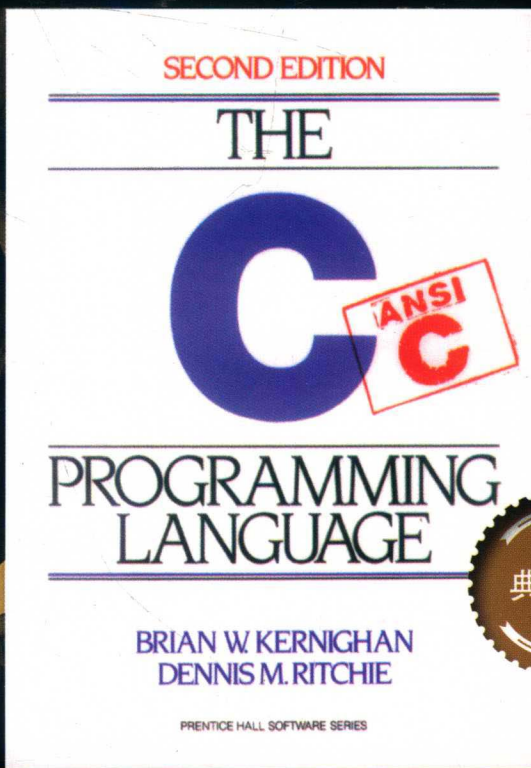


C程序设计语言

[美] 布莱恩·W. 克尼汉 (Brian W. Kernighan) 著
丹尼斯·M. 里奇 (Dennis M. Ritchie)
徐宝文 李志 译 尤晋元 审校

The C Programming Language
Second Edition



典藏版



机械工业出版社
China Machine Press

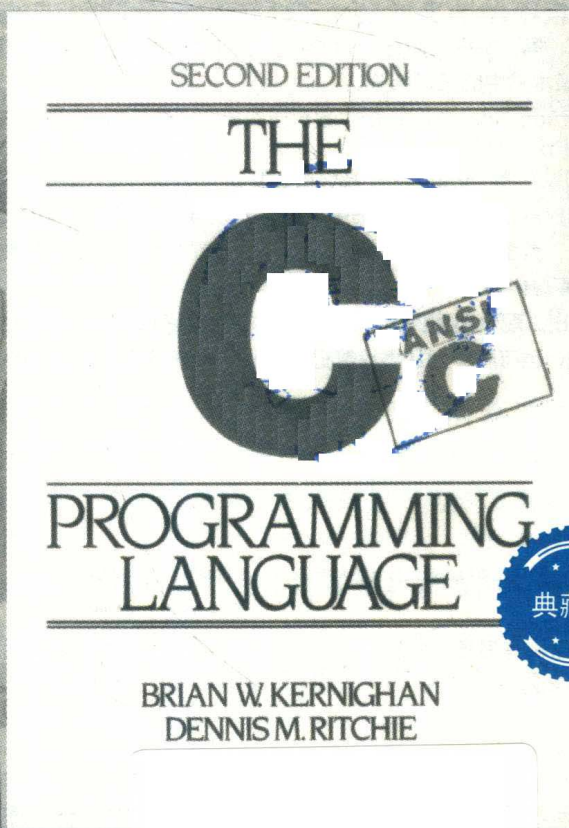
计 算 机 科 学 丛 书

第2版·新版

C程序设计语言

[美] 布莱恩·W. 克尼汉 (Brian W. Kernighan) 著
丹尼斯·M. 里奇 (Dennis M. Ritchie)
徐宝文 李志 译 尤晋元 审校

The C Programming Language
Second Edition



典藏版



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

C 程序设计语言 (第 2 版·新版) (典藏版) / (美) 布莱恩·W. 克尼汉 (Brian W. Kernighan), (美) 丹尼斯·M. 里奇 (Dennis M. Ritchie) 著; 徐宝文, 李志译. —北京: 机械工业出版社, 2019.3

(计算机科学丛书)

书名原文: The C Programming Language, Second Edition

ISBN 978-7-111-61794-5

I. C… II. ①布… ②丹… ③徐… ④李… III. C 语言 - 程序设计 - 高等学校 - 教材
IV. TP312.8

中国版本图书馆 CIP 数据核字 (2019) 第 012738 号

本书版权登记号: 图字 01-2018-8104

Authorized translation from the English language edition, entitled The C Programming Language, Second Edition, ISBN: 9780131103627, by Brian W. Kernighan, Dennis M. Ritchie, published by Pearson Education, Inc., Copyright © 1988, 1978 by Bell Telephone Laboratories, Incorporated.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by China Machine Press, Copyright © 2019.

