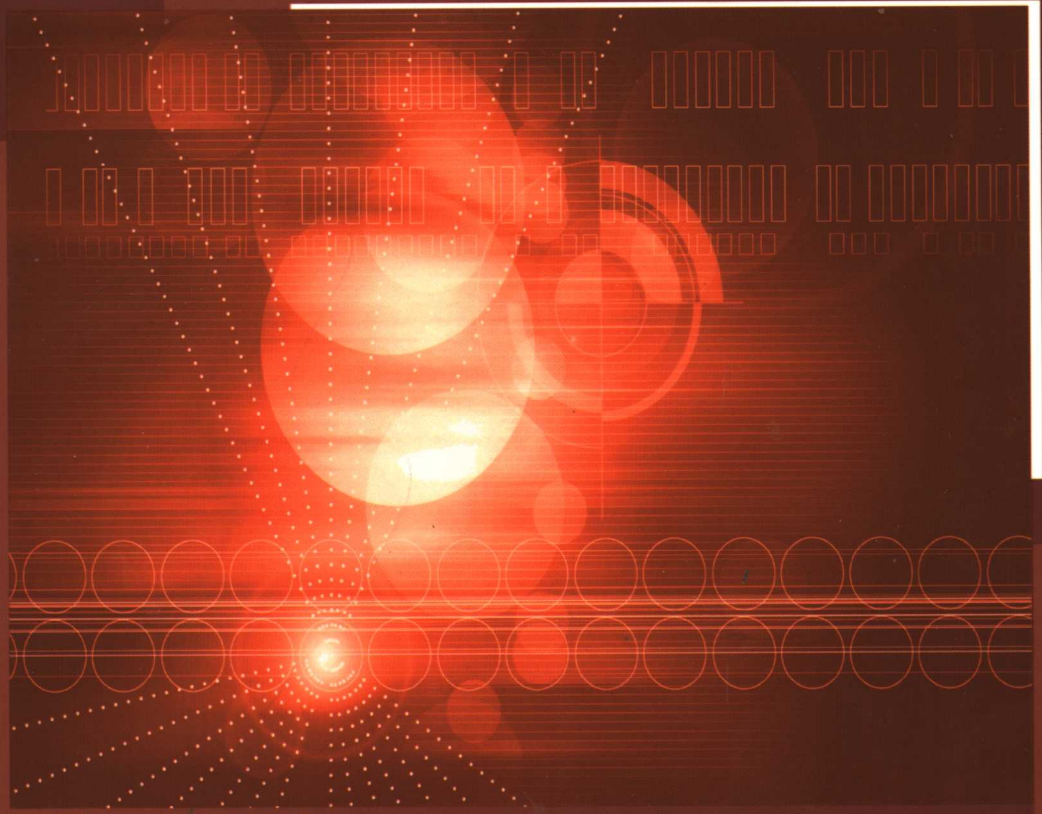
HZ BOOKS

计算机基础课程系列教材

C语言程序设计教程



汪同庆 张华 杨先娣 主编

本书为教师配有电子教案

机械工业出版社
China Machine Press

计算机基础课程系列教材

C语言程序设计教程

汪同庆 张华 杨先娣 主编
滕冲 黄磊 关焕梅 汤洁 黄文斌 参编

机械工业出版社
China Machine Press

C语言作为一种适用于开发系统软件及应用软件的计算机语言，已经成为计算机程序设计语言的主流语种。本书从培养应用型人才的角度出发，系统地介绍了C语言编程的基本知识和程序设计的基本方法，内容包括：C语言概述、基本数据类型、运算符和表达式、控制结构、函数、作用域和存储类别、数组、指针、字符串、结构体和共用体、编译预处理、文件，最后介绍了高级数据结构的基础知识。各章配有案例和习题，教师可免费下载电子教案。

本书可作为各类高等院校非计算机专业计算机公共基础课程的教学用书，也可供计算机等级考试和自学参考。

图书在版编目（CIP）数据

C语言程序设计教程/汪同庆，张华，杨先娣主编. －北京：机械工业出版社，2007.3
（计算机基础课程系列教材）
ISBN 978-7-111-20760-3

Ⅰ.C…　Ⅱ.①汪…　②张…　③杨…　Ⅲ.C语言－程序设计－教材　Ⅳ.TP312

中国版本图书馆CIP数据核字（2007）第007982号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：李东震
北京京北制版厂印刷·新华书店北京发行所发行
2007年3月第1版第1次印刷
184mm×260mm·15印张
定价：24.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

