面向21世纪信息管理与信息系统专业
东南大学出版社 核心课程系列教材

C/C++ 程序设计语言

顾/主编

C/C++程序设计语言

成颖　主编

东南大学出版社

内容提要

本书向读者介绍主要支持结构化程序设计的C语言，以及在C语言基础上进行扩展的支持面向对象程序设计的C++语言。在介绍C/C++语言的同时，还介绍了结构化程序设计方法以及面向对象程序设计方法的主要内容。本书侧重于基本概念、基本理论和基本方法的介绍，并通过实例培养读者分析问题和解决问题的能力，适用范围较广，可作为高等学校信息管理专业或其他非计算机专业学生的C或C++语言程序设计教材，也可作为电大、函大、自考及各类培训班教材。

图书在版编目（CIP）数据

C/C++ 程序设计语言 / 成颖主编. —南京：东南大学出版社，2002.12
ISBN 7-81089-101-4

Ⅰ.C… Ⅱ.成… Ⅲ.C语言—程序设计—高等学校—教学参考资料 Ⅳ.TP312

中国版本图书馆CIP数据核字（2002）第094275号

东南大学出版社出版发行
（南京四牌楼2号 邮编210096）
出版人：宋增民
江苏省新华书店经销 南京京新印刷厂印刷
开本：B5 印张：30 字数：639千字
2003年1月第1版 2003年1月第1次印刷
印数：1-3000 定价：36.00元

（凡因印装质量问题，可直接向发行科调换。电话：025-3795802）

前　言

C 语言是 AT&T 贝尔实验室的 Dennis Ritchie 和 Ken Thompson 于 1972 年开发的一种程序设计语言。C 语言具有与 ALGOL 及 Pascal 类似的结构化语言风格，同时也具有 PL/I 语言的诸多特点，其简洁的语法和高效的执行效率使其成为一种流行的系统程序设计语言。现在，C 语言已经演变为一种通用型的程序设计语言，可运行于多种系统平台以及多种结构的计算机。同时，C 语言也已成为主流的教学语言，本书的介绍以 ANSI C 为基准。

C++语言是由 AT&T 贝尔实验室的 Bjarne Stroustrup 博士设计、实现的，该语言以广泛使用的、适合系统程序设计的 C 语言为基础，同时吸收了 Simula 语言在组织与设计方面的特长。C++最初的版本称为“带类的 C（C with classes）”，1980 年首次投入使用，当时它只支持系统程序设计技术以及数据抽象；1983 年，C++语言增加了对基本的面向对象程序设计的支持；1985 年，C++第一次走向市场；1987 至 1989 年，增加了对类属程序设计支持。C++的标准化工作于 1990 年启动，主要由 ANSI（American National Standard Institute）以及后来加入的 ISO（International Standards Organization）负责，1998 年正式发布了第一个国际标准（ISO/IEC14882）。

C++在 C 语言的基础上，增加了对面向对象编程、类属编程、数据抽象等技术的支持，还对 C 语言进行了非面向对象的扩充。使用 C++语言进行程序设计可以获得可重用性、可靠性、连续性、访问控制、继承性以及多态性等优势。

本书向读者介绍主要支持结构化程序设计的 C 语言，以及在 C 语言基础上进行扩展的支持面向对象程序设计的 C++语言。在介绍 C/C++语言的同时，还介绍了结构化程序设计以及面向对象程序设计的主要内容。第 1～8 章主要介绍 ANSI C 的内容，由于 C++是 C 语言的超集，因此，这部分内容也可以认为是 C++的内容；第 9～15 章是 C++的特有内容，介绍了 C++对 C 的非面向对象的扩充以及面向对象的扩充。需要指出的是第 14 章介绍了 C++近几年新增的内容，包括新的强制类型转换运算符、名字空间、运行时类型信息，以及一般教材都不太介绍的异常处理，不过由于涉及的内容比较多，因此本书只介绍了主要的部分，一些细节性的内容没有展开。考虑到本书主要的读者对象是初学者，因此，本书的例子大多数都是完整的，并且都已经在 Visual C++ 6.0 上测试通过。对例子的分析和研究有助于读者了解实际应用中采用面向对象方法解决问题的基本策略。

本书第 1 章以及第 9～15 章由成颖编写，第 2、3、4 章由戴莉苇编写，第 5、6、7、8 章由曹英编写，各章的习题由张薇薇准备，韩晓静、孙立栋完成了全书的校对工作。全书由成颖修改定稿。

