国外计算机科学教材系列

国家教育部
双语教学示范课程
使用教材

双语版
C程序设计

Learn C through English and Chinese

[爱尔兰] Paul Kelly
苏小红 著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

国外计算机科学教材系列

双语版C程序设计

Learn C through English and Chinese

［爱尔兰］ Paul Kelly 著
苏小红

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书由在计算机程序设计方面有着丰富教学和实践经验的中外作者合作编写。共分14章内容，由浅入深全面介绍C程序设计方法，包括基本数据类型和基本输入/输出方式、各种控制结构和语句、指针和数组、字符串、函数、结构、文件输入和输出等内容，最后讨论了C预处理器。本书所有实例经过精心挑选、贴近生活，尤其强调读者的亲自参与意识。每章都为初学者提供了常见错误分析，所选习题可提高读者上机编程的兴趣。本书采用中英文对照混排，既方便初学者熟悉相关概念和内容，也便于英文非母语的读者熟悉英文专业词汇。

本书可作为高等学校计算机相关专业或软件学院的C程序设计双语教材，也可供程序员和编程爱好者参考使用。

图书在版编目（CIP）数据

双语版C程序设计 = Learn C through English and Chinese: 汉英对照 /（爱尔兰）凯利（Kelly, P.），苏小红著. 北京：电子工业出版社，2013.3
国外计算机科学教材系列
ISBN 978-7-121-19043-8

Ⅰ.①双… Ⅱ.①凯… ②苏… Ⅲ.①C语言－程序设计－高等学校－教材－汉、英 Ⅳ.①TP312

中国版本图书馆CIP数据核字（2012）第281623号

策划编辑：马 岚
责任编辑：冯小贝
印 刷：北京中新伟业印刷有限公司
装 订：北京中新伟业印刷有限公司
出版发行：电子工业出版社
北京市海淀区万寿路173信箱 邮编：100036
开 本：787 × 980 1/16 印张：15 字数：499千字
印 次：2013年3月第1次印刷
定 价：35.00元

凡所购买电子工业出版社的图书有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系及邮购电话：（010）88254888。
质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。
服务热线：（010）88258888。

Preface

Learn C through English and Chinese assumes no previous programming knowledge and has been carefully written to teach the fundamentals of programming through the medium of the C programming language. Although the book is primarily intended for use in a first year undergraduate Computer Science course, it will be equally useful to experienced programmers who intend to learn C programming on their own.

The book presents a thorough introduction to programming, using working programs to demonstrate the key features of the C language in a step-by-step logical sequence.

Using their professional teaching experience, the authors have carefully chosen each example program to carefully explain an important programming concept, confidently leading the learner to competence in C and programming in general. Emphasis is placed on key programming concepts and features of programming in C, while keeping explanations as simple and clear as possible.

Key concepts are re-enforced throughout the book by the use of 'Quick Syntax Reference' and 'Programming Pitfalls' sections at the end of each chapter. These are useful reference points for learners while writing their own programs.

Although mainly written in English, this book includes additional explanatory annotations in Chinese. This bi-lingual approach will be appreciated by Chinese learners and will help them focus on the learning task in hand, without being over-burdened by English technical terminology.

Despite C being available on a wide variety of platforms, this book is not specific to any particular machine, compiler, or operating system. All programs are designed to be portable with little or no modification to a wide variety of platforms.

Learn C through English and Chinese

- Is a comprehensive introduction to C programming.
- Uses practical examples to explain theoretical concepts.
- Uses an active learning approach with detailed explanations of working programming examples.
- Uses additional explanatory annotations written in Chinese.
- Each chapter contains
  - A 'Quick syntax reference' section.
  - A 'Programming pitfalls' section.
  - End-of-chapter exercises, allowing the learner to test and re-enforce their understanding of the programming concepts covered in the chapter.
- Is accompanied by a web site containing the example programs used in the book in addition to solutions to selected end-of-chapter exercises.

Typographic conventions used in this book.

The line numbers to the left of the program examples are for reference purposes only and are not part of the C language.

Keyboard input to a program is printed in this typeface.

When a C keyword is first introduce it is in bold and in this typeface.

```
Program examples, screen output, reference to part of a C statement, a variable, a
value in a program or a C keyword are in this typeface.
```

When a new term is introduced it is in *italic* type.

