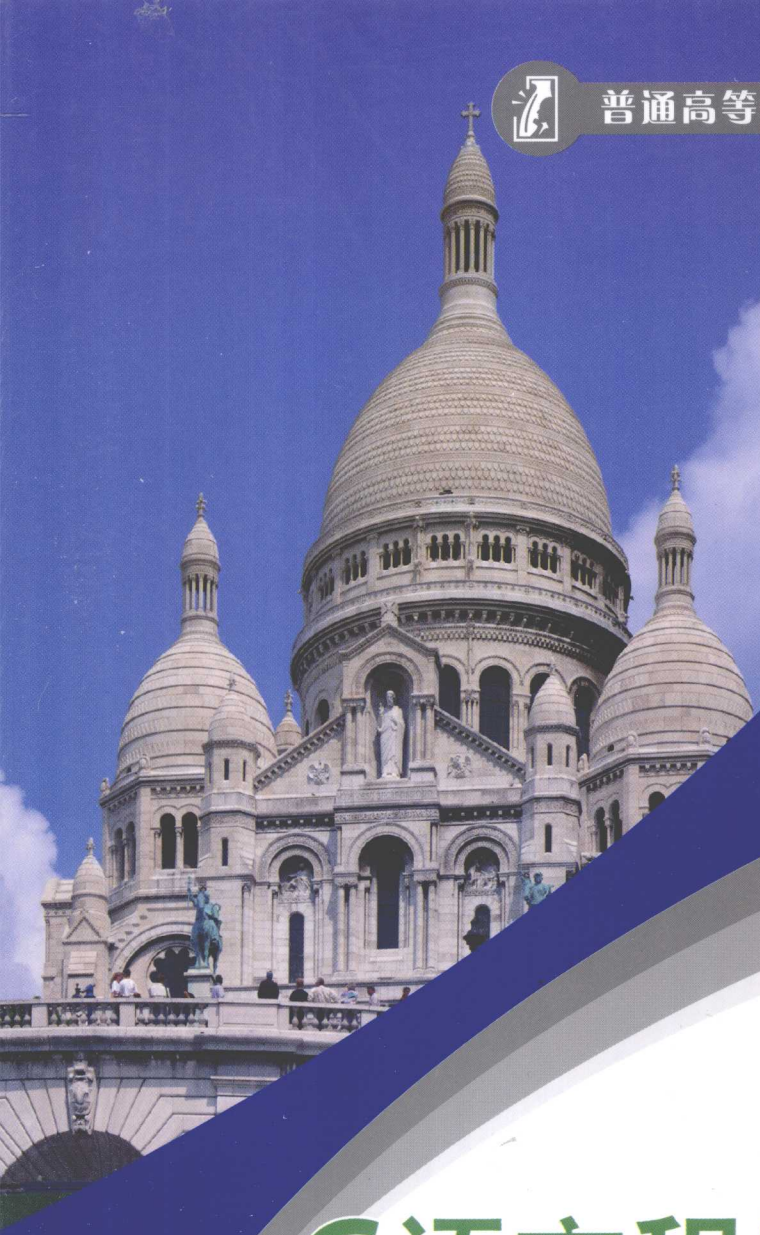
普通高等教育信息技术类系列规划教材



# C语言程序设计

唐新来 李春贵 主编

科学出版社
www.sciencep.com

普通高等教育信息技术类系列规划教材

# C语言程序设计

唐新来　李春贵　主　编

王晓荣　向　荣　何春华　唐　杰　副主编

科学出版社

北　京

## 内 容 简 介

本书主要讲解如何用C语言进行编程。书中涵盖了C语言的基本特性，包括主要的ANSI C99标准新增加的特性。本书通过示例讲解C语言编程，使用完整的程序来阐释每个概念和编程思想，而且对C函数都提供了具体说明。

本书共分9章，分别介绍了计算机与C编程入门、C语言程序设计初步、C程序控制结构、数组、函数、指针、复杂数据类型、文件、预编译和位运算等知识。此外，本书每章结尾都配有习题，帮助学生巩固所学知识。

本书不仅可以作为普通本科院校独立学院（三本）的教材，也可以作为高职高专院校、成人高校、广播电视大学等各类高等院校的教材，同时还可作为相关等级考试的教材，以及程序设计爱好者的自学用书。