本书中文简体字版由 Pearson Education(培生教育出版集团)授权机械工业出版社在中华人民共和国境内(不包括香港、澳门特别行政区及台湾地区)独家出版发行。未经出版者书面许可,不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

本书是由 C 语言的设计者 Dennis M. Ritchie 和 Brian W. Kernighan 编写的一部介绍标准 C 语言及其程序设计方法的权威性经典著作。书中全面、系统地讲述了 C 语言的各个特性及程序设计的基本方法,包括基本概念、类型和表达式、控制流、函数与程序结构、指针与数组、结构、输入与输出、UNIX 系统接口、标准库等内容。

本书的讲述深入浅出,配合典型例证,通俗易懂,实用性强,适合作为大专院校计算机专业或非计算机专业的 C 语言教材,也可以作为从事计算机相关软硬件开发的技术人员的参考书。

出版发行: 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: 何 方

责任校对: 李秋荣

印 刷: 北京市兆成印刷有限责任公司

版 次: 2019 年 4 月第 1 版第 1 次印刷

开 本: 185mm×260mm 1/16

印 张: 17.25

书 号: ISBN 978-7-111-61794-5

定 价: 69.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线: (010) 88378991 88379833

投稿热线: (010) 88379604

购书热线: (010) 68326294

读者信箱: hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域中取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson、McGraw-Hill、Elsevier、MIT、John Wiley & Sons、Cengage等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Andrew S. Tanenbaum、Bjarne Stroustrup、Brian W. Kernighan、Dennis Ritchie、Jim Gray、Afred V. Aho、John E. Hopcroft、Jeffrey D. Ullman、Abraham Silberschatz、William Stallings、Donald E. Knuth、John L. Hennessy、Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近500个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章科技图书出版中心

Since its original design and implementation by Dennis Ritchie in 1973, the C programming language has spread far beyond its origins at Bell Labs. It has become the common language for programmers throughout the world, and has given birth to two other major languages, C++ and Java, that build on its syntax and basic structure. C and its derivatives are the base upon which much of the world's software rests.

The spread of C required action to describe the language itself completely, and to accommodate changes in the way it was being used. In 1988, the American National Standards Institute (ANSI) created a precise standard for C that preserved its expressiveness, efficiency, small size, and ultimate control over the machine, while at the same time providing assurance that programs conforming to the standard would be portable without change from one computer and operating system to another. This standard was also accepted as an international standard under the auspices of the International Standards Organization (ISO), and thus brought the benefits of standardization to a worldwide user community.

The standards committee was aware of the multi-national use of the C language, and thus provided, both in the language itself and in the library, support for “wide characters”, which are needed to represent text in Chinese as well as other languages that do not use the Roman character set.

In spite of these evolutionary changes, C remains as it was from its inception, a compact and efficient tool for programmers of all backgrounds.

The C language, and also the Unix technology from which it grew, have been present in China for many years, as we know from visits to universities and the Chinese Academy of Sciences. Students' learning has always been made more difficult by the lack of an authoritative translation of the material describing this work into a form convenient for study in China. We are delighted that Professor Xu has made this Chinese translation of “The C Programming Language” available so that C will be more readily accessible to our colleagues in the People's Republic of China.

C程序设计语言最早是由Dennis Ritchie于1973年设计并实现的。从那时开始，C语言从其位于贝尔实验室的发源地传播到世界各地。它已经成为全球程序员的公共语言，并由此诞生了两个新的主流语言C++与Java——它们都建立在C语言的语法和基本结构的基础上。现在世界上的许多软件都是在C语言及其衍生的各种语言的基础上开发出来的。

C语言的传播需要我们对语言加以完整的描述，并适应它在使用过程中所进行的一些变化。1988年，美国国家标准协会（ANSI）为C语言制定了一个精确的标准，该标准保持了C的表达能力、效率、小规模以及对机器的最终控制，同时还保证符合标准的程序可以从一种计算机与