在此，要特别感谢南京大学博士生导师孙建军教授在本书成书过程中给予的帮助和指导，以及就编写策略提出的许多宝贵意见。东南大学出版社张煦老师为本书的编写和出版提供了热情的帮助，在此表示最衷心的谢意。

编　者

2002 年 10 月

目　录

1 C程序设计语言概述

C语言是AT&T贝尔实验室的Dennis Ritchie和Ken Thompson于1972年设计的一种程序设计语言。C语言具有与ALGOL及Pascal类似的结构化语言风格，同时也具有PL/I语言的诸多特点，其简洁的语法和高效的执行效率使其成为一种流行的系统程序设计语言。现在C语言已经演变为一种通用型的程序设计语言，可运行于多种系统平台以及多种结构的计算机。在介绍C语言之前，首先对程序设计语言的发展概况做简单介绍。

1.1 程序设计语言概述

人类之间的交流需要通过自然语言进行，而人与计算机之间的交流则必须通过程序设计语言，用它来描述需要计算机处理的工作。程序设计语言按照出现时间的先后顺序可以分为三类：机器语言、汇编语言和高级语言。其中机器语言与汇编语言都是面向机器的，因此相对于高级语言而言，它们也可以称为低级语言。

1.1.1 机器语言

机器语言是面向计算机的语言，也是最底层的程序设计语言。用机器语言编写的程序，计算机硬件可以直接识别。在用机器语言编写的程序中，每一条机器指令都是二进制形式的指令代码。对于不同的计算机硬件(主要是CPU)，其机器语言是不同的，因此，针对一种计算机所编写的机器语言程序不能在另一种计算机上运行。

由于机器语言程序是直接针对计算机硬件编写的，因此它的执行效率比较高，能充分发挥计算机的性能。

机器语言完全用0和1组成的代码表示，一个简单的算式，也要用很多条命令才能表达出来，因此，用机器语言编写程序的难度比较大，容易出错，编写程序效率低，而且程序的直观性比较差，也不容易移植。

1.1.2 汇编语言

汇编语言也是面向计算机的语言。在汇编语言中人们为了便于理解与记忆，采用能帮助记忆的英文缩写符号（称为指令助记符）来代替机器语言指令代码中的操作码，用地址符号来代替地址码。

汇编语言与机器语言通常是一一对应的，因此，汇编语言也是与具体的计算机对应的。由于汇编语言采用了助记符，因此，它比机器语言直观，容易理解和记忆。但是，计算机不能直接识别用汇编语言编写的程序，必须通过“汇编程序”将汇编语言翻译成机器语言才能进行工作，因此，汇编语言的运行速度比机器语言要慢一些。

机器语言、汇编语言与人类自然语言差别极大，不易交流和推广，都是面向机器的低级语言。

1.1.3 高级语言

为了克服机器语言和汇编语言的缺点，在20世纪50年代中期学者们开发了高级语言，它接近于人类的自然语言——英语，用高级语言编写的程序不再是一条条指令，而是一句句的语句，这些语句的词义、语法都与英语差不多，同时，每一条语句可以对应若干条机器指令，因此，程序量大大减少。用高级语言编制程序不需要考虑计算机内部的具体组织结构，通用性很强，程序几乎可以不加修改或只作微小变化，就能运行在不同的计算机上，这些特点给使用者带来了极大方便。

现在已经出现了成百上千种高级语言，其中最早出现的是1954～1957年由IBM公司设计的FORTRAN（FORmula TRANslator）语言，它应用于需要复杂数学计算的科学和工程领域，目前仍然广为使用。COBOL（Common Business Oriented Language）是一群计算机制造商以及政府和工业界的计算机用户于1959年开发成功的，主要用于需要对大量数据进行精确和有效处理的商业领域，目前，许多商业软件仍然是用COBOL语言编写的。Pascal语言几乎是和C同时设计的，成为80年代计算机科学的主流教学语言，目前其教学语言的地位正逐渐为C以及C++语言所取代。

高级语言的特点是：①独立于具体的计算机，通用性好；②具有适合描述计算过程的语言结构；③易学易用，具有较高的可读性、可维护性和可移植性。

与汇编语言一样，高级语言也不能被计算机直接理解，必须通过编译方式或者解释方式将用高级语言写的源程序变为机器语言的目标程序，计算机才能执行，所以高级语言的运行速度较慢。