**图书在版编目(CIP)数据**

C语言程序设计 / 唐新来，李春贵主编. —北京：科学出版社，2009
（普通高等教育信息技术类系列规划教材）
ISBN 978-7-03-025390-3

Ⅰ.C… Ⅱ.①唐…②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150225号

责任编辑：李太铼 文 戈 / 责任校对：耿 耘
责任印制：吕春珉 / 封面设计：一克米工作室

科学出版社 出版
北京东黄城根北街16号
邮政编码：100717
http://www.sciencep.com

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2009年9月第 一 版 开本：787×1092 1/16
2009年9月第一次印刷 印张：16 1/2
印数：1—3 000 字数：376 000

**定价：26.00元**

（如有印装质量问题，我社负责调换〈环伟〉）
销售部电话 010-62134988 编辑部电话 010-62135763-8220

# 前　言

C 语言是目前国际上广泛流行的一种结构化的程序设计语言，它具有高级语言和汇编语言（低级语言）的功能，提供类型丰富、使用灵活的基本运算和数据类型，具有较高的可移植性。C 语言不仅适合于开发系统软件，而且也是开发应用软件和进行大规模科学计算的常用程序设计语言。

本书的两个目标，即讲授程序设计以及讲授 C 语言。C 语言之所以被很多初学者认为非常困难，可以追溯到它的历史。C 语言是作为在 UNIX 操作系统下编程的工具而设计的，它最初的使用者就是理解操作系统和底层机器复杂性的程序员，这些使用者认为在其程序中利用这些知识很自然。所以，C 语言的基本概念复杂、内容丰富、使用灵活，一些初学者经常会发现，学习 C 语言的过程是一个充满挫折的艰难过程。一方面觉得学习 C 语言内容枯燥，难度大；另一方面，即便学完了 C 语言程序设计课程，而一旦要用 C 语言来独立编程解决一些实际问题时会感到无从下手，不能很好地将理论和实际结合起来。

计算机教育面向应用，学生学习的主要目的是“应用”程序设计语言，是学会如何用程序解决应用领域的问题，这不需要细致地研究程序设计语言本身十分严格的语法和语义。基于这样的观点，并针对以上问题，编者通过认真分析和研究，并结合多年从事 C 语言课程教学的丰富实践经验，制定了本书编写的基本特色，既能够讲授程序开发的正确方法又可以讲授 ANSI C 语言，而且本书选择前者作为最主要的目标。

（1）强调程序设计思想和相应的学习方法，构建相应的知识体系。使读者形成良好的编程思维方式，具备程序设计的能力和实际解决问题的能力，以及相应的知识扩展能力。

（2）以现代 C 语言标准 ANSI C 为主导，以成熟的 Visual C++6.0 为编译环境，全面介绍了 C 语言的基本理论、基本知识以及编程的基本技能和方法。

（3）逻辑条理清楚，语言叙述通俗易懂，内容由浅入深，循序渐进，难易程度过渡自然，有利于初学者学习。

（4）采用了大量与实际问题紧密结合的实例贯穿整个学习过程，使理论和实践紧密结合，突出应用，有利于激发学生的学习兴趣，提高应用能力。

（5）对一些涉及算法的典型例题都采取先分析后给出程序代码的顺序，以期开拓学生的思维，提高学生分析问题和解决问题的能力。针对典型例题还提供了举一反三的练习题，以培养学生迁移知识的能力。

（6）每章后面提供了对本章知识点进行总结的复习思考题和难度呈梯次分布的习题，有助于学生抓住本章重点和难点，深入掌握所学知识。

本书由广西工学院的李春贵、何春华、王萌、王晓荣，广西工学院鹿山学院的唐新来，桂林电子科技大学信息科技学院的向荣，桂林理工大学博文管理学院的唐杰，广西大学行健文理学院的甘秋玲、彭颖等联合编写；王萌完成全书策划，王萌、李春贵、何