操作系统移植到另一种计算机与操作系统而无须改变。这个标准同时也被国际标准化组织 (ISO) 接受为国际标准，使世界各地的用户都受益。

标准委员会考虑到C语言在多民族使用的情况，在语言本身以及库中都提供了对“宽字符”的支持，这是以中文以及其他不使用罗马字符集的语言来表示文本所需要的。

除了这些渐进的变化外，C仍保持着它原来的样子——适用于各种背景的程序员的一种紧凑而有效的工具。

在我们访问中国的大学和中国科学院时，我们获悉C语言以及基于它发展起来的UNIX技术引入中国已经有很多年了。由于缺少把描述这一工作的素材翻译成在中国易于学习的形式权威译本，学生们在学习时遇到了许多困难。我们欣喜地看到徐宝文教授完成了《C程序设计语言》的中译本，我们希望它的出版有助于我们在中国的同行更容易理解C语言。

Brian W. Kernighan

Dennis M. Ritchie

《The C Programming Language》不仅在C与C++语言界，而且在整个程序设计语言教学与研究界都是耳熟能详的经典著作。最主要的两点原因是：

其一，这部著作自第1版问世后就一直深受广大读者欢迎，畅销不衰，是计算机学术界与教育界著书立说的重要参考文献。可以说，几乎所有的程序设计语言著作以及C与C++著作的作者都把这部著作作为参考文献。早在30多年前我国就翻译出版过这部著作的第1版。

其二，这部著作的原作者之一Dennis M. Ritchie是C语言的设计者，这样就保证了在著作中能完整、准确地体现与描述C语言的设计思想。本书讲述的程序设计方法以及各种语言成分的细节与用法具有权威性，这有利于读者把握C语言的精髓。

《The C Programming Language》的第1版问世于1978年，第2版自1988年面世后一直被广泛使用，至今仍未有新的版本出版，由此可见该著作内容的稳定性。

本书英文原著叙述深入浅出、条理清楚，加之辅以丰富的例证，非常通俗易懂。无论对于计算机专业人员还是非计算机专业人员，也无论用于C语言教学还是用作参考书，它都是当之无愧的正确选择。这也许就是这部著作自第1版问世以来长期畅销不衰的原因之一。

机械工业出版社曾经于2000年出版过中文版。众多高校师生在使用过程中提出了大量的宝贵意见，出版社和我们悉心听取并总结了这些意见，更加深入地领会了原书的要旨，重新认真精读了原书中的每句话，在此基础上，我们推出了新版中文版。此新版中文版在语言、术语标准化、技术细节等方面都对原中文版本进行了更进一步的雕琢。希望本书能够更好地帮助您学习C语言！

本书由东南大学计算机系徐宝文教授和上海交通大学计算机系李志博士翻译，上海交通大学计算机系的尤晋元教授审校了全书内容。在本书出版之际，我们感谢所有曾经给予我们帮助的人们！

本书的原著是经典的C语言教材，我们在翻译本书的过程中，无时无刻不感觉如履薄冰，唯恐因为才疏学浅，无法正确再现原著的风范，因此，我们一直在努力做好每件事情。但是，无论如何尽力，错误和疏漏在所难免，敬请广大读者批评指正。我们的邮件地址是lizhi_mail@263.net，随时欢迎您的每一点意见。如果您在阅读中遇到问题，或者遇到C语言的技术问题，可随时与我们联系，我们将尽力提供帮助。最后，感谢关心本书成长的每一位读者！

译者简介



徐宝文，东南大学计算机科学与工程系教授，博士生导师，曾任江苏省政协常委，江苏省计算机学会副理事长，江苏省软件行业协会副会长，中国计算机学会理事，中国软件行业协会理事。主要从事程序设计语言、软件工程等方面的教学与研究工作，承担过10多项国家级、省部级科研项目；在国内外发表论文130多篇，出版著译作10多部；担任“实用软件详解丛书”与“新世纪计算机系列教材”的主编，第五次国际青年计算机学术会议(ICYCS'99)大会主席；发起并主办过两次“全国程序设计语言发展与教学学术会议”；先后获航空航天部优秀青年教师、江苏省优秀教育工作者、江苏省优秀青年骨干教师、江苏省跨世纪学术带头人等称号。



李志，毕业于国防科技大学计算机学院，于上海交通大学获计算机科学与工程博士学位，主要从事网格计算、中间件技术等方面的研究。已经出版的译作有《IP技术基础：编址和路由》《ISDN与Cisco路由器配置》等。

审校人简介



尤晋元，上海交通大学计算机科学与工程系教授、博士生导师，国务院学位委员会学科评议组成员。主要从事操作系统、分布式对象计算、中间件技术等方面的研究，并长期从事操作系统及分布式计算等课程的教学工作。主编和翻译了多本与操作系统相关的教材和参考书，包括《UNIX操作系统教程》《UNIX环境高级编程》《操作系统设计与实现》等。