1.2 C语言发展历史

20世纪60年代末期，AT&T贝尔电话实验室取消了与MIT以及GE（General Electric）合作开发Multics操作系统的计划，然而，设计一个可用的操作系统仍是KenThompson追求的目标，为此他开始研制UNIX系统(UNIX是Multics的别名)。Multics是用PL/I语言实现的，由于使用PL/I语言编程不太方便，因此Thompson希望使用其他的高级语言来实现该系统。具有BCPL(一种低层次且缺乏运行支持的系统语言)编程经验的Thompson为此专门设计了B语言，B是专用于系统程序设计的BCPL的最小子集(“BCPL仅占PDP—7计算机内存的8K字节”)。

1970年，UNIX项目组得到了一台具有24K存储容量的PDP-11计算机。与此同时，研究组普遍感到B语言使用不便，他们在B的基础上加入了类型和结构定义，增加了其他一些操作，结果诞生了被称为C的新程序设计语言。

尽管C语言是一个通用型的程序设计语言，但它已同系统程序设计紧密地联系在一起。它首先被用来编写UNIX操作系统的核心代码，并使UNIX的实现同大多数C语言版本相联系。70年代，研究和使用C语言的主要场所是研究机构以及高等学校。80年代，随着UNIX操作系统的广泛使用，C语言开始逐渐流行起来。随着C语言的

流行，它的不同实现版本之间出现了或多或少的差异，这对于C语言的进一步推广是不利的。为了解决该问题，1982年，ANSI成立了工作小组X3J11负责C语言的标准化工作，该项工作于1989年完成，标准的代号为ANSI 1989，该标准1990年被国际标准化组织接受并作为国际标准(ISO / IEC 9899：1990)。

与其他语言相比，C 语言主要有以下几方面的特点：

（1）简洁、紧凑，使用方便、灵活，易于学习和应用。C 语言中只有 32 个关键词，9 种控制语句，书写形式自由。

（2）运算符丰富。34 种运算符的灵活使用保证了它可以实现其他高级语言难以实现的运算。

（3）C 语言是结构化程序设计语言，能实现各种复杂的数据结构和运算。

（4）C 语言以函数作为程序模块以实现程序的模块化，符合现代编程风格。

（5）C 语言允许直接对位、字节和地址进行操作，能实现汇编语言的大部分功能。正因为 C 语言同时具有高级语言和低级语言的特点，所以又有人把它称为中级语言。

（6）C 语言的语法限制不严格，自由度较大。

（7）C 语言生成的目标代码质量高。对同一问题，C 语言编写的代码效率仅比汇编语言低 10%～20%。

（8）可移植性较好，基本适用于各种型号的计算机、操作系统，但是与诸如 BASIC 语言等高级语言相比，C 语言的可移植性又要差一些。

1.3 C 源程序的结构特点

为了说明C语言源程序的结构特点，介绍以下几个由简到难的源程序，例子中有些内容尚未介绍，但可从这些例子中了解 C 语言源程序的基本组成以及基本的书写规范。

1.3.1 最简单的 C 程序：打印文本

对初学者而言，学习编程的第一项工作总是使程序在显示器上显示一些信息，下面我们从 K&C 第一个最简单的程序：在屏幕上显示"hello,world!" 开始学习 C 程序设计。

```
/*This is the first example.*/
#include <stdio.h>
int main()
{
    printf("hello, world! \n");
    return 0;
}
```

用文本编辑器将这段程序键入到一个以".C" 结尾的文件，所选择的文件名应有助于记忆。假设用 hello.c 作为已编辑程序的文件名，在编译、连接之后执行该程

序时，在显示器上会显示：

hello, world!