前　言

C 语言是一种功能强大、编程方式灵活、特色鲜明、深受广大科技人员和专业编程者喜爱的程序设计语言。全国计算机等级考试、全国计算机应用技术证书考试、全国计算机软件专业技术资格及水平考试等都将 C 语言纳入其考试科目；许多高校也将 C 语言列为程序设计课程的首选语种。C 语言已经成为最重要、最流行的编程语言之一。

本书在编写过程中力求取材得当、通俗易懂、结构清晰、层次分明，通过精选典型案例验证和说明语句成分、语法结构和程序设计方法。注重对程序设计语言基本概念、语句规则、程序结构和编程方法的讲解，摒弃了一些复杂的应用，以期让读者能尽快和轻松地迈进程序设计的大门。

本书共分 14 章，主要内容包括：

第 1 章：C 语言概述。介绍了 C 语言的发展、特性，C 语言程序开发过程和 C 语言程序的开发环境。

第 2 章：C 语言快速入门。通过几个简单例程帮助读者快速浏览 C 语言程序的组成元素和基本特征，并介绍如何检测程序中出现的错误，以及基本的调试方法。

第 3 章：基本数据类型。介绍了 C 语言常量和变量的基本概念、基本类型数据的使用、格式输入输出函数的使用。

第 4 章：运算符和表达式。介绍了算术运算符、关系运算符、逻辑运算符、赋值运算符、逗号运算符、位运算符及其表达式。

第 5 章：结构化程序设计。介绍了算法的概念以及表示，分支结构中的 if 语句、switch 语句，循环结构中的 while 语句，do-while 语句和 for 语句。

第 6 章：函数。介绍了 C 语言程序的模块化结构、库函数的使用、函数的概念和定义、函数原型、函数间的数据传递、递归调用函数。

第 7 章：作用域和存储类别。介绍了变量的作用域、局部变量和全局变量、变量的存储类别以及内部函数和外部函数。

第 8 章：数组。介绍了一维数组和二维数组的概念、定义、初始化以及数组元素的引用和数组作为函数参数。

第 9 章：指针。介绍了指针的概念，指针变量的应用，指针与数组，指针与函数。

第 10 章：字符串。介绍了字符串的定义，用字符数组存储和处理字符串，使指针变量指向字符串和字符串处理函数。

第 11 章：结构体、共用体和枚举。介绍了结构体、共用体和枚举三种数据类型的概念、定义格式与使用方法。

第 12 章：编译预处理。介绍了编译预处理的概念和常用的编译预处理命令，如宏定义、文件包含和条件编译等。

第 13 章：文件。介绍了文件的概念、文件指针、文件处理的基本过程和用于处理文件的函数。

第 14 章：高级数据结构。以链表、栈和队列为例，介绍了数据结构的基本概念和常用算法。

本书基于 Windows XP 的 Visual C/C++ 6.0 编程环境，考虑到全国计算机等级考试 C 语言仍使用 Turbo C 2.0 环境，所以同时介绍了基于 DOS 的 Turbo C 2.0 的使用。本书中全部例程均在

Visual C/C++ 6.0 编程环境中编译、链接，在 Windows XP 上运行。

本书由汪同庆、张华、杨先娣主编和统稿。第 1、2 章和附录由张华编写，第 3、4 章由汪同庆编写，第 5 章由黄磊编写，第 6、7 章由汤洁编写，第 8、10 章由滕冲编写，第 9、12 章由杨先娣编写，第 11、13 章由关焕梅编写，第 14 章由黄文斌编写。在本书编写过程中，得到了有关领导和机械工业出版社华章分社的大力支持，在此表示衷心感谢。

因编者水平所限，书中难免存在疏漏之处，恳请广大读者提出宝贵意见。

为了帮助读者学习，与本书配套的《C 语言程序设计实验教程》也将出版。为教师免费提供与本书配套的电子教案和实例源程序。请登录华章网站（www.hzbook.com）下载。