自从1978年本书第1版出版以来，计算机领域经历了一场革命。大型计算机的功能越来越强大，而个人计算机的性能也可以与十多年前的大型机相媲美。在此期间，C语言也在悄悄地演进，其发展早已超出了它仅仅作为UNIX操作系统的编程语言的初衷。

C语言普及程度的逐渐增加以及该语言本身的发展，加之很多组织开发出了与其设计有所不同的编译器，所有这一切都要求对C语言有一个比本书第1版更精确、更适应其发展的定义。1983年，美国国家标准协会（ANSI）成立了一个委员会，其目标是制定“一个无歧义性的且与具体机器无关的C语言定义”，而同时又要保持C语言原有的“精神”。结果产生了C语言的ANSI标准。

ANSI标准规范了一些在本书第1版中提及但没有具体描述的结构，特别是结构赋值和枚举。该标准还提供了一种新的函数声明形式，允许在使用过程中对函数的定义进行交叉检查。标准中还详细说明了一个具有标准输入/输出、内存管理和字符串操作等扩展函数集的标准库。它精确地说明了在C语言原始定义中并不明晰的某些特性的行为，同时还明确了C语言中与具体机器相关的一些特性。

本书第2版介绍的是ANSI标准定义的C语言。尽管我们已经注意到了该语言中变化了的地方，但我们还是决定在这里只列出它们的新形式。最重要的原因是，新旧形式之间并没有太大的差别，最明显的变化是函数的声明和定义。目前的编译器已经能够支持该标准的大部分特性。

我们将尽力保持本书第1版的简洁性。C语言并不是一种大型语言，也不需要用一本很厚的书来描述。我们在讲解一些关键特性（比如指针）时做了改进，它是C语言程序设计的核心。我们重新对以前的例子进行了精练，并在某些章节中增加了一些新例子。例如，我们通过实例程序对复杂的声明进行处理，以将复杂的声明转换为描述性的说明或反之。像前一版中的例子一样，本版中所有例子都以可被机器读取的文本形式直接通过了测试。

附录A只是一个参考手册，而非标准，我们希望通过较少的篇幅概述标准中的要点。该附录的目的是帮助程序员更好地理解语言本身，而不是为编译器的实现者提供一个精确的定义——这正是语言标准所应当扮演的角色。附录B对标准库提供的功能进行了总结，它同样是面向程序员而非编译器实现者的。附录C对ANSI标准相对于以前版本所做的变更进行了小结。

我们在第1版中曾说过：“随着使用经验的增加，使用者会越来越感到得心应手。”经过十几年的实践，我们仍然这么认为。我们希望这本书能够帮助读者学好并用好C语言。

非常感谢帮助我们完成本书的朋友们。Jon Bentley、Doug Gwyn、Doug McIlroy、Peter Nelson和Rob Pike几乎对本书手稿的每一页都提出了建议。我们非常感谢Al Aho、Dennis Allison、Joe Campbell、G. R. Emlin、Karen Fortgang、Allen Holub、Andrew Hume、Dave Kristol、John Linderman、Dave Prosser、Gene Spafford和Chris Van Wyk等人，他们仔细阅读了本书。我们也收到了来自Bill Cheswick、Mark Kernighan、Andy Koenig、Robin Lake、

Tom London、Jim Reeds、Clovis Tondo和Peter Weinberger等人的很好的建议。Dave Prosser为我们回答了很多关于ANSI标准的细节问题。我们大量地使用了Bjarne Stroustrup的C++翻译程序进行程序的局部测试。Dave Kristol为我们提供了一个ANSI C编译器以进行最终的测试。Rich Drechsler帮助我们进行了大量的排版工作。

真诚地感谢每个人！

Brian W. Kernighan

Dennis M. Ritchie

第1版前言

The C Programming Language, Second Edition

C语言是一种通用的程序设计语言，其特点包括简洁的表达式、流行的控制流和数据结构、丰富的运算符集等。C语言不是一种“很高级”的语言，也不“庞大”，并且不专用于某个特定的应用领域。但是，C语言的限制少，通用性强，这使得它比一些公认为功能强大的语言使用更方便、效率更高。

C语言最初是由Dennis Ritchie为UNIX操作系统设计的，并在DEC PDP-11计算机上实现。UNIX操作系统、C编译器和几乎所有的UNIX应用程序（包括编写本书时用到的所有软件）都是用C语言编写的。同时，还有一些适用于其他机器的编译器产品，比如IBM System/370、Honeywell 6000和Interdata 8/32等。但是，C语言不受限于任何特定的机器或系统，使用它可以很容易地编写出不经修改就可以运行在所有支持C语言的机器上的程序。

