

TURING

图灵计算机科学丛书

C语言程序设计 现代方法

C Programming: A Modern Approach

[美] K. N. King 著
吕秀锋 译



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵计算机科学丛书

TP312/2547

2007

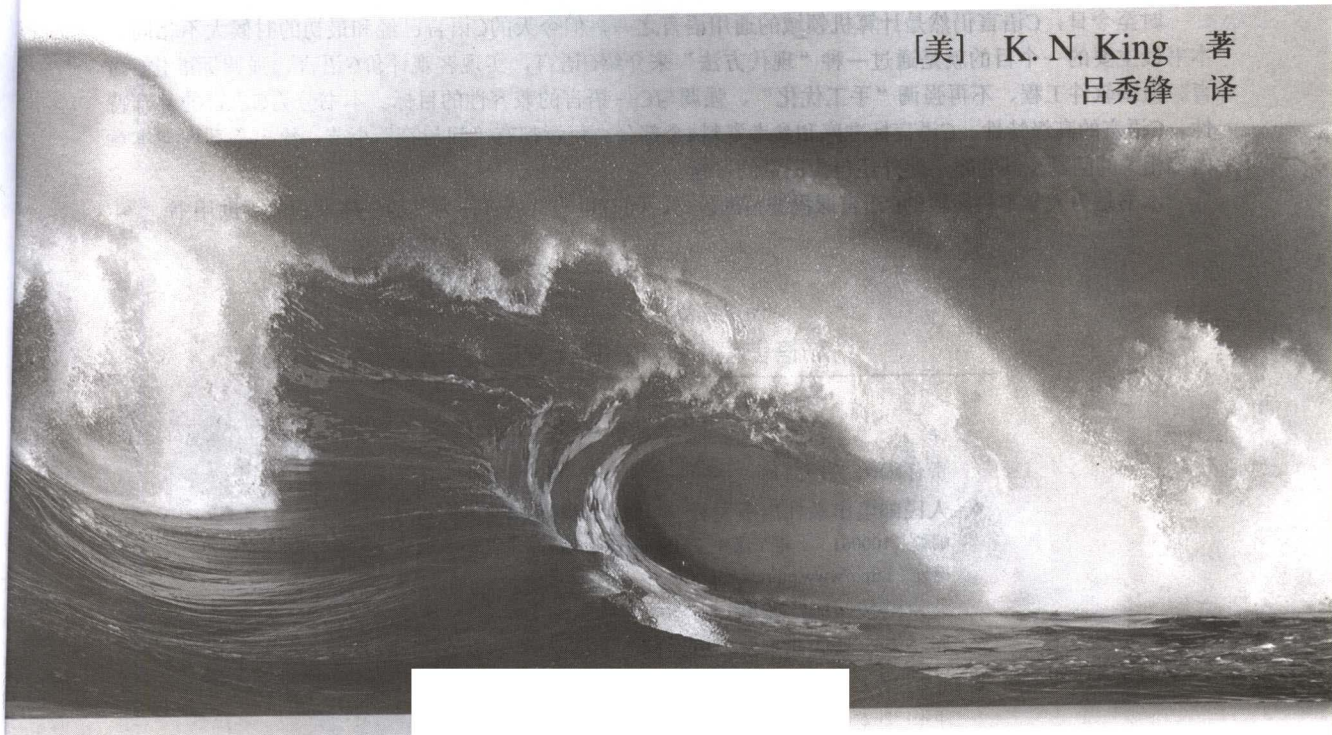
C语言程序设计 现代方法

C Programming: A Modern Approach

要 献 容 内

[美] K. N. King 著
吕秀锋 译

人民邮电出版社
北 京



图书在版编目 (CIP) 数据

C 语言程序设计: 现代方法/ (美) 金 (King, K. N.) 著;
吕秀锋译. —北京: 人民邮电出版社, 2007.11
(图灵计算机科学丛书)
ISBN 978-7-115-16707-1

I. C… II. ①金…②吕… III. C 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 130773 号

Copyright © 1996 W. W. Norton & Company, Inc.

All rights reserved. No portion of this book may be reproduced in any form or by any means without the prior written permission of the publisher.