编　者

2006 年 11 月

于武汉大学珞珈山

目　录

第1章 C语言概述

C语言是一种通用的编程语言，深受广大科技人员和专业编程者的喜爱。本章将为学习和使用这一强大而流行的语言做准备，带领读者开始充满挑战和欣喜的学习之旅。

首先，让我们看一看C语言的起源和特性以及与其他几种流行语言的关系。接着，将介绍编程的起源和一些基本原则。最后，介绍开发C语言程序的编程环境。

1.1 C语言简史

1.1.1 C语言的起源

C语言是美国贝尔实验室的Dennis Ritchie在1972年开发的，当时的特定目的是用于设计UNIX操作系统。之所以称为C语言，是因为它的前身是B语言。而B语言是由Ken Thompson于1970年为第一个UNIX系统开发的语言。

C语言是作为从事实际编程工作的程序员的一种工具而出现的，所以其主要目标是成为一种实用的语言。由于C语言很适合用来编写编译器和操作系统，因而被称为“系统编程语言”。尽管如此，它同样很适合用来编写不同应用领域的程序。

1.1.2 C语言的标准

由于C语言功能强大而灵活，因此很快被传播到贝尔实验室之外，世界各地的程序员都使用它来编写各种程序。然而，不久后，不同的组织开始使用自己的C语言版本，不同实现之间微妙的差别令程序员头痛。为解决这个问题，美国国家标准化组织（ANSI）于1983年成立了一个委员会（X3J11），以确定C语言的标准。该标准（ANSI C）于1989年正式采用。国际标准化组织（ISO）于1990年采用了一个C标准（ISO C）。ISO C和ANSI C实质上是同一个标准，通常被称为C89或C90。现代的C语言编译器绝大多数都遵守该标准。

最新的标准是C99标准。制定该标准的意图不是为语言添加新特性，而是为了满足新的目标（例如支持国际化编程）。所以该标准依然保持了C语言的本质特性：简短、清楚和高效。目前，大多数C语言编译器没有完全实现C99的所有修改。本书将遵循C89标准，并不涉及C99的修改。

1.2 为什么学习和使用C语言

1.2.1 C语言的特性

从诞生至今的30多年中，C语言已经成为最重要和最流行的编程语言之一。它之所以得到发展，是因为它自身具有许多优势，得到很多计算机编程人员的喜爱。归纳起来，C语言具有以下主要特性：

C语言是一种强大而灵活的语言，可以用来编写复杂的程序。C语言既有高级语言的特性，又具有低级语言的性能，因此可用于编写系统软件（如操作系统和编译器）、办公软件（如字处理软件和电子表格程序）、专业软件（如图形图像处理软件和音、视频处理软件）、游戏软件等。

C语言简洁、紧凑，使用方便、灵活。C语言一共有32个关键字，9种控制语句，程序书写形式自由。

C语言程序效率高、运行速度快。这得益于它丰富的数据类型和范围广泛的运算符。

C语言是可移植的。这意味着为一种计算机系统（如一般的PC机）编写的C语言程序，可以在不同的系统（如HP的小型机）中运行，而只需作少量的修改，甚至无须修改。这种可移植性也体现在不同的操作系统之间，例如Windows和Linux。

C语言很适合结构化程序设计，要求用户以功能模块的方式来思考问题。这些模块的恰当结合可以形成一个完整的程序。这种模块化的结构使得程序的调试、测试和维护更加容易。

C语言具有自我扩展能力。一个C语言程序基本上是各种函数的集合，这些函数由C语言的函数库支持，并可以再次被用在其他程序中。用户可以不断地将自己开发的函数添加到C语言函数库中去。由于有了大量的函数，程序设计也就变得简单了。

1.2.2 C语言与C++、Java和C#

如果仅仅因为前面的特性，就说C语言是编程初学者的首选，未免掩盖了其他语言的优势。在当前的计算机编程领域，还有其他“炙手可热”的高级语言可供选择，如C++、Java和C#。那么C语言和它们又有什么区别和联系呢？

C++语言是贝尔实验室于20世纪80年代在C语言的基础上开发的，它是C语言的超集，包含了C语言的所有内容，同时增加了面向对象编程方面的内容。面向对象的基础是面向过程，是为了解决编写大型软件的问题而产生的。C++是一门非常复杂的语言，在学习C++的时候，几乎有关C语言的所有知识都用得上。因此可以说，学习C语言不但是在学习当今最强大、最流行的编程语言，同时还在为学习C++语言做准备。