前　言

为了适应国内高等教育逐步与国际接轨的发展趋势，英语教学和双语教学越来越受到人们的重视。尤其是以“国际化、工业化”为办学理念并注重国际化、工业化人才培养的国家示范性软件学院的部分课程，还常常邀请一些外籍教师来国内进行全英语授课。但是由于办学经费有限，国内学生的英语水平又参差不齐，导致全英语教学目前还无法普及。因此，利用国内的优秀师资（包括留学回国人员）进行双语教学成为首选。针对目前许多双语课程教学缺少双语版教材的问题，我们组织国内和国外大学教师合作编写了本书。

本书的第一作者是爱尔兰都柏林工业大学（DIT）的高级讲师 Paul Kelly。他长期从事程序设计类课程的教学工作，在程序设计类课程教学方面具有丰富的教学实践经验，在国外已先后出版多本程序设计语言类书籍，还曾数次被哈尔滨工业大学软件学院作为外聘教师应邀来校讲授程序设计方面的课程。Kelly 对中国学生比较了解，针对其在教学中发现的问题，即初学者面临着既不熟悉专业术语和基本概念、又不熟悉英文专用词汇的双重困难，提出了出版中英文对照混排式双语版教材的思路，帮助学生在克服语言障碍的同时，能够更快、更好地熟悉和掌握计算机程序设计方面的基础知识，为国内的双语教学提供了一种最佳的解决方案。

本书内容共分 14 章，由浅入深、全面介绍了 C 语言程序设计方法。本书的特点如下：

1. 使用实际生活中的例子和直观的图示，通俗易懂地讲解难于理解的概念。
2. 采用案例驱动和循序渐进的方式，从一个应用实例出发，先利用现有知识编写出一个较为简单的程序，然后在此基础上不断扩充，在扩充的过程中引入一个新的概念和知识点，逐渐编写出一个较大的程序。
3. 每个例程都有详细的讲解，重点内容和段落给出了中文注解，适合以 C 作为入门语言的读者对照阅读，既方便初学者熟悉相关概念和内容，也便于母语不是英语的读者熟悉英文的专业词汇，尤其适合双语教学。
4. 每章后面都有一节介绍初学者编程时易犯的错误，以帮助初学者在程序设计中避免这些错误。
5. 每章后面都有快速语法参考，总结本章内容，便于读者快速查询相关内容。
6. 每章后面都有精心设计的、有趣的习题，便于读者测试和强化对相关内容的理解。
7. 有相关的教学网站（华信教育资源网，网址 http://www.hxedu.com.cn）和教材网站（http://book.sunner.cn），方便读者下载示例的源代码和教学课件等资料。

本书是继国内首次出版的中英文对照混排式双语版教材《双语版C++程序设计》之后出版的第二本双语版程序设计教材。Paul Kelly是一位治学非常严谨的教师，本书的第二作者苏小红在与他合著过程中，经常为一个细节内容的编写进行交流与讨论，书稿完成后又进行了多次校对工作。本着对所有读者负责的精神，我们真诚地欢迎读者对教材提出宝贵意见，可以通过发送电子邮件或在网站（http://book.sunner.cn）上留言等多种方式与我们交流讨论。此外，作者的电子邮件地址为Paul.Kelly@dit.ie及sxh@hit.edu.cn。

苏小红
哈尔滨工业大学计算机科学与技术学院
2013年3月

目　录

Chapter One
Introduction to C
第1章 引 言

1.1 Brief history of C（C语言简史）

The C programming language was originally developed at Bell Laboratories, New Jersey, USA in 1972. It was developed by Dennis Ritchie as a tool for writing the UNIX operating system. It evolved from a language called B, which in turn evolved from a language called BCPL (Basic Combined Programming Language).

C语言于1972年诞生在美国新泽西的贝尔实验室。最初，C语言是由Dennis Ritchie作为编写UNIX操作系统的工具而开发的，它是在B语言的基础上发展起来的，B语言是由一种称为BCPL（基本组合编程语言）的编程语言演变而来。

The American National Standards Institute (ANSI) developed the first standardised specification for the language in 1989 and is commonly referred to as the C89 standard. C89 was further revised in 1999 resulting in a new C99 standard.

1989年，美国国家标准化协会（ANSI）开发了第一个C语言标准，通常称为C89。1999年，C89经过修订，成为新的C99标准。

C is now widely used with many different operating systems running on many different hardware platforms. C, and other languages derived from it (such as Java, PHP and Perl), are widely used in Linux, Solaris, Macintosh, and Windows programming.

现在，C语言已经广泛运行于不同硬件平台的不同的操作系统中。C语言及由C语言派生的其他语言（如Java、PHP和Perl），已广泛应用于基于Linux、Solaris、Macintosh和Windows操作系统的程序设计中。

1.2 Why programmers use C（为什么程序员爱用C语言）

Here are just some of the reasons why programmers use C in preference to any other programming language:

相比其他编程语言，程序员更喜欢用C语言的几个原因如下。

1.2.1 C is portable

C is a portable language, which means that programs written in C on one computer system can, with little or no modifications, be transferred to and run on a completely different computer system. This should be a basic necessity for any programming language, but C comes nearest to meeting this important requirement.