本书的目的是帮助读者学习如何用C语言编写程序。本书的开头有一个指南性的引言，目的是使新用户能尽快地开始学习；随后在不同的章节中介绍了C语言的各种主要特性；本书的附录中还包括一份参考手册。本书并不仅仅讲述语言的一些规则，而是采用阅读别人的代码、自己编写代码、修改某些代码等不同的方式来指导读者进行学习。书中的大部分例子都可以直接完整地运行，而不只是孤立的程序段。所有例子的文本都以可被机器读取的文本形式直接通过了测试。除了演示如何有效地使用语言外，我们还尽可能地在适当的时候向读者介绍一些高效的算法、良好的程序设计风格以及正确的设计原则。

本书并不是一本有关程序设计的入门性手册，它要求读者熟悉基本的程序设计概念，如变量、赋值语句、循环和函数等。尽管如此，初级的程序员仍能够阅读本书，并借此学会C语言。当然，知识越丰富，学习起来就越容易。

根据我们的经验，C语言是一种令人愉快的、具有很强表达能力的通用语言，适合于编写各种程序。它容易学习，并且随着使用经验的增加，使用者会越来越感到得心应手。我们希望本书能帮助读者用好C语言。

来自许多朋友和同事的中肯批评和建议对本书的帮助很大，也使我们在写作本书过程中受益匪浅。在此特别感谢Mike Bianchi、Jim Blue、Stu Feldman、Doug McIlroy、Bill Roome、Bob Rosin和Larry Rosler等人，他们细心地阅读了本书的多次修改版本。我们在这里还要感谢Al Aho、Steve Bourne、Dan Dvorak、Chuck Haley、Debbie Haley、Marion Harris、Rick Holt、Steve Johnson、John Mashey、Bob Mitze、Ralph Muha、Peter Nelson、Elliot Pinson、Bill Plauger、Jerry Spivack、Ken Thompson和Peter Weinberger等人，他们在不同阶段提出了非常有益的意见，此外还要感谢Mike Lesk和Joe Ossanna，他们在排版方面给予了我们很宝贵的帮助。

Brian W. Kernighan

Dennis M. Ritchie

C语言是一种通用的程序设计语言。它与UNIX系统之间具有非常密切的联系——C语言是在UNIX系统上开发的，并且，无论是UNIX系统本身还是其上运行的大部分程序，都是用C语言编写的。但是，C语言并不受限于任何一种操作系统或机器。由于它很适合用来编写编译器和操作系统，因此被称为“系统编程语言”，但它同样适合于编写不同领域中的大多数程序。

C语言的很多重要概念来源于由Martin Richards开发的BCPL语言。BCPL对C语言的影响间接地来自于B语言，它是Ken Thompson为第一个UNIX系统而于1970年在DEC PDP-7计算机上开发的。

BCPL和B语言都是“无类型”的语言。相比较而言，C语言提供了很多数据类型。其基本类型包括字符、具有多种长度的整型和浮点数等。另外，还有通过指针、数组、结构和联合派生的各种数据类型。表达式由运算符和操作数组成。任何一个表达式，包括赋值表达式或函数调用表达式，都可以是一个语句。指针提供了与具体机器无关的地址算术运算。

C语言为实现结构良好的程序提供了基本的控制流结构：语句组、条件判断（if-else）、多路选择（switch）、终止测试在顶部的循环（while、for）、终止测试在底部的循环（do）、提前跳出循环（break）等。

函数可以返回基本类型、结构、联合或指针类型的值。任何函数都可以递归调用。局部变量通常是“自动的”，即在每次函数调用时重新创建。函数定义可以不是嵌套的，但可以用块结构的方式声明变量。一个C语言程序的不同函数可以出现在多个单独编译的不同源文件中。变量可以只在函数内部有效，也可以在函数外部但仅在一个源文件中有效，还可以在整個程序中都有效。

编译的预处理阶段将对程序文本进行宏替换、包含其他源文件以及进行条件编译。

C语言是一种相对“低级”的语言。这种说法并没有什么贬义，它仅仅意味着C语言可以处理大部分计算机能够处理的对象，比如字符、数字和地址。这些对象可以通过具体机器实现的算术运算符和逻辑运算符组合在一起并移动。