Java是Sun公司于1995年发布的面向对象编程语言。和C++一样，Java也是基于C语言的。如果打算以后学习Java，那么几乎C的所有知识都是适用的。

C#是一门新生的语言，由微软在2000年6月与.NET平台一同推出。同C++和Java一样，C#也是从C语言派生的一种面向对象语言。同样，很多有关C的知识也适用于C#编程。

1.3 预备知识

1.3.1 计算机工作的基本原理

在学习C语言编程之前，应该了解计算机工作原理方面的一些知识。这些知识有助于理解用C编写程序与运行程序结果之间的联系。

计算机是能够执行计算和做出逻辑决策的机器。它是在一系列指令的控制下处理数据的，这些指令就构成了计算机程序。也就是说，人们通过编制程序来确切地告诉计算机要做的事以及如何去做，而计算机执行程序来完成任务。

现代计算机可分为几个部件：CPU（中央处理单元）担负着绝大部分计算工作；内存作为一个具有记忆功能的部件用来保存程序和文件；辅助存储器（一般指硬盘）保证在计算机关机时也能记下程序和文件；还有各种外围设备（如键盘、鼠标和显示器）用来提供人与计算机之间的通信。CPU负责处理程序，所以下面将集中讨论它的功能。

CPU从内存中读取一条指令并执行该指令，然后从内存中读取下一条指令并执行。一个主频为1GHz的CPU可以在一秒钟内进行大约一亿次这样的操作。CPU内部包括若干个称为寄存器的存储部件，每个寄存器可以保存一个数。一个寄存器（指令计数器）保存下一条指令的内存地址，CPU使用该信息获取一条指令，并保存在另一个寄存器（指令寄存器）中，然后将指令计数器的值更新为下一条指令的地址。CPU只能理解有限的指令，这些指令就构成了该CPU的指令集。每一条指令所能完成的任务是非常具体的，例如把一个数从内存复制到某一个寄存器。

从前面的说明中，可以得到下面两个概念：

第一，存储在计算机中的一切内容都是数。数（例如整数、小数）是以数字形式存储的，

字符是以数字形式的编码存储的，图形图像也是以数字形式的编码存储的。当然，CPU 执行的指令也是以数字形式存储的，而且一个指令集中的每一条指令都被分配一个数字代码。指令的数字代码被称为指令码。

第二，既然程序是一系列指令的集合，那么程序最终在执行的时候是以数字形式的指令码来表示的。

1.3.2 计算机语言及其处理程序

1. 机器语言

数字形式的指令码称为机器语言，用机器语言编写的程序可以直接交付计算机执行。不过，用机器语言来编写程序是一项非常烦琐的工作。因为即使是一件非常简单的事，例如两个数相加，也必须分成若干个步骤：

1）将地址为 2000 的内存单元中的数复制到寄存器 1；

2）将地址为 2004 的内存单元中的数复制到寄存器 2；

3）将寄存器 2 中的数与寄存器 1 中的数相加，结果保留在寄存器 1 中；

4）将寄存器 1 中的数复制到地址为 2008 的内存单元中。

这里必须以指令的数字形式来书写程序。一旦程序中间有错误（常常发生），在一堆数字中查找出错点犹如大海捞针。

2. 汇编语言和汇编程序

后来发展出了符号化的汇编语言。程序员不再写数字形式的指令代码，而是写指令的符号代码，并且可以为每个数据的存储位置定义一个名字。下面是用汇编语言来表示两个数相加所要执行的动作：

1）ldreg n1，r1　　把变量 n1 的值复制到寄存器 1（r1）；

2）ldreg n2，r2　　把变量 n2 的值复制到寄存器 2（r2）；

3）add r1，r2　　把 r2 中的数与 r1 中的数相加，结果保留在 r1 中；

4）store r1，sum　　把 r1 中的数复制到变量 sum。

其中，每一个变量对应一个内存单元，变量的值就是该内存单元中保存的数。

尽管汇编语言程序读起来清楚一些，但计算机却无法理解它，需要将它翻译成机器语言。汇编程序就是用来完成这项任务的语言处理程序。在把汇编语言程序翻译成机器语言程序时，汇编程序将为变量分配内存单元。

3. 高级语言和编译程序

使用汇编语言编写程序时仍然需要很多指令才能够实现最简单的任务。为了加速编程的过程，人们开发了高级语言，在高级语言中，单个语句就能够实现基本任务。下面是用 C 语言来表示两个数相加的一条语句：

```
sum = n1 + n2;
```