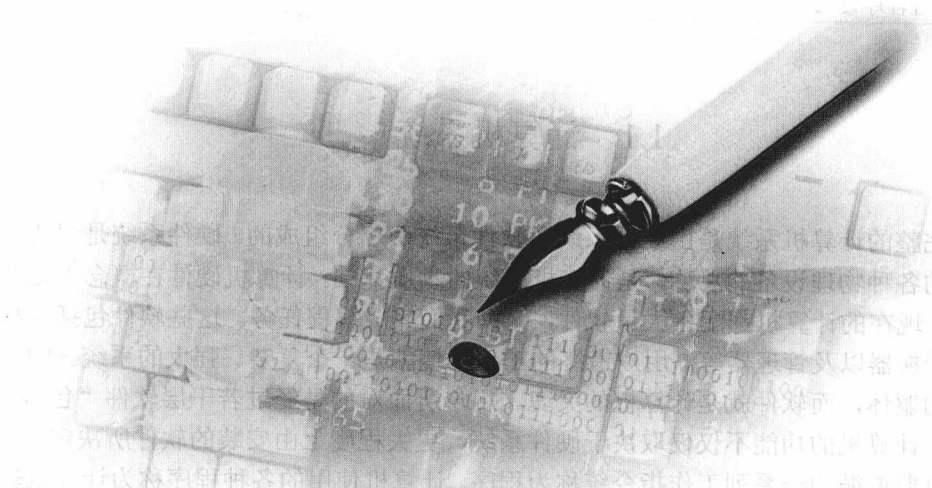
春华、王晓荣讨论完成编写提纲，王晓荣最后统稿，并由唐新来、李春贵任主编，王晓荣、向荣、何春华、唐杰任副主编。其中，第 1 章由李春贵编写，第 2、3 章由何春华、唐杰编写，第 4 章由王萌编写，第 5、6 章由王晓荣编写，第 7 章由甘秋玲、彭颖编写，第 8 章由向荣编写，第 9 章由唐新来编写。

本书在编写过程中得到了广西工学院和广西工学院鹿山学院的大力支持，并得到有关专家、教师的指导和帮助，在此一并表示衷心的感谢。

由于作者水平有限，书中难免有不妥之处，欢迎读者多提宝贵意见。

编　者

# 目　录



# 第 1 章　计算机和 C 编程入门

## 教学目标

- 了解计算机系统的基本组件
- 了解设计和编写程序的基本技术
- 熟悉 C 程序的工作原理
- 熟悉 C 语言的基本成分

## 教学重点

- 计算机的基本组件，以及设计和编写程序的基本技术
- C 语言的成分，以及“先见森林后见树木”的效果

# 1.1 计算机系统

一个完整的计算机系统是由硬件系统和软件系统两部分组成的。硬件系统是组成计算机系统的各种物理设备的总称，是计算机系统的物质基础。计算机硬件在概念上是非常简单的。现在的计算机都配备大量软件，帮助我们完成编程任务。这些软件包括各种编译器、转换器以及管理器等。用户最终的工作环境就是一个复杂、强大的系统。硬件是计算机的躯体，而软件就是计算机的灵魂。用户所面对的是经过若干层软件“包装”的计算机，计算机的功能不仅仅取决于硬件系统，更大程度上由安装的软件所决定。

计算机要遵循的一系列工作指令统称为程序。计算机使用的各种程序称为计算机软件。本书内容围绕软件展开，介绍 C 语言的相关知识，但初学者还是应该对硬件知识作一些简单的了解。

## 1.1.1 硬件

大多数计算机系统的硬件由 5 个主要组件构成：输入设备、输出设备、中央处理器（也称为中央处理单元或者 CPU）、主存储器以及辅助存储器，如图 1.1 所示。计算机的中央处理器、主存储器以及辅助存储器通常安装到机箱内部。中央处理器和主存储器是一台计算机的核心，可将其视为一个集成单元。其他组件连接主存储器，并遵照处理器的指示工作。图 1.1 中的箭头指明了信息流动的方向。

输入设备是允许用户将信息发送给计算机的设备。主要的输入设备是键盘和鼠标。

输出设备是允许计算机将信息输出给用户的设备。最常用的输出设备是显示屏幕，通常称之为显示器。但计算机系统通常不止一个输出设备。例如，除了显示器之外，计算机还可能连接了一台打印机，可在纸上产生输出。有时可将键盘和显示器视为一个单元，并统称为终端。