C语言是可移植的编程语言，即在一种计算机系统中用C语言编写的程序在少量修改甚至无须修改的情况下就可以转换并运行于完全不同的其他计算机系统中。这是对所有编程语言的基本要求，但是，C语言是其中最能满足这个重要需求的一种语言。

C's portability is due to the adherence of C compilers to a specified standard. The specifications for C were originally laid down by Brian Kernighan and Dennis Ritchie in their book *The C Programming Language* (Prentice Hall, 1978). These specifications were further developed by ANSI, and most modern C compilers follow these specifications.

C语言的可移植性源于C编译器遵循了特定的标准。这些规范最初是由Brian Kernighan和Dennis Ritchie在其合著的*C Programming Language*中提出的。这些规范后来被美国国家标准化协会进一步完善，现在大部分现代的C编译器都遵循这个规范。

1.2.2 C is a structured programming language

C contains all the constructs required by modern structured programming techniques. The use of structured techniques results in programs that are reliable, understandable, and easy to maintain.

C语言包含现代结构化程序设计技术所需的所有结构，使用结构化程序设计技术可以编写可靠性高、可读性好、易于维护的程序。

1.2.3 C is efficient

Programs developed in C are smaller and faster than corresponding programs written in most other programming languages.

相比用其他大多数语言编写的程序，用C语言编写的程序更小、更快。

1.2.4 C is flexible

C can be used to write applications in a variety of areas, such as:

C语言可用来编写各种应用领域中的程序，例如：

- artificial intelligence applications, such as expert systems and robotics
- commercial computer systems, such as accounts and distribution
- telecommunications applications
- computer-aided design (CAD)
- computer games
- database systems
- process control and real-time applications
- system programs, such as text editors, compilers, interpreters, and linkers.
  In fact C can be used to write a full operating system. For example, the major part of the Linux and Macintosh operating systems are written in C.
- mobile hand-held devices

- 人工智能应用，例如专家系统和机器人
- 商业计算机系统，例如财务报表和销售
- 通信应用
- 计算机辅助设计（CAD）
- 计算机游戏
- 数据库系统
- 过程控制和实时应用
- 系统程序，例如文本编辑器、编译器、解释器和链接器
  事实上，C语言可以用来编写整个操作系统。例如，Linux和Macintosh操作系统的主要部分都是用C语言编写的。
- 移动手持设备

1.2.5 C is powerful

The C language contains some unique statements that allow for the efficient manipulation of system resources, such as memory, which would not be possible in other languages.

C语言包含一些独特的语句用于有效管理系统资源（例如内存），而其他编程语言是不具备这些功能的。

1.2.6 C is concise

Conciseness allows the programmer to give instructions to the computer in as few words as are necessary. Conciseness leads to smaller source programs, which are easier to edit and manage.

C语言的简洁性允许程序员只用所必须的只言片语就能向计算机发出指令，这使得用C编写的源程序更加短小，更易于编辑和管理。

1.3 Developing a C program（开发 C 程序）

Your first step in writing any computer program is to define and understand the problem to be solved. If you cannot understand the problem then you certainly will not be able to tell a computer how to solve the problem.

The second step is to devise an *algorithm* to solve the problem. An algorithm is a series of well-defined steps that, if taken one by one, will solve the problem.

The third step is to design a program to implement the chosen algorithm.

These three steps are an extremely important stage of the development process and are similar in many ways to the work done by an architect before a house is built. Neglect at this stage will cause you to end up with something that does not suit your needs.

Next comes the building stage, i.e. the actual writing of the program.

Firstly, the C program statements are typed and stored in a file using a *text editor*. The file containing the C statements is called the *source file* and is usually stored on disk.

Before you can run a C program on a computer you must pass the program through a *compiler*, which translates the C instructions to machine instructions that the computer 'understands'. The compiler reads the source file, translates the C statements into machine or *object code*, and stores the object code in an *object file*.