C语言不提供直接处理诸如字符串、集合、列表或数组等复合对象的操作。虽然可以将整个结构作为一个单元进行拷贝，但C语言没有处理整个数组或字符串的操作。除了由函数的局部变量提供的静态定义和堆栈外，C语言没有定义任何存储器分配工具，也不提供堆和无用内存回收工具。最后，C语言本身没有提供输入/输出功能，没有READ或WRITE语句，也没有内置的文件访问方法。所有这些高层的机制必须由显式调用的函数提供。C语言的大部分实现已合理地包含了这些函数的标准集合。

类似地，C语言只提供简单的单线程控制流，即测试、循环、分組和子程序，它不提供多线程程序设计、并行操作、同步和协同例程。

尽管缺少其中的某些特性看起来好像是一个严重不足（“这就意味着必须通过调用函数来比较两个字符串吗？”），但是把语言保持在一个适度的规模会有很多益处。由于C语言相对较小，因此可以用比较少的篇幅将它描述出来，这样也很容易学会。程序员有理由期望了解、理解并真正彻底地使用完整的语言。

很多年来，C语言的定义就是本书第1版中的参考手册。1983年，美国国家标准协会(ANSI)成立了一个委员会以制定一个现代的、全面的C语言定义。最后的结果就是1988年完成的ANSI标准，即“ANSI C”。该标准的大部分特性已被当前的编译器所支持。

这个标准是基于以前的参考手册制定的。语言本身只做了相对较少的改动。这个标准的目的一就是确保现有的程序仍然有效，或者当程序无效时，编译器会对新的定义发出警告信息。

对大部分程序员来说，最重要的变化是函数声明和函数定义的新语法。现在，函数声明中可以包含描述函数实际参数的信息；相应地，定义的语法也做了改变。这些附加的信息使编译器很容易检测到因参数不匹配而导致的错误。根据我们的经验，这个扩充对语言非常有用。

新标准还对语言做了一些细微的改进：将广泛使用的结构赋值和枚举定义为语言的正式组成部分；可以进行单精度的浮点运算；明确定义了算术运算的属性，特别是无符号类型的运算；对预处理器进行了更详尽的说明。这些改进对大多数程序员的影响比较小。

2

该标准的第二个重要贡献是为C语言定义了一个函数库。它描述了诸如访问操作系统（如读写文件）、格式化输入/输出、内存分配和字符串操作等类似的很多函数。该标准还定义了一系列的标准头文件，它们为访问函数声明和数据类型声明提供了统一的方法。这就确保了使用这个函数库与宿主系统进行交互的程序之间具有兼容的行为。该函数库很大程度上与UNIX系统的“标准I/O库”相似。这个函数库已在本书的第1版中进行了描述，很多系统中都使用了它。这一点对大部分程序员来说，不会感觉到有很大的变化。

由于大多数计算机本身就支持C语言提供的数据类型和控制结构，因此只需要一个很小的运行时库就可以实现自包含程序。由于程序只能显式地调用标准库中的函数，因此在不需要的情况下就可以避免对这些函数的调用。除了其中隐藏的一些操作系统细节外，大部分库函数可以用C语言编写，并可以移植。

尽管C语言能够运行在大部分的计算机上，但它同具体的机器结构无关。只要稍加用心就可以编写出可移植的程序，即可以不加修改地运行于多种硬件上。ANSI标准明确地提出了可移植性问题，并预设了一个常量的集合，借以描述运行程序的机器的特性。

C语言不是一种强类型的语言，但随着它的发展，其类型检查机制已经得到了加强。尽管C语言的最初定义不赞成在指针和整型变量之间交换值，但并没有禁止，不过现在已经不允许这种做法了。ANSI标准要求对变量进行正确的声明和显式的强制类型转换，这在某些较完善的编译器中已经得到了实现。新的函数声明方式是另一个得到改进的地方。编译器将对大部分的数据类型错误发出警告，并且不自动执行不兼容数据类型之间的类型转换。不过，C语言保持了其初始的设计思想，即程序员了解他们在做什么，唯一的要求是程序员要明确地表达他们的意图。

同任何其他语言一样，C语言也有不完美的地方：某些运算符的优先级是不正确的，语法

的某些部分可以进一步优化。尽管如此，对于大量的程序设计应用来说，C语言仍是一种公认的非常高效的、表示能力很强的语言。

本书是按照下列结构编排的：第1章将对C语言的核心部分进行简要介绍。其目的是让读者能尽快开始编写C语言程序，因为我们深信，实际编写程序才是学习一种新语言的好方法。这部分内容的介绍假定读者对程序设计的基本元素有一定的了解。我们在这部分内容中没有解释计算机、编译等概念，也没有解释诸如 $n = n + 1$ 这样的表达式。我们将尽量在合适的地方介绍一些实用的程序设计技术，但是，本书的中心目的并不是介绍数据结构和算法。在篇幅有限的情况下，我们将专注于讲解语言本身。