本书中文版由 W. W. Norton 公司授权人民邮电出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

内 容 提 要

时至今日, C 语言仍然是计算机领域的通用语言之一, 但今天的 C 语言已经和最初的时候大不相同。本书最主要的一个目的就是通过一种“现代方法”来介绍 C 语言, 实现客观评价 C 语言、强调标准化 C 语言、强调软件工程、不再强调“手工优化”、强调与 C++ 语言的兼容性的目标。本书分为 C 语言的基础特性、C 语言的高级特性、C 语言标准库和参考资料 4 个部分。每章都有“问与答”小节, 给出一系列与本章内容相关的问题及其答案, 此外还包含适量的习题。

本书是为大学本科阶段的 C 语言课程编写的教材, 同时也非常适合作为其他一些课程的辅助用书。

图灵计算机科学丛书

C 语言程序设计: 现代方法

-
- ◆ 著 [美]K. N. King
 - 译 吕秀锋
 - 责任编辑 杨海玲
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 26.5
字数: 748 千字 2007 年 11 月第 1 版
印数: 1—5 000 册 2007 年 11 月北京第 1 次印刷
著作权合同登记号 图字: 01-2007-3589 号

ISBN 978-7-115-16707-1/TP

定价: 55.00 元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

对本书的赞誉

“问：有什么好的学习C的书？答：……许多活跃在新闻组comp.lang.c的人都推荐K. N. King写的《C语言程序设计：现代方法》。”

——C FAQs

“本书相对于一般的编程书而言简直是鹤立鸡群。它将清晰简明的阐述与对深入问题的深刻讨论完美结合起来。期待作者的更多著作……”

——Mike Biglan

“总之，这是一部杰出的教材，非常适合用于大学本科教学。书中的源代码示例也非常精彩，清晰地说明了C语言的工作原理。”

——*Mathematics and Computer Education*杂志

“我觉得这是我看过的最好的C语言书籍……真希望某天在我的课上可以使用它。”

——Arthur B. Maccabe, 新墨西哥大学

“我在看了USENET上的推荐后购买了本书。真是一本好书！我曾经尝试过Kochan的《C语言编程》，但是到指针和位运算部分就无法继续了。本书挽救了我！”

——Greg McNichol

“本书很好地实现了作者的既定目标：一本示例丰富的理想课堂教材……本书使我受益良多，我不仅了解到许多新知识，还大大改变了对C语言的原有认识。”

——*Computing Reviews*杂志

“作者的教学风格非常独特，各种语言要素都在需要的时候再介绍，这样初学者不会一下子就被大量信息所淹没，而比较复杂的主题放在较后章节介绍，很自然地就达到了循序渐进的效果。书中对指针的阐述是我所见过的最清晰的。……本书有很多优点，我希望其他作者也能借鉴。”

——ACCU.org (C/C++用户协会)

“本书之所以广受欢迎，是因为作者不仅精彩地阐述了C语言本身，而且在其间融合了大量有关C语言实际编程的宝贵经验和真知灼见。”

——*Choice*杂志

“有了本书，学习C语言不再困难了。系统、全面、易懂，习题也非常精彩。”

——Amazon书评

“我认为King的书在这一领域是出类拔萃的……我还没见过哪一本书像他这本这样具有精确且高水平的阐述。”

——Manuel E. Bermudez, 佛罗里达大学

“我所在的加州大学戴维斯分校使用本书作为教材。我非常喜欢，非常好懂，尤其是例子。”

——Tim Tully

“这本书实在是太赏心悦目了……写作手法如此清晰、直接……”

——Carolyn Rosner, 威斯康星大学

“我见过的最好的C语言教材！可以与K&R媲美。”

——Kevin Davis

“我已经在教学中使用本书多年。它深受学生喜爱，的确是一部杰作。我尤其欣赏书中明确地深入讨论了许多其他书里没有或者无力涉及的实践性主题。”

——Diane Horton, 加拿大多伦多大学

译者序

C语言是国际上最流行的、应用最广泛的高级编程语言之一。时至今日，它依然保持着旺盛的生命力，深受广大程序员的欢迎。作为一种“个性鲜明”的编程语言，C语言既具有高级语言的优点，又有着低级语言的特性，因此它在编写操作系统、编译器等系统软件方面有着得天独厚的优势。

目前，市面上介绍C语言的书籍众多，其中很多书的内容可以用“事无巨细，面面俱到”来形容。但是，熟记编程语言的语法规则并不是掌握程序设计的捷径，只有通过大量的实践练习才能真正做到学以致用，并最终达到灵活高效地开发软件的目标。因此，本书从软件工程师的角度，运用简洁、易懂的语言，结合丰富且实用的程序样例来介绍C语言，初学者或有经验的程序员都会从中受到启发。除了上述特色外，本书的另外一大亮点就是每个章节后面丰富的练习和问答题，这些都是原书作者结合多年的软件开发经验和教学经验提炼出来的，具有极强的针对性，非常适合作为高等院校C语言课程的教材使用。