它包含常用的数学符号和数学表达式。很明显，用高级语言编写的程序更容易被人们理解和接受，也将从多方面提高编程的效率。首先，不必去考虑 CPU 的指令集。其次，不必考虑 CPU 实现特定任务所需采取的精确步骤，而是采用更接近人类思考问题的方式去书写语句，表达意图。

与汇编语言程序类似，高级语言程序需要被编译程序（或编译器）翻译成机器语言，才能被计算机所理解。采用编译器还有另一个好处。一般来说，每种计算机在设计上都有其自身特有的机器语言。为英特尔的奔腾 CPU 编写的机器语言或汇编语言程序对苹果的麦金塔 CPU 来说是不能理解的。但可以选择正确的编译器将同一个高级语言程序转换为各种不同的机器语言程序。也就是说，程序员解决一个编程问题只需一次，然后可以让编译器将该解决方案解释为各种机器

语言，即可以在各种机器上运行同一个高级语言程序。

1.4 C语言程序的开发过程

进行C语言程序开发，首先必须定义问题。如果不知道问题是什么，将无法找到解决方案。知道问题是什么之后，便可以设计解决它的方案。有了方案后，要实现它。方案实现后，必须对结果进行测试，以确定问题是否得到解决。

一般，把创建C语言程序（或其他语言的计算机程序）的过程分成五个步骤（或阶段）。

1.4.1 定义程序目标

在开始解决问题之前，程序的开发人员必须彻底了解问题是什么，对即将要创建的程序要做的事情有个清晰的想法。这需要仔细考虑程序需要什么信息、程序需要执行哪些计算和操作以及程序应该报告什么信息。

假设需要计算一个圆环的面积。开始需要有人说明，“我希望程序可以计算圆环的面积”。这样就可以定下总体目标。但是，要弄清楚问题还必须回答一些问题，例如“什么是圆环”，“计算圆环的面积的公式是什么”，“需要的计算精度是多少”等。

1.4.2 设计程序

明确要解决的问题是什么之后，就要决定程序如何去完成它。这需要考虑如何给程序提供必要的数据，程序怎样报告结果，程序如何组织，完成这个程序需要多长时间。还需要确定在程序中如何表示数据以及用什么方法来处理数据。选择一个好的方式来表示信息通常可以使程序设计和数据处理容易很多。

此时，应该用一般的概念考虑设计中的问题，而不是考虑具体的代码。不过，有些决策可能会取决于具体语言的一般特性。例如，C语言的数据表示方式比Pascal语言多，那么在面对同样的问题时，C编程人员有较多的选择。

1.4.3 编写代码

在这一阶段就可以开始编写代码来实现前面的设计，即用某一门语言来表示程序设计。这一阶段需要开发人员熟练掌握该语言的知识。开发人员可以在草稿纸上勾画自己的想法或书写代码，但最终必须将代码输入计算机。输入代码所采用的机制则取决于具体的编程环境。一般来说，需要使用文本编辑器（例如Windows的记事本程序）来创建一种包含程序设计的C语言表示形式（称为源程序或源代码）的文件，即源文件。

1.4.4 编译

接下来，就需要把C语言源程序转换成机器语言表示的程序，这种程序可以直接被计算机理解和执行，称之为可执行程序，放在可执行文件中。编译的具体细节取决于使用的编程环境。

C语言分两步完成这一工作：编译和链接。编译器将源代码转换为目标代码，并存在目标文件中，链接器将目标代码与其他代码结合起来生成可执行文件。这种把编译和链接分开来做的方法便于程序的模块化。可以分别编译程序的各个模块，然后用链接器把编译过的模块结合起来。这样，如果需要改变一个模块，则不需要重新编译所有其他模块。

1.4.5 运行、测试和调试程序

运行包含可执行程序的文件，观察运行的结果。在不同的系统中运行程序的方式可能不同。例如，在Windows和Linux的控制台中，要运行某个程序，只需输入相应的可执行文件的名字即可。而在Windows的资源管理器中，可以通过双击可执行文件的名字和图标来运行程序。