下面详细解释上面的例子：

①/*This is the first example.*/

该行是注释，编译器忽略了"/*"和结束的符号对"*/"之间的文本，也就是说注释在编译时不会产生机器语言的目标代码，因此，它也不会使计算机产生任何动作。注释是为阅读程序的人提供的文档，可以提高程序的可读性。

需要注意的是注释符号"/*...*/"必须成对使用，忘记了注释的结束符"*/"，以及采用"*/"开头或用"/*"结束都是错误的，也不允许嵌套，下面的注释形式是错误的：

/*This is the /*novice.*/ first example.*/

在每一个函数的前面加上描述函数用途的注释是一种良好的程序设计习惯。原则上讲，注释可以位于程序的任何位置，但是，在软件工程实践中，出于可读性的考虑，对于注释的位置也给出了一些原则性的规定。

根据需要注释可以占一行的一部分、一整行或跨越几行，因此这种注释方式也称为页注释符。C++语言中还支持一种"//"的注释方式，它只是将注释符开始到本行结束的部分作为注释，也称为行注释符。

②#include <stdio.h>

用#开始的行称为预处理指令（preprocessing directive），用于与预处理器通信。#include 指令使得预处理器在代码的当前位置上加入标准头文件 stdio.h 的备份，该头文件是 C 系统提供的。<stdio.h>的尖括号指明该文件能在系统缺省的位置找到，具体位置与编译器在安装时的设置有关，引入的原因是它包含了有关 printf()函数的函数原型信息。

③int main()

每个程序都必须有且只有一个名为 main 的函数，称为主函数，该函数是程序执行的入口，main 后面的圆括号告诉编译器这是一个函数，圆括号中为空，表示该函数没有参数，关键字 int 声明该函数的返回值类型是整型值，int 可以缺省，但是除了 main 函数外，通常不缺省。

④{

各函数体以左花括号开始，相对应的右花括号必须出现在函数末尾，通常函数的书写风格是每个花括号独占一行，并靠左放置。花括号也用于将语句组织在一起，该种用法称为复合语句，我们会在后面详细介绍。

⑤printf()

C 系统包含一个能直接用于程序中的标准函数库，其中 printf 函数的作用是将参数表中的内容依据一定的规则将其显示到标准的输出设备（通常为显示器），前面引入头文件 stdio.h 是因为它向编译器提供了函数 printf()的函数原型。

⑥printf("hello, world! \n");

函数调用语句，通过一个参数调用 printf（）函数，该参数为字符串"hello, world!\n"，它控制显示的内容。在字符串尾部的"\n"代表一个称为换行符的字符，这是一个非打印字符，其作用是把光标移到下一行的开始，需要注意的是本行用一个分号结束，C 语言中分号是语句的标志。

函数体中语句的书写规范通常采用缩进格式，即以花括号为基准，向左缩进一个 tab 键（通常为 4 字符的宽度），这样可以使得程序美观、清晰，具有较好的可读性。

⑦return 0;

此函数表示 main()向操作系统返回整型值 0，0 表示程序已经成功地结束，而非 0 值用于告诉操作系统 main()没有成功，与上面的语句一样，本行也以分号结束，表示它是 C 语言的一条语句，同时保持与上一行同样缩进的书写规范。

⑧ }

该右花括号与上面的左花括号相匹配，结束了函数 main()，也独占一行，并且与前面的左花括号在同一列。

函数 printf()产生的效果是在显示器上连续地显示，当输出到一个换行符'\n'时，会移动到新的一行。显示器是从左到右、从上到下二维显示的。为了读起来方便，程序员必须考虑输出在显示器上的布局。

下面用两条 printf()语句重写前面的程序，虽然程序看起来不同，但是输出结果是一样的。

```
/*This is the second example.*/
#include <stdio.h>
int main()
{
    printf("hello,");
    printf("world!\n");
    return 0;
}
```

注意第一个 printf()语句的参数的字符串用一个空格字符结束，如果此处没有该空格字符，则输出结果中也将没有空格。

作为该例子的最后一个变型，再增加一个字符串" Welcome to c world!"，并在两行显示。

```
/*This is the third example.*/
#include <stdio.h>
int main()
{
    printf("Hello, world!\n");
    printf("Welcom to c world!\n");
    return 0;
}
```

编译该程序，并执行后，输出的结果为：

```
Hello, world!
Welcom to c world!
```

在第三个例子的函数体中，也可以用一条语句代替两条 printf（）语句，下面仍然给出完整的源文件。

```
/*This is the fourth example.*/
#include <stdio.h>
int main()
{
    printf("Hello, world!\nWelcom to c world!\n");
    return 0;
}
```

使用类似于 printf()的库函数编写程序是非常重要的，因为 ANSI C 标准函数的编写是严格的，而且所有的 ANSI C 系统都实现了这些函数。用库函数编写的程序具有很好的可移植性，同时，ANSI 标准库函数是高效的，使用这些函数可提高程序的效率。