在本书的翻译过程中，得到了许多人的帮助。首先要感谢李凌、陈寿福、马锐对本书内容提出的修改意见。同时，要感谢人民邮电出版社的编辑们为本书的翻译提供的大力支持和帮助，特别要感谢杨海玲编辑在翻译、审校过程中提供的建议。最后，要感谢我的家人，正是他们的支持和鼓励才使本书的翻译工作顺利完成。

原书的内容严谨，结构清晰，具有较强的理论性和实践性。译者力求反映本书的特点和原貌，但由于时间关系及水平所限，疏漏或不当之处在所难免，敬请广大读者批评指正。

2007年7月

前 言

在计算机领域中，把显而易见的转变为有实用价值的，这一过程是“挫折”一词的生动体现。

我首先要敞开心扉告诉大家：多年来我与C语言之间一直保持着一种“爱恨交加”的关系。一方面，我享受着轻松自如地编写C程序的快乐，并且酷爱当今多种C编译器提供的各式开发环境；另一方面，我讨厌自己在编程中容易犯错，也厌倦了要留意C语言编程中常常要求的细节。更重要的是，我憎恶许多C程序员蔑视其他编程语言的态度。让我们一起客观地评价一下C语言：C语言不是编程语言的终结（当然，C++语言也不是）；然而，C语言却是每位软件开发人员都应该掌握的一种编程语言。无论褒贬如何，C语言都已成为计算机领域的通用语言。

1975年，当C语言还是一种新兴的、尚未完全成熟的编程语言时，我就开始接触它了。随后，我有好几年没有和C打交道。然而，C语言被标准化之后，我决定重新审视一下这种语言。结果我惊喜地发现，一些C语言中致命的缺陷已经在标准化过程中得到了修正。（当然，C语言仍旧有不少值得改进的地方！）于是，我决定写一本书来展示C语言的崭新面貌，并且同时收集整理一下过去多年来C程序员所创造的智慧结晶。

目的

下面这些是本书试图实现的目标：

- **清晰易读，而且尽可能带有趣味性。**对普通读者来说，许多C语言的书籍都过于简洁了。甚至某些C语言书籍不是编写得一塌糊涂，就是平淡无趣。而我试图对C语言进行一种清晰、全面的讲解，并且决定用适当的幽默来激发读者阅读的兴趣。
- **适用于广泛的读者群。**我假设阅读本书的读者都只有一点点编程经验，而且他们都尚未精通某种具体的编程语言。我尽量减少“行话”而改用通俗易懂的词汇来定义用到的术语。同时，为了鼓励初学者，我还尝试了将某些高级内容从基本主题中分离出来。
- **有权威性，但不是学究气十足。**为了减少读者的麻烦，我在书中尽量涵盖了所有标准C的特性和库函数，包括信号、setjmp/longjmp和可变长度实际参数列表。同时，为了避免给读者造成负担，我还忽略了一些不必要的细节。
- **具备简单易学的组织结构。**根据多年教授C语言的经验，我强调循序渐进地展示C语言特性的重要性。针对有一定难度的主题，我采用了螺旋式的介绍方法。也就是，对于较难的主题先进行简要介绍，然后，在后续章节中再多次介绍该主题，逐渐增加细节内容。本书的进度是经过深思熟虑的。每章都按照循序渐进的方式进行组织，并且前后内容由浅入深，相互呼应。对于大多数学生来说，这种循序渐进的方法是最合适的：既避免产生无聊的内容，又防止出现“信息超载”。
- **深入探讨语言特性。**我不是仅描述语言的每个特性，或者只展示几个简单的特性应用的例题，而是尝试深入讲解每一种特性，并且探讨如何将这特性应用到实际问题中。

- **强调编写风格。**对每位C程序员来说，采用一种统一的代码编写风格是非常重要的。但是，与其指定某种风格，我更愿意给出多种编写风格，让读者自己做出合理的选择，因为了解多种编写风格对阅读别人的程序是很有帮助的（有些程序员经常要花费大量时间阅读别人的程序）。
- **避免依赖任何特定的计算机、编译器或者操作系统。**因为C语言可以应用在如此多样的平台上，所以我试图避免编写的程序依赖于任何特定的计算机、编译器或操作系统。当然，使用C这样的语言完全忽略机器的细节也是不可能的。当此类问题不可避免时，我都以16位计算机和32位计算机的两种体系结构进行举例说明。当示例要依赖于某种特定操作系统时，我会讨论DOS和UNIX两种系统。
- **用图示的方法阐明关键概念。**因为图在理解C语言方方面面都起着至关重要的作用，所以我在书中加入了尽可能多的图。特别是我还试图通过图显示运算中不同阶段的数据状态来动态地描述算法。