3

第2章到第6章将更详细地讨论C语言的各种特性，所采用的方式将比第1章更加形式化一些。其中的重点将放在一些完整的程序例子上，而并不仅仅是一些孤立的程序段。第2章介绍基本的数据类型、运算符和表达式。第3章介绍控制流，如if-else、switch、while和for等。第4章介绍函数和程序结构——外部变量、作用域规则和多源文件等，同时还会讲述一些预处理器的知识。第5章介绍指针和地址运算。第6章介绍结构和联合。

第7章介绍标准库。标准库提供了一个与操作系统交互的公用接口。这个函数库是由ANSI标准定义的，这就意味着所有支持C语言的机器都会支持它，因此，使用这个库执行输入、输出或其他访问操作系统的操作的程序可以不加修改地运行在不同机器上。

第8章介绍C语言程序和UNIX操作系统之间的接口，我们将把重点放在输入/输出、文件系统和存储分配上。尽管本章中的某些内容是针对UNIX系统所写的，但是使用其他系统的程序员仍然会从中获益，比如深入了解如何实现标准库以及有关可移植性方面的一些建议。

附录A是一个语言参考手册。虽然C语言的语法和语义的官方正式定义是ANSI标准本身，但是，ANSI标准的文档首先是写给编译器的编写者看的，因此，对程序员来说不一定最合适。本书中的参考手册采用了一种不很严格的形式，更简洁地对C语言的定义进行了介绍。附录B是对标准库的一个总结，它同样是为程序员而非编译器实现者准备的。附录C对标准C语言相对最初的C语言版本所做的变更做了一个简短的小结。但是，如果有不一致或疑问的地方，标准本身和各个特定的编译器则是解释语言的最终权威。本书的最后提供了本书的索引。