编写计算机程序的第一步是定义和理解所要解决的问题。如果不理解要解决的问题，那么你当然不能告诉计算机如何去解决这个问题。

第二步是设计用于解决这个问题的算法，**算法**是一系列精心定义的解决问题的步骤，依次执行这些步骤就可以解决问题。

第三步是设计程序，实现选定的算法。

上述三个步骤是程序开发过程中最为重要的阶段，与在建造房子前建筑师所做的工作有些类似。如果忽视了它们，那么将使程序最终在某些方面达不到用户的需求。

接下来进入构建阶段，即实际编写程序。

首先，使用**文本编辑器**将C语句键入并存储于某个文件中。这个包含C语句的文件称为**源文件**，通常存储在磁盘中。

在计算机上运行C程序之前，必须通过**编译器**将C指令翻译成计算机能够“理解”的机器指令，编译器首先读取源文件，然后将C语句翻译成机器代码或是**目标代码**，并将目标代码存储在**目标文件**中。

运行程序前的最后一步是使用**链接器**对C程序进行链接，在链接过程中，将C程序

The final stage before running the program is to *link* the program using the *linker* program. Linking involves combining the object file of your program with other object files from the C run-time library to form an *executable file*.

The final step is to run the program and observe the results.

In summary, the steps are:

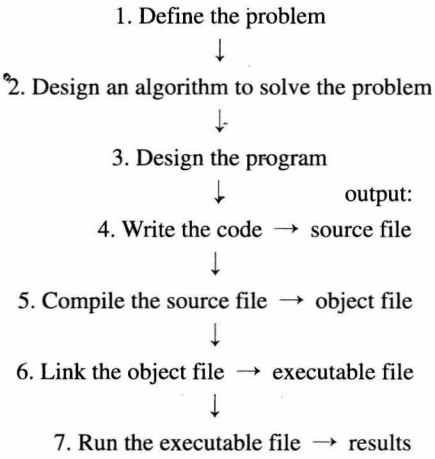

Figure 1.1 steps in developing a program

You are unlikely, except with the simplest of programs, to go from step 1 to step 7 without encountering some problem that will force you back to a previous step. For example, when you write the C statements you are likely to make some coding errors that the compiler will inform you of in step 5. These types of errors are called *syntax* errors. An example of a syntax error is the omission of a comma or semicolon in a C statement. To correct syntax errors go back to step 4, make the corrections, and return to step 5.

Similarly, when you run the program you may find that it is not working as expected. Perhaps the C statements you wrote in step 4 do not implement the algorithm of step 2 correctly. This will involve you going back to step 4, but hopefully not any further. Going back to step 2 or 3 at this stage would be like asking an architect to redesign parts of the house while it is being built!

的目标文件和来自于C运行库的其他目标文件进行组合，从而形成**可执行文件**。

最后一步是运行程序，观察程序的输出结果。

除非是编写最简单的程序，否则，在上述的步骤1~7中，常常会遇到一些问题，使你不得不返回前一个步骤。例如，在编辑C语句时，很可能出现代码输入错误，编译器在步骤5提示错误，这种类型的错误称为**语法**错误，例如，在C语言中少写了一个逗号或分号就是一个语法错误。为了修正语法错误就要回到步骤4，修正错误后，再继续执行步骤5。

同样，在运行程序时，可能会发现程序并未按预期工作，这可能是因为在步骤4中编写的代码并不能正确地实现在步骤2中设计的算法。这将导致不得不回到步骤4修改代码，希望不是步骤2或步骤3的错误，因为如果是更前面步骤的问题，那么就类似于在建造房子时要求建筑师重新设计房子的部分结构一样。

Before being released for use, the program should be thoroughly tested to ensure it solves the problem defined in step 1.

在程序正式发布使用之前，必须对程序进行全面测试，以保证它能解决步骤1中定义的问题。

1.4 Suggestions for learning C programming（学习C语言程序设计的建议）

This book encourages an active learning approach to gaining competence in programming. As you are reading, run the example programs and relate the program statements to the output from the programs. To save typing, the source code for all the example programs can be downloaded from the book's web site

http://book.sunner.cn

or from

http://dl.dropbox.com/u/4035182/cprogs.zip

本书鼓励读者采用积极的学习方法来提高自己的程序设计能力，一边阅读、一边上机运行程序并分析语句和程序输出结果之间的关系。

Also, keep in mind that you'll learn more from designing, writing, running, and correcting programs than you ever will by simply reading the programs in the book. Practice is important.

To help you practise, there are exercises at the end of each chapter and there are links on the web site to free compilers and other resources. Do as many of the exercises as possible and get some feedback on your solutions from people who know C. However, do not rush into writing programs (step 4 in section 1.3) without giving steps 1 to 3 in section 1.3 careful consideration.

另外还要牢记，相对于单纯地阅读书上的程序而言，在设计、编写、运行和修改程序的过程中，你能学到得更多。实践是很重要的。

为了帮助读者练习，每章都设置了一些练习题，并且可以通过前面提到的网站链接免费获取编译程序和其他资源。尽可能地多做练习题，然后向一些熟悉C语言的人请教问题的解决方法。但是建议读者不要急于编写程序，避免不经过前面讨论的步骤1~3的仔细分析就直接进入步骤4。