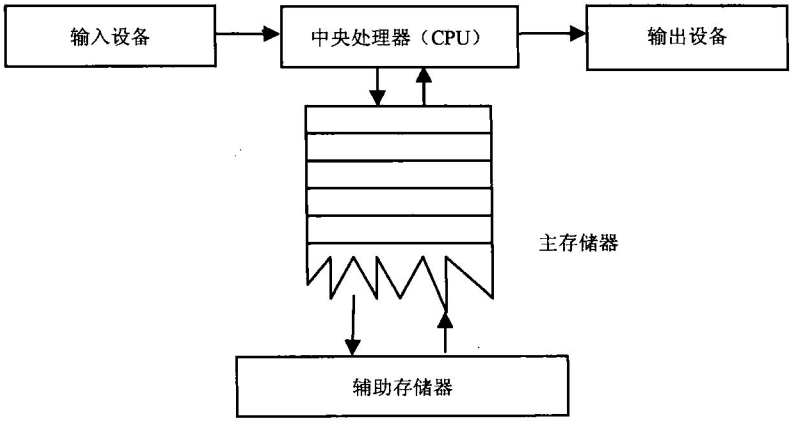



图 1.1 计算机的主要组件

为了存储输入，计算机提供了存储器。计算机执行的程序也存储在存储器中。计算机支持两种形式的存储器，称为主存储器（内存）和辅助存储器（外存）。准备执行的程序存储在主存储器中。正如“主存储器”这个名称所暗示的，它是最重要的存储器。主存储器由一个很长的编号位置列表构成，这些位置称为存储位置或者内存位置。在不同计算机上，内存位置的数量也是不同的，范围从几千个到几百万个，有时甚至能达到数十亿个。每个内存位置包含一系列 0 和 1。这些位置的内容可以改变。所以，每个内存位置可被视为一块小黑板，计算机可在上面涂写和擦除。在大多数计算机中，每个内存包含的 0、1 个数都是相同的。假如一个数字只能选择 0 或 1 这两个值，就称为一个二进制数字，或者称为 1 位、1 比特（bit）。大多数计算机的内存位置都包含 8 位（或者 8 位的倍数）。每 8 位都称为一个字节（byte），所以我们可将这些编号的内存位置称为字节。对一个字节进行标识的编号称为该字节的地址。一个数据项（如一个数字或者一个字母）可存储在其中的一个字节中。以后需要数据项时，就根据字节的地址来查找数据项。

如果计算机需要处理的数据项（如一个很大的数字）太大，以至于单独一个字节装不下，就使用几个相邻的字节来容纳数据项。在这种情况下，用于容纳该数据项的整个内存块仍然称为一个内存位置。构成这个内存位置的第一个字节作为这个较大的内存位置的地址使用。图 1.2 展示了一台计算机的主存储器。内存位置的长度不是固定的，在计算机上运行一个新程序时，它们就有可能改变。

在计算机存储器中，信息实际表示成 0 和 1 组成的字符串，计算机必须将这些 0、1 序列解释成字母、数字、指令或者其他类型的信息。计算机根据特定的编码方案来自动执行这些解释。不过，程序员编程时不需要关心这些编码。使用 C（或其他大多数编程语言）编程时，不必过于关心这个事实。

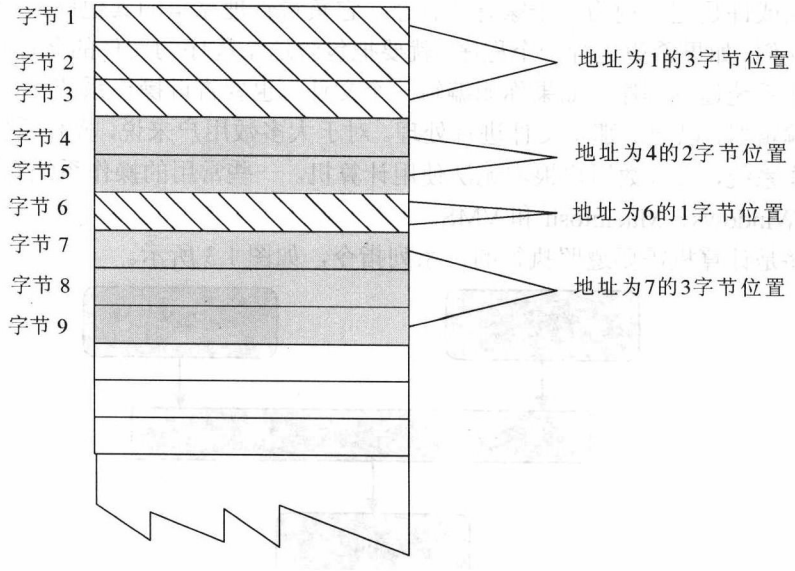



图 1.2　内存位置和字节

到目前为止，我们讨论的都是主存储器。没有主存储器，计算机不能做任何事情。但是，只有计算机真正按照一个程序中的指令工作时，才会用到主存储器。计算机还配备了另一种形式的存储器，称为辅助存储器或者辅助存储（此处，“存储器”和“存储”的含义是一样的）。辅助存储器用于持久性保存数据。程序运行之后（和之前），数据都会予以保留。通常也可以将辅助存储器称为辅助存储设备、外部存储或者外存。

