 (第 4 行)。

第 4 行: printf("hello world!");

这行代码使用标准库中的 printf 函数来输出一行信息到屏幕上，printf 后面的一对圆括号“()”表明 printf 是一个函数名。圆括号中包括的内容是从 main 函数传递到 printf 函数的信息，这样的信息被称为参数，如“hello world!”就是参数（前后的双引号也是参数的一部分），C 语言程序将识别这对双引号之间的内容并把它们显示在屏幕上。

第 4 行的最后面是一个分号“;”，表示这行代码是语句。在 C 语言程序中，语句通常占一行，并以分号结束。分号是 C 语句的必要组成部分，语句的最后必须要有一个分号（很多初学者经常忘记这个分号）。

(5) return 语句 (第 5 行)。

第 5 行: return 0;

return 语句是返回语句，作用是从函数中返回一个值，返回 return 后面的 0，在程序结束前将 0 这个值返回给操作系统。return 语句是 main 函数的最后一个语句，执行它将终止 main 函数的运行。

return 语句包括 return，后面紧跟着要返回的值，然后是一个分号。如果函数返回值类型为 void（表示函数没有返回值），就直接写 return，然后是一个分号。

(6) 右花括号“}” (第 6 行)。

第 6 行: }

右花括号“}”表示 main 函数的结束，它和第 3 行的左花括号“{”对应。

C 语言代码中所有的符号都必须是英文符号，如圆括号、花括号、分号等。

### 1.3.2 一个复杂点的 C 语言程序

本节介绍一个复杂点的 C 语言程序，用于详细分析 C 语言程序的具体内涵和细节。该程序示例在此用来分析 C 语言程序的组成部分，暂时不要求读懂。

例 1.2 求一个圆的面积。

```
#include <stdio.h>
```

```
#include <math.h>
#define PI 3.14 //定义圆周率 PI 为符号常量

/*
double circleArea(int r)是自定义函数
功能：返回半径为 r 的圆的面积
*/
double circleArea(int r);

int main()
{
    double area = 0;
    area = circleArea(3); //圆半径为 3
    printf("area=%f\n",area);
    return 0;
}

double circleArea(int r)
{
    /*pow()是库函数*/
    double area = 2 * PI * pow(r,2);
    return area;
}
```

运行结果为：

```
area=56.520000
```

### 说明：

(1) 预处理。第 1 行，`#include` 指令，`stdio.h` 包含第 15 行的 `printf` 函数的信息。第 2 行，`#include` 指令，`math.h` 包含第 22 行的 `pow` 函数的信息。第 3 行，`#define` 指令，`#define` 是 C 语言中提供的宏定义命令，以后程序中出现的 `PI` 都将以 3.14 代替。

**注意：**第 1、2、3 行的末尾都没有分号 (;)。

(2) `main` 函数。`main` 函数位于程序示例的第 11~17 行。一个 C 语言程序有且只有一个 `main` 函数，C 语言程序总是从 `main` 函数开始执行。一般情况下，C 语言程序从 `main` 函数的第一条语句开始执行，到 `main` 函数的最后一条语句结束。

(3) 程序注释。第 5~8 行是注释，第 21 行是注释，第 14 行//后面的内容也是注释。注释是在程序中用来标记一些特殊的文本，使用注释的目的是使人们更容易理解这个程

序，注释的文本不参与编译运行。

注释分为多行注释和单行注释，多行注释是以/\*开始到\*/结尾，单行注释是以//开始的单独一行。多行注释不能嵌套使用。

(4) 函数原型。第 9 行是自定义函数 circleArea 的函数原型，circleArea 是函数的名字，circleArea 后面紧跟着一对圆括号。circleArea 的前面是 double，double 说明 circleArea 函数将返回一个实数。圆括号内的 int r，说明 circleArea 函数将接收一个整数作为参数，实际意义是 circleArea 函数接收一个整数作为圆的半径。

函数原型是将自定义函数的函数名称和函数参数告诉编译器。函数原型代码行的末尾有一个分号“;”。

(5) 函数定义。第 19~24 行是自定义函数 circleArea 的函数定义。C 语言的函数分库函数和自定义函数，库函数是 C 语言自带的，自定义函数是程序员在开发过程中编写的，库函数和自定义函数没有本质的区别，只是开发者不同。

函数定义与函数原型不同，函数定义包含了组成函数的实际语句。

函数包括函数首部和函数体两部分，第 19 行是函数首部，第 20~24 行是函数体。

(6) 语句。第 9 行、第 13~16 行、第 22~23 行都是 C 语言程序的语句，语句以分号作为结束标志，C 语言程序的具体工作都是由语句完成的。C 语言程序的语句分为声明语句和执行语句，第 13 行是声明语句，第 14~16 行是执行语句。单一的分号也算一条语句，称为空语句，作用是什么也不执行。

(7) 花括号。C 语言使用花括号“{}”将函数中的语句聚集在一起。花括号是成对出现的，以左花括号表示函数体的开始，右花括号表示函数体的结束。可用花括号括起一条或多条语句聚集成代码块。第 12~17 行、第 20~24 行是代码块。

(8) 关键字。关键字是 C 语言中的词汇，有着特殊的含义，不能做其他用途。第 9 行的 double、第 11 行的 int、第 16 行的 return 都是关键字。

(9) 运算符。在 C 语言中，运算符是指可以进行运算操作的符号。第 13 行的=是赋值运算符，第 22 行的\*是乘法运算符。

(10) 表达式。在 C 语言中，一切计算结果为数值的东西都可以成是表达式，表达式总能返回一个值。第 13 行的 area = 0 是表达式。C 语言中的所有运算都通过表达式来实现。