1.3.2 C 程序另一简单例子

下面的程序首先声明了两个双精度浮点型变量，然后用标准库函数 scanf()读取用户从键盘输入的一个双精度浮点数值，接着计算其正弦值，最后用 printf()函数显示结果。

```
/*另一个简单的 C 程序*/
#include<math.h>
#include<stdio.h>
int main()
{
    double data, result;
    printf("input number:\n");
    scanf("%lf", &data);
    result = sin(data);
    printf("sine of %lf is %lf\n", data, result);
    return 0;
}
```

该程序经编译后，示例的运行结果为：

```
input number:
20.0
sine of 20.000000 is 0.912945
```

下面详细解释该例子：

①/*另一个简单的 C 程序*/

注释行，有关注释的较为详细的内容，已经在第一个例子中介绍过。

②#include<math.h>

在使用 sin()等数学函数时，需要包含数学头文件 math.h。前面已经介绍过：在编译程序之前，凡以#开头的代码行都先由预处理程序进行处理，该代码行通知预处理程序把数学头文件的内容包含到程序中，头文件 math.h 中包含了编译器在编译数学函数时要用到的信息和声明，还包括帮助编译器确定对库函数调用是否正确的信息。

③#include<stdio.h>

printf()函数以及 scanf()函数的使用都需要包含标准头文件 stdio.h。

④int main()

{

这两行的介绍与第一个例子相同。

⑤double data, result;

这是一条声明语句，data、result 是变量名。变量名对应内存中的存储单元，能够存储在程序中使用的值，变量 data、result 被声明为 double 类型，double 是 C 的基本类型之一，还有其他一些基本数据类型。

所有的变量在使用前都必须声明，结尾的分号表示这是一条语句。同一个类型的多个变量可以放在一条声明语句中，变量名之间用逗号分隔，上面的两个变量也可以用两条声明语句分别声明，后面这种做法的好处是可以方便地为变量添加注释。

在每一个逗号后面加一个空格可提高程序的可读性。

⑥printf(″input number:\n″);

在显示器上显示″input number:″ 提示信息，由于″\n″ 的存在，光标将定位在下一行的开始位置，该语句的作用是提示用户采取指定的动作。

⑦scanf(″%lf ″ , &data);

用 scanf()函数读取用户输入的值。scanf()函数的作用是读取来自标准输入设备的输入内容（一般情况下，标准输入设备为键盘）。

上面的 scanf()函数有两个参数：″%lf″ 以及″ &data″ ，第一个参数为格式控制串（format control string），指示用户应该输入的数据类型为双精度浮点型，第二个参数以&开始，其后紧跟变量名，在 C 中&称为″ 取地址运算符″ 。&与变量名 data 连用是把存储 data 的内存地址传递给 scanf()函数，然后计算机就会把 data 的值存储在这个地址单元。初学者或者学过其他语言的读者可能对这种使用方法不太适应，现在只要记住在使用 scanf()函数时，每个变量名前都要加&就可以了，深入的内容在后面介绍。

计算机在执行 scanf()语句时等待用户输入变量 data 的值，用户通过键入一个双精度浮点数并按下回车键响应请求，然后计算机将用户输入的值存储到变量名 data 所对应的内存单元中去。完成操作后，以后对 data 的操作都会引用这个值。

函数 printf()与 scanf()的结合使用提高了用户与计算机之间的交互性，这种交互方式也称为交互式 I/O。在使用 scanf()函数之前，通过 printf()函数给用户以相应的

提示是良好的程序设计风格。

⑧result = sin(data);

计算变量 data 的正弦值，并用赋值运算符"="将计算结果赋给变量 result。运算符"="有两个操作数，也称为二元（目）运算符，对于本例而言，=的操作数是变量 result 以及表达式 sin(data)的运算结果。

在二元运算符的两侧各添加一空格，可以使二元运算符比较突出，从而提高程序的可读性。

⑨printf("sine of %lf is %lf\n", data, result);

用 printf()函数在显示器上输出"sine of "、data 的值、字符串"is"以及 result 的值。printf()函数有三个参数，第一个参数是格式控制串，格式控制串中包含了某些要直接显示的字符以及指示打印双精度的转换说明符"%lf"。第二、三个参数指定了要打印的值，可以发现，scanf()函数以及 printf()函数的双精度的转换说明符是相同的，其他基本数据类型也类似于这种情况。