在辅助存储设备中，信息以文件为单位来保存。取决于用户的需要，文件可大可小。例如，程序平时存储在辅助存储设备的一个文件中，并在程序运行时复制到主存储器（内存）。任何形式的信息都可存储到一个文件中，包括程序、信函以及存货单等。

最常见的辅助存储设备包括硬盘、软盘和 CD。软盘和 CD 的优点在于便宜和可移动，但硬盘能储存更多的数据，而且速度是所有辅助储存设备中最快的。虽然还有其他形式的辅助存储设备（比如磁带机），但上述 3 种设备是最常用的。有时，几种不同的辅助存储设备可同时连接到一台计算机。

中央处理器是计算机的“大脑”。计算机广告中，厂商可能指出它包含是什么芯片。这里的芯片就是处理器。处理器遵循一个程序的指令进行操作，并执行程序要求的计算。然而，处理器只是一个非常简单的大脑。它唯一能做的就是按照程序员提供的一系列简单的指令进行操作。不同的计算机可能采用不同的处理器指令。现代计算机的处理器通常都支持几百个指令。然而，如同上述描述的那样，每个指令所执行的任务都是非常简单的。

### 1.1.2 软件

用户通常不是直接与计算机沟通，而是通过一个操作系统和它交互。操作系统将计算机的资源分配给计算机。操作系统实际是一个程序，或者多个相互协作的程序，但更好的方法或许是把它视为一个家庭的管家。它负责管理家里的其他佣人，把主人的要求传递给他们。如果希望运行一个程序，就要把包含这个程序的文件的名称告诉操作系统，再由操作系统运行程序。如果你想编辑一个文件，也要告诉操作系统文件名是什么，它会启动编辑器，以便对那个文件进行处理。对于大多数用户来说，操作系统就是计算机。没有操作系统，大多数用户根本无法使用计算机。一些常用的操作系统有 UNIX、DOS、Linux、Windows、Macintosh 和 VMS。

程序是计算机需要遵照执行的一系列指令，如图 1.3 所示。

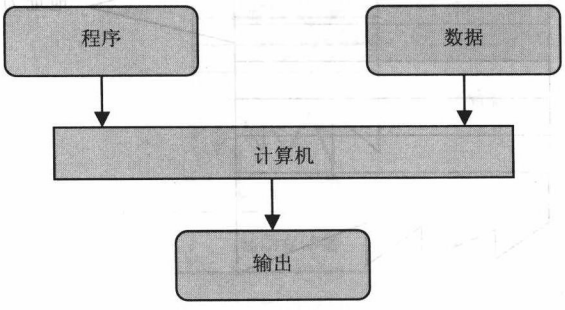



图 1.3 运行一个程序的简单示意图

计算机获得的输入可划分为两大部分：程序和数据。计算机根据程序中的指令进行操作，在这个过程中，会执行一些具体的处理。从概念上说，数据就是我们向一个程序提供的输入。例如，假定程序要对两个数字进行相加，这两个数字就是数据。换言之，数据是程序获得的输入。但程序和数据共同构成了一台计算机获得的输入（通常通过操作系统来输入）。任何时候只要向计算机提供了一个它必须遵照执行的程序，并为程序提供了一些数据，那么计算机就必须根据数据来执行程序。“数据”一词还具有更广的含义，从广义上来说，数据是指适用于计算机的任何信息。

### 1.1.3　高级语言

许多语言都可以用来编写程序，本书将讨论 C 编程语言，用它来写程序。C 是一种高级语言，Java、Pascal、Visual Basic、FORTRAN、COBOL、LISP、Scheme 和 Ada 等也是高级语言。高级语言在许多方面都类似于人类使用的自然语言。它们的设计宗旨是方便人们写程序，以及方便阅读和理解。高级语言包含的指令比计算机的中央处理器能够执行的简单指令要复杂得多。

计算机能理解的语言称为低级语言。在不同类型的计算机上，低级语言的细节也是不同的。

一个典型的低级语言指令可能是：ADD　X　Y　Z

该指令的功能是将内存位置 X 的数据加上内存位置 Y 的数据上，再将结果放到内存位置 Z 处。上述简单的指令是用汇编语言写成的。虽然汇编语言已非常接近计算机能直接理解的语言，但仍然要经历一个简单的转换，才能由计算机真正理解。计算机要遵循一个汇编语言指令，所有单词都必须转换成 0，1 序列。例如，若 ADD 可能转换成 0110，X 可能转换成 1001，Y 可能转换成 1010，而 Z 转换成 1011，则执行上述指令时，计算机最终实际执行的是 0110 1001 1010 1011。在不同机器使用的汇编语言指令以及它们转换成 0、1 序列的方式是不同的。