程序可以运行是好的，但有可能运行不正确。因此，应该对程序进行仔细的检查，看程序是

否在做该做的事。比较好的做法是为验证程序的正确性设计一个测试计划，而且这项工作越早做越好，因为它有助于理清程序员的思路。

程序中可能存在各种各样的错误，调试就是发现并修正错误的过程。

1.5 C 语言的编程环境

1.5.1 C 语言的编程环境

前面讲的五个步骤是开发 C 语言程序一般应遵循的过程，但具体的操作却取决于所使用的计算机环境，特别是后面三个阶段。因为 C 语言是可移植的，所以它在许多环境中都是可用的，例如 UNIX、Linux、Windows 和 MS-DOS。在介绍具体的环境之前先来介绍 C 语言编程环境所共有的一些特征。

C 语言编程环境包括一系列程序，这些程序允许程序员输入代码创建程序、编译程序、链接程序、执行和调试程序。图 1-1 说明了在编程环境中创建程序的过程。

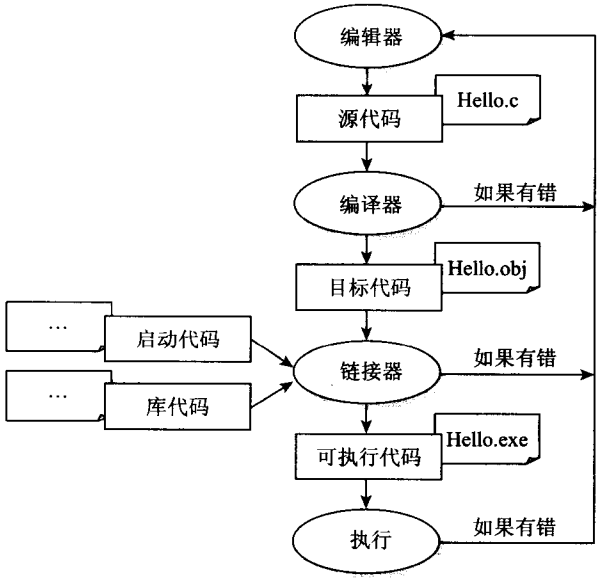


图 1-1 C 语言的编程环境

1. 编辑器

用 C 语言编写程序时，需要使用一个文本编辑程序输入源代码，并将代码保存在源文件中。一般，C 程序的源文件名称的扩展名是.c，例如 welcometoyou.c 和 Hello.c。该名称应该遵循特定的操作系统的命名规则。例如，MS-DOS 要求基本名包含的字符数不能大于 8，所以 welcometoyou.c 不是合法的 DOS 文件名。Windows 允许长文件名，所以 welcometoyou.c 是合法的 Windows 文件名。

2. 编译器

正如前面所介绍的，不同的计算机有不同的机器语言，C 编译器用来把 C 语言转换成特定的机器语言。编译器接收源文件，生成目标文件，扩展名为.obj 或.o。

编译器还会检查输入的程序是否为有效的 C 语言程序。如果编译器发现错误，就会报告出错，且不能生成可执行程序。这时就必须修改错误，然后再编译。显然，为了能迅速找到错误，理解特定编译器的报错信息是必要的。

3. 链接器

虽然编译后生成的目标文件包含的已经是机器语言代码，但是该文件还不是一个完整的程序，不能运行。目标文件中所缺少的第一个元素是一种叫做启动代码的东西，该代码相当于程序和操作系统之间的接口。所缺少的第二种元素是库函数的代码。几乎所有的C语言程序都利用标准C库中所包含的函数。目标文件中不包含这些函数的代码，实际的代码存储在另一个称为“库”的文件中。库文件中包含许多函数的目标代码。

链接器的作用就是将这三个元素（目标代码、启动代码和库代码）结合起来，并将它们放在一个文件中，即可执行文件，扩展名为.exe或.out。对库代码来说，链接器只从库中提取使用到的函数的代码。

在一些系统上，必须分别运行编译器和链接器，而在另一些系统上，编译器可以自动启动链接器。

1.5.2 UNIX环境

C的早期目标就是为了编写UNIX，所以UNIX本身提供C的编译器，称为cc。可以从众多的UNIX文本编辑器中选择一种来创建C语言程序的源文件，例如用vi编辑器创建hello.c，然后用下面的命令编译该程序：

```
cc hello.c
```