对于输入和输出函数 scanf 和 printf，这里先简单介绍一下它们的格式，以便下面使用。scanf 和 printf 这两个函数分别称为格式输入函数和格式输出函数。其意义是按指定的格式输入/输出。因此，这两个函数在括号中的参数表都由以下两部分组成：

"格式控制串"，参数 2，参数 3……

"格式控制串"是一个字符串，必须用双引号括起来，它用于控制输入/输出量的数据类型，各种类型的格式表示法将在后面介绍。在 printf 函数中还可以在格式控制串内出现非格式控制字符，这些在显示时按照原文显示。"参数 2，参数 3……"给出了输入或输出的量，当有多个量时，用逗号间隔。例如：

printf("sine of %lf is %lf\n", data, result);

其中%lf 为格式控制字符，表示后面对应的数据按双精度浮点数处理，它在格式串中两次出现，分别对应了 data 和 result 两个变量，其余字符为非格式字符，原样输出在屏幕上，"\n"为不可显示字符，作用为将光标定位到下一行的开始位置。

⑩return 0;

}

程序的后两行与第一个例子相同，"return 0;"将整数 0 返回给操作系统，"}"与前面"{"相匹配，表示程序的结束。

1.3.3 C 源程序的结构特点

从上面介绍的两个例子中，不难看出 C 源程序的结构特点：

①一个 C 语言源程序可以由一个或多个源文件组成。

②每个源文件可由一个或多个函数组成，函数可以是用户自定义函数，也可以是系统标准库函数。

③一个源程序不论由多少个文件组成，都有且只能有一个 main 函数，即主函数。

④源程序中可以有预处理命令(include 命令仅为其中的一种)，预处理命令通常应放在源文件或源程序的最前面。

⑤每一个说明、每一个语句都必须以分号结尾，但预处理命令、函数头和花括号"}"之后不能加分号。

1.3.4 书写程序时应遵循的规则

从书写清晰，便于阅读、理解、维护的角度出发，在书写程序时应遵循以下规则：

①一个说明或一条语句占一行。

②用{}括起来的部分，通常表示程序的某一层次结构。"{"一般与前一行的第一个字母对齐，"}"与"{"对应，并与其在同一列，二者最好单独占一行。

③采用悬行式书写，即低一层次的语句（或说明）可比高一层次的语句（或说明）缩进若干字符后书写（通常为四个字符），以便看起来更加清晰，增加程序的可读性。

④每个二元运算符两侧最好各添加一空格，以突出运算符。每个逗号后面也应留有空格，以增加可读性。

⑤预处理指令放在一起并与后面的语句之间空一行；函数之前的注释应与函数首部之间空一行，变量的声明与可执行语句之间空一行，联系紧密的语句与其他语句之间应当空一行；最后的 return 语句与前面的语句空一行。

上面的约定不是 ANSI C 的内容，主要的目的是为了增加程序的可读性以提高程序的可维护性，它不会增加最终的可执行代码的空间，不过，如果要打印源程序则可能要多消耗一些纸张。在编程时应力求遵循这些原则，以养成良好的编程风格。(由于篇幅的限制，规则⑤的空行原则本书中将不体现，但读者应养成这样的习惯。)

1.4 C 开发环境的基本知识

C 程序从开发到执行通常需要经过六个阶段，分别为：编辑、预处理、编译、连接、加载和执行。下面以常用的 C/C++开发环境 Visual C++6.0 为例介绍这 6 个步骤，因为 Visual C++6.0 是集成的编译环境，下面以图示方式说明 Visual C++6.0 的简单使用，VC++6.0 更加详细的使用请读者参阅相关的手册或说明书。

第一步是编辑，用编辑器完成。程序员用编辑器键入 C 程序，并在需要的时候用编辑器加以修改，然后把该程序存储在二级存储设备（如磁盘等）中。C 程序文件要使用扩展名".c"。

第二步是预编译、编译，在第一步形成的源文件基础上进行。在 Visual C++6.0 中没有将预编译和编译严格区分开来，而是通过菜单以及快捷键将它们集成在一起了，使用起来比较方便。

第三步是连接，C 程序通常会引用定义在其他地方的函数，如标准库中的函数以及其他公用库中的函数，因此，C 编译器所产生的目标代码通常会缺少这部分内容。连接程序把目标代码和这些函数的代码连接起来产生可执行文件。