(11) 转义字符。

转义字符是一种以“\”开头的字符，转义字符中的“\”表示它后面的字符已失去它原来的含义，转变成另外的特定含义。例如用“\n”表示换行符。第 15 行的\n 就起到回车并换行的作用。

(12) return 语句。

第 16 行和第 23 行包含了 return 语句。第 16 行的 return 语句在程序结束前，将 0 返回给操作系统。第 23 行的 return 语句是 circleArea 函数的一部分，它将圆面积的值（存储在 area 中）返回给调用了 circleArea 函数的 main 函数。

以上结合程序示例 `circle_area.c`，简要介绍了 C 语言程序的不同组成部分及其使用方法，这些是 C 语言程序的编程基础，是步入 C 语言程序设计领域的第一步。

## 1.4 C 语言程序的开发步骤

在理想化的情况下，C 语言程序的开发步骤分为编辑源代码、编译源代码、生成可执行文件和运行程序四步。

### 1.4.1 编辑源代码

C 语言源代码是一系列 C 语句或命令，指导计算机执行特定的任务。大多数 C 语言程序的集成开发环境（Integrated Development Environment，IDE）提供了编辑器，用来编写 C 语言源代码。用编辑器编写好源代码后，以文本的方式保存，扩展名为 `.c`，就成为了 C 语言的源程序文件（简称源文件），如前面展示的程序示例 `helloworld.c`、`circle_area.c` 等。

### 1.4.2 编译源代码

绝大多数的 C 语言程序的 IDE 自带有编译器。编译器是一个程序，其功能是将源代码转换为目标文件。编译器将文本方式的 C 源文件编译成二进制的机器语言指令，并生成扩展名为 `.obj` 或 `.o` 的目标文件，目标文件与源文件同名。

### 1.4.3 生成可执行文件

源代码要经过编译和链接两个过程才能变成可执行文件。通过链接器，将一个 C 程序的目标文件和要用到的库文件以及要用到的其他目标文件一起链接起来，生成扩展名为 `.exe` 的可执行文件，这个可执行文件与源文件同名。

### 1.4.4 运行程序

可执行文件是可以运行的一个程序，可以通过单击或双击文件名或图标直接从操作系统运行。在 C 语言程序的 IDE 中，也能够通过选择选项或按下某特殊键来执行 C 程序。

如图 1.2 所示，说明了源文件生成可执行文件的过程。

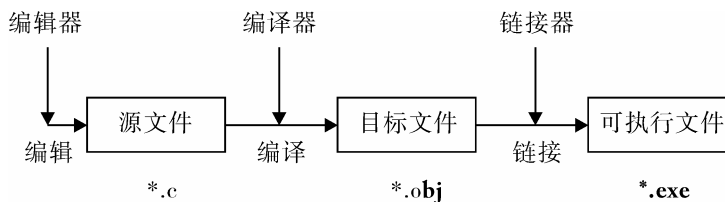


图 1.2 源文件生成可执行文件的过程

## 1.5 C 语言的开发环境

在 C 语言程序的实际开发中,除了编译器是必要的工具,往往还需要很多其他辅助工具(如编辑器、调试器、文件管理等),这些工具通常被打包在一起,统一发布和安装。有 Windows 平台下的 Visual C++ 6.0、Visual Studio、Code::Blocks、C-Free 等;有 Linux 平台下的 GCC 等;有 Mac OS X 平台下的 Xcode,它们统称为 IDE。

### 1.5.1 主流开发工具的介绍

这里主要介绍四款 Windows 平台下的 IDE。

#### 1. Visual C++ 6.0

Microsoft Visual C++ 6.0(简称 VC 6.0)是美国微软公司开发的一款很经典的 IDE,是一个基于 Windows 操作系统的可视化集成开发环境。VC 6.0 具有界面简洁、占用资源少、操作方便等优点,很多高校都以 VC 6.0 为教学工具来讲解 C 语言,教育部门的考证中还沿用 VC 6.0。但 VC 6.0 是毕竟 1998 年的产品,在 Win7、Win8、Win10 平台下会存在兼容性问题,甚至不能运行。

#### 2. Visual Studio

Microsoft Visual Studio(简称 VS)是美国微软公司的开发工具包系列产品,是 VC 6.0 的升级版,增加了很多新特性,支持了更多的语言,是 Windows 平台下的标准 IDE。VS 是一个基本完整的开发工具集,它包括了整个软件生命周期所需要的大部分工具,如 UML(Unified Modeling Language)工具、代码管控工具、IDE 等。所写的目标代码适用于微软支持的所有平台,包括 Microsoft Windows、Windows Mobile、Windows CE、.NET Framework、.NET Compact Framework、Microsoft Silverlight 和 Windows Phone。

为了适应最新的 Windows 操作系统,微软每隔一段时间就会对 VS 进行升级。VS 的不同版本以发布年份命名,例如 VS 2013 是微软于 2013 年发布的,VS 2017 是微软于 2017 年发布的。

VS 具有界面清晰、操作简单、代码自动补全强大、语法高亮、自动编译等优点,程序员大都使用 VS 进行 C 语言程序的实际开发。

#### 3. Code::Blocks

Code::Blocks(简称 CodeBlocks)是一个开放源码的、全功能的、免费的、跨平台的 C/C++ 集成开发环境。CodeBlocks 支持语法彩色醒目显示,支持代码完成,支持工程管理、项目构建、调试。它支持十几种常见的编译器,安装后占用较少的硬盘空间,功能十分强大,而且易学易用。

CodeBlocks 是一个轻量级的 IDE,对于 C 语言程序的初学者而言,是一个很好的选择。

#### 4. C-Free

C-Free 是一款国产的 Windows 下的 C/C++ IDE,使用者可以轻松地编辑、编译、链接、