4

出版者的话
 中文版序
 译者序
 校译者简介
 第2版前言
 第1版前言
 引言

第1章 导言1

1.1 入门1

1.2 变量与算术表达式3

1.3 for语句8

1.4 符号常量9

1.5 字符输入/输出9

1.5.1 文件复制10

1.5.2 字符计数11

1.5.3 行计数13

1.5.4 单词计数14

1.6 数组15

1.7 函数17

1.8 参数——传值调用19

1.9 字符数组20

1.10 外部变量与作用域22

第2章 类型、运算符与表达式27

2.1 变量名27

2.2 数据类型及长度27

2.3 常量28

2.4 声明31

2.5 算术运算符32

2.6 关系运算符与逻辑运算符32

2.7 类型转换33

2.8 自增运算符与自减运算符37

2.9 位运算符38

2.10 赋值运算符与表达式40

2.11 条件表达式41

2.12 运算符优先级与求值次序42

第3章 控制流45

3.1 语句与程序块45

3.2 if-else语句45

3.3 else-if语句46

3.4 switch语句48

3.5 while循环与for循环49

3.6 do-while循环52

3.7 break语句与continue语句53

3.8 goto语句与标号54

第4章 函数与程序结构57

4.1 函数的基本知识57

4.2 返回非整型值的函数60

4.3 外部变量62

4.4 作用域规则68

4.5 头文件69

4.6 静态变量70

4.7 寄存器变量71

4.8 程序块结构72

4.9 初始化72

4.10 递归73

4.11 C预处理器75

4.11.1 文件包含75

4.11.2 宏替换76

4.11.3 条件包含78

第5章 指针与数组79

5.1 指针与地址79

5.2 指针与函数参数81

5.3 指针与数组83

5.4 地址算术运算	86	8.5 实例——fopen和getc函数的实现	154
5.5 字符指针与函数	89	8.6 实例——目录列表	157
5.6 指针数组以及指向指针的指针	92	8.7 实例——存储分配程序	162
5.7 多维数组	95	附录A 参考手册	167
5.8 指针数组的初始化	97	A.1 引言	167
5.9 指针与多维数组	97	A.2 语法规则	167
5.10 命令行参数	98	A.2.1 记号	167
5.11 指向函数的指针	102	A.2.2 注释	167
5.12 复杂声明	105	A.2.3 标识符	167
第6章 结构	111	A.2.4 关键字	168
6.1 结构的基本知识	111	A.2.5 常量	168
6.2 结构与函数	113	A.2.6 字符串字面值	170
6.3 结构数组	115	A.3 语法符号	170
6.4 指向结构的指针	119	A.4 标识符的含义	170
6.5 自引用结构	121	A.4.1 存储类	171
6.6 表查找	125	A.4.2 基本类型	171
6.7 类型定义 (typedef)	127	A.4.3 派生类型	172
6.8 联合	128	A.4.4 类型限定符	172
6.9 位字段	130	A.5 对象和左值	172
第7章 输入与输出	133	A.6 转换	173
7.1 标准输入/输出	133	A.6.1 整型提升	173
7.2 格式化输出——printf函数	135	A.6.2 整型转换	173
7.3 变长参数表	136	A.6.3 整数和浮点数	173
7.4 格式化输入——scanf函数	137	A.6.4 浮点类型	173
7.5 文件访问	140	A.6.5 算术类型转换	173
7.6 错误处理——stderr和exit	143	A.6.6 指针和整数	174
7.7 行输入和行输出	144	A.6.7 void	175
7.8 其他函数	145	A.6.8 指向void的指针	175
7.8.1 字符串操作函数	145	A.7 表达式	175
7.8.2 字符类别测试和转换函数	146	A.7.1 指针生成	176
7.8.3 ungetc函数	146	A.7.2 初等表达式	176
7.8.4 命令执行函数	146	A.7.3 后缀表达式	177
7.8.5 存储管理函数	147	A.7.4 一元运算符	179
7.8.6 数学函数	147	A.7.5 强制类型转换	180
7.8.7 随机数发生器函数	148	A.7.6 乘法类运算符	180
第8章 UNIX系统接口	149	A.7.7 加法类运算符	181
8.1 文件描述符	149	A.7.8 移位运算符	181
8.2 低级I/O——read和write	150	A.7.9 关系运算符	182
8.3 open、creat、close和unlink	151	A.7.10 相等类运算符	182
8.4 随机访问——lseek	153		

A.7.11 按位与运算符	183	A.12.1 三字符序列	206
A.7.12 按位异或运算符	183	A.12.2 行连接	206
A.7.13 按位或运算符	183	A.12.3 宏定义和扩展	206
A.7.14 逻辑与运算符	183	A.12.4 文件包含	208
A.7.15 逻辑或运算符	183	A.12.5 条件编译	209
A.7.16 条件运算符	184	A.12.6 行控制	210
A.7.17 赋值表达式	184	A.12.7 错误信息生成	210
A.7.18 逗号运算符	185	A.12.8 pragma	210
A.7.19 常量表达式	185	A.12.9 空指令	210
A.8 声明	185	A.12.10 预定义名字	211
A.8.1 存储类说明符	186	A.13 语法	211
A.8.2 类型说明符	187	附录B 标准库	219
A.8.3 结构和联合声明	188	B.1 输入与输出: <stdio.h>	219
A.8.4 枚举	191	B.1.1 文件操作	220
A.8.5 声明符	191	B.1.2 格式化输出	221
A.8.6 声明符的含义	192	B.1.3 格式化输入	223
A.8.7 初始化	195	B.1.4 字符输入/输出函数	224
A.8.8 类型名	197	B.1.5 直接输入/输出函数	225
A.8.9 typedef	198	B.1.6 文件定位函数	225
A.8.10 类型等价	199	B.1.7 错误处理函数	226
A.9 语句	199	B.2 字符类别测试: <ctype.h>	226
A.9.1 带标号语句	199	B.3 字符串函数: <string.h>	227
A.9.2 表达式语句	199	B.4 数学函数: <math.h>	228
A.9.3 复合语句	200	B.5 实用函数: <stdlib.h>	229
A.9.4 选择语句	200	B.6 诊断: <assert.h>	232
A.9.5 循环语句	201	B.7 可变参数表: <stdarg.h>	232
A.9.6 跳转语句	201	B.8 非局部跳转: <setjmp.h>	232
A.10 外部声明	202	B.9 信号: <signal.h>	233
A.10.1 函数定义	202	B.10 日期与时间函数: <time.h>	234
A.10.2 外部声明	204	B.11 与具体实现相关的限制: <limits.h> 和 <float.h>	236
A.11 作用域与连接	204	附录C 变更小结	237
A.11.1 词法作用域	205	索引	241
A.11.2 连接	205		
A.12 预处理	205		