如果程序没有错误，将得到一个可执行的文件hello.out。要运行该文件，只需输入：

```
hello.out
```

cc在编译生成目标文件hello.o后，立即启动链接器生成可执行文件，并删除目标文件。

1.5.3 Windows环境

因为Windows并不包含C编译器，所以需要获得并安装一个C编译器。而许多现代的编译器都是集成开发环境（或叫IDE）的一部分，其中包括一个编辑器、编译器、链接器和一个包括符号调试程序在内的运行支持系统。这些系统程序在菜单驱动的命令解释器中使用，让开发人员不用离开集成开发环境就能编写、编译和运行程序。

许多软件厂商都提供了基于Windows的集成开发环境，例如微软的Visual C/C++、Borland公司的C/C++ Builder、Metrowerks的Code Warrior等。目前，大多数IDE把C和C++编译器结合在一起。本书所有例程都是在Windows XP上使用微软的Visual C/C++ 6.0开发的，并且将直接用Visual C++或VC来指代Visual C/C++ 6.0或其更新的版本。

1.5.4 DOS环境

虽然DOS的应用已经很少见了，但是某些情况下可能还需要在DOS上编写、运行C语言程序。DOS也不带C编译器，开发人员需要获得并安装一个C编译器。那么，Borland公司的Turbo C 2.0/3.0是不错的选择，它们是基于DOS的IDE。其实，许多Windows上的IDE也提供了在DOS命令行环境中编程的命令行工具。

本章小结

C语言是一种通用的程序设计语言，它具有很多突出的特性，利用它能够写出功能强大、代码简洁、结构优雅、可移植的程序，因而深受广大编程者的喜爱。

程序开发应该按照一定的步骤进行，尤其是程序设计的初学者更应该严格要求自己，以便尽快掌握程序设计的基本原则。

C语言是一种需要编译的语言。C编译器和链接器是将C语言源代码转换为可执行代码的程

序。进行 C 语言编程的具体操作依赖于所使用的系统环境，Windows 系统中的 IDE 为编程人员高效地开发 C 程序提供了较好的支持。

习题

1. C 语言的可移植性指的是什么？
2. 试比较机器语言、汇编语言和高级语言。
3. 简述利用 C 语言编程的 5 个步骤。
4. 解释源代码文件、目标代码文件和可执行文件之间的区别。
5. C 语言的编程环境包括哪些系统程序？并简述各自的功能。

第2章　C语言快速入门

读者一定很想看看完整的C语言程序的样子，甚至想亲自动手写一个程序，看看运行的结果。本章将通过几个程序实例快速浏览C语言程序的组成元素和基本特征，并讲述如何检测程序中的错误以及基本的调试方法。

2.1　一个简单的程序实例

这里先从一个最简单的C语言程序开始，通过解析它的结构去了解C语言程序的组成和基本特征。

【例2-1】用C语言编写一个程序，在屏幕上显示一行文字：This is my first C program。

```
#include <stdio.h>

void main()
{
  printf("This is my first C program.");
}
```

在控制台窗口中执行该程序，将在屏幕上看到下面的结果：

```
This is my first C program.
```

2.1.1　程序结构

图2-1是对这个简单C语言程序每一行的说明。

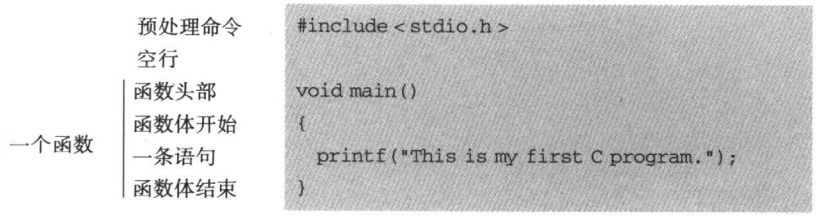


图2-1　简单的C语言程序结构分析

2.1.2　字符集

C语言程序代码是由来自一个字符集的字符组成的。C语言的字符集包括字母、数字、空白符和一些特殊字符。

1）字母：26个英文大写字母（A，B，…，Z）和26个英文小写字母（a，b，…，z）。

2）数字：10个十进制的数字字符：0，1，2，…，9。

3）空白符：包括空格、制表符、回车符和换行符。

4）特殊字符：特殊字符包括+、-、*、/、%、=、(、)、<、>、[、]、{、}、!、&、|、,、?、~、#、_、'、"、;、:、\。

在编写C语言程序时，只能使用C语言字符集中的字符，且区分大小写字母。如果使用其他字符，编译器将把它们视为非法字符而报错。