对于以 0 和 1 的形式写成的程序，认为它是用机器语言来编写的，那才是计算机能够真正理解并遵循的语言。汇编语言和机器语言差别不大，它们之间的差别对我们来说并不重要。最重要的是机器语言和高级语言区别：用高级语言编写的所有程序都必须转换成机器语言版本，以便计算机理解并遵照执行。

### 1.1.4　编译器

编译器是一种特殊的程序，它能将高级语言转换成机器语言程序，使计算机能直接理解和执行。编译器获取的输入或数据是一个程序，输出的是另一个程序，如图 1.4 所示。为避免混淆，我们将输入程序称为源程序或者源码，由编译器转换成的版本则称为目标程序或者目标码。

转换和运行 C 程序的完整过程要稍微复杂一些。任何 C 程序都会用到某些已经写

好的操作（如输入和输出）。这些操作已经编写成程序，而且均已通过编译，并有了它们的目标码。它们等待着与用户的目标码合并，生成一个完整的机器语言程序，统一在计算机上运行。所以在这个过程中，要用另一个名为链接器的程序将事先准备好的目标码与基于用户的 C 程序而生成的目标码合并到一起。图 1.5 展示了编译器和链接器的交互。但在目前的大多数系统中，这个链接过程都是自动完成的。所以，一般用不着关心链接问题。

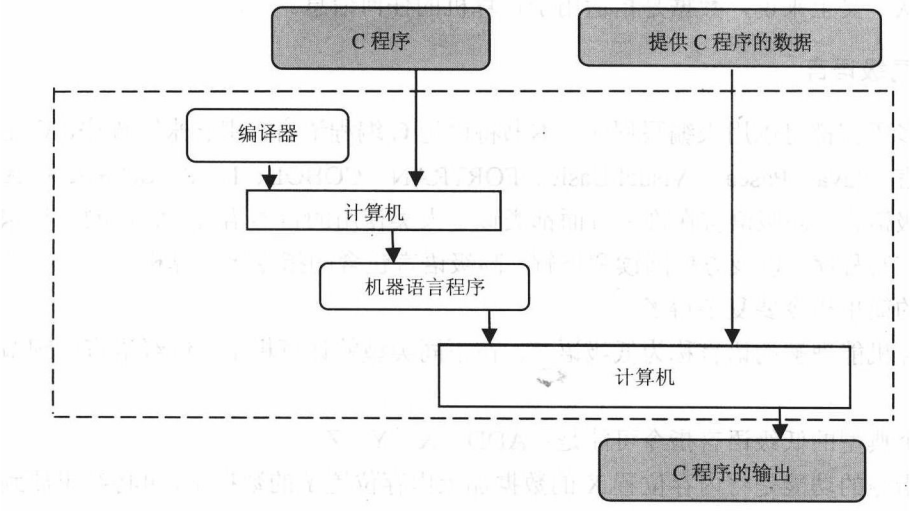



图 1.4 编译和运行 C 程序（基本过程）

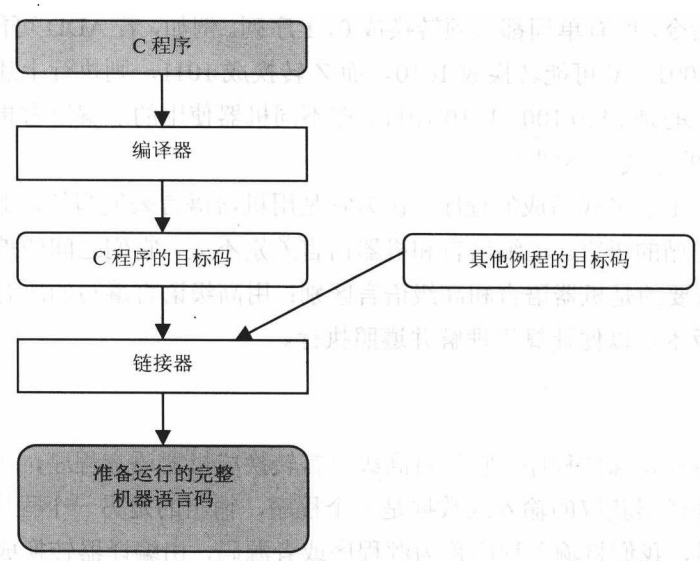



图 1.5 运行 C 程序之前的准备工作