现代方法到底是什么

本书最主要的一个目的就是想通过一种“现代方法”来介绍C语言。我试图通过以下这些途径来达成目标：

- **正确看待C语言。**我没有把C语言看成是唯一值得学习的编程语言，而是把它作为众多有用语言中的一种进行介绍。我在书中提到了最适合用C语言编写的程序类型。此外，我还展示了如何扬长避短地使用C语言。
- **强调标准化C语言。**我较少关注旧版C语言。只在某些章节偶尔提到经典（K&R）C，多在“问与答”部分。附录C列出了标准C和经典C之间的主要差异。
- **揭穿神话。**现今的某些书籍的作者常常会在C语言某些常见的假设上争论不休，而我却乐于揭穿C语言的某些神话，或者说只是想对长久以来构成C语言传说的某些信条提出挑战。例如，有种说法始终认为指针的算术运算一定比数组下标操作快。我重新审查了C语言的旧惯例，并且保留了那些仍然有帮助的惯例。
- **强调软件工程。**我把C语言视为一种成熟的软件工程工具，因此我强调如何运用C语言来处理大规模程序开发过程中产生的问题。我坚持程序要易读、可维护、可靠且容易移植，同时还特别看重信息隐藏。
- **不是在开始就介绍C语言的低级特性。**虽然这些特性曾为早期C语言的各种系统编程提供了便利，但是现在它们已经不再适宜使用，因为C语言已经应用于大量不同的程序中。本书没有像其他大多数C语言书籍那样把这部分内容放在前面章节进行介绍，而是推迟到第20章再进行讲述。
- **不再强调“手工优化”。**某些书籍指导读者编写并不简单、清晰的代码，仅仅是为了稍稍提高程序效率。然而，面对现今C语言编译器的大量优化技术，这种代码优化工作常常不再必要；事实上，它们适得其反。
- **强调与C++语言的兼容性。**有关这方面的内容，我会稍后进行详细介绍。

“问与答”部分

每章的末尾都有一个“问与答”部分，汇集了与本章内容相关的问题及其答案。“问与答”部分的内容包括：

- **常见问题。**我尽力回答了某些频繁出现在我的课堂里、其他书籍中，以及和C语言相关的新闻组里的问题。
- **对一些难以理解的问题的进一步讨论和澄清。**虽然具有多种编程语言经验的读者会满足于简明扼要的说明和少量的示例，但是缺乏经验的读者却需要更多的内容以帮助理解。
- **非主流的问题。**某些问题所引发的并不是所有读者都感兴趣的技术讨论。
- **某些对普通读者来说过于超前或深奥的内容。**这类问题都用星号(*)进行了标记。好奇且有一定编程经验的读者也许希望立刻深入研究这些问题，而另外一些读者则需要首次阅读时跳过这部分内容。提示：这类问题往往引用后续章节的内容。
- **多种C语言编译器的常见差异。**我讨论了一些经常用到的（而非标准的）特性。DOS系统和UNIX系统都对这类特性提供支持。

“问与答”中的某些问题与对应章中某些特定的内容直接相关，这些特定内容会在对应位置用一种特殊的图标 **Q&A** 标记，从而提示读者有进一步的信息。

其他特色

除了“问与答”部分，我还加入了另外一些有用的特色，其中一些特色用简单但是独特的图标进行了标识。

- **警告 (△)** 警示读者一些常见的陷阱。C语言以其陷阱多而出名。要记录所有的陷阱非常困难。我试图挑选出最常见或最重要的陷阱供大家参考。
- **引用 (➤前言)** 提供一种类似超文本的能力来查找信息。多数引用指向稍后的章节中才能提到的主题，但是确有某些引用指向先前的主题供读者回顾。
- **惯用法**是经常可以在C语言程序中看到的代码模式。它常被标记出来以便于速查参考。
- **可移植性技巧**为编写不依赖于特定的计算机、编译器或操作系统的程序所需的技巧和心得。
- **附加说明**包含一些严格来讲并不属于C语言的内容，但却是每位熟练的C程序员都应该知道的知识，比如无符号整数、IEEE浮点数标准以及Unicode编码等。（附加说明的示例可参见本页下面的“源代码”说明。）
- **附录**提供有价值的参考资料信息。

程序

选择作为例证的程序并不是件轻松的工作。如果程序过于简洁和做作，那么读者将无法体会如何将这特性应用于现实世界里。一方面，如果程序过于真实，那么它的要点将很容易被忽略在过多的细节中。我采取了折中方案。在首次介绍概念时，先通过小而简单的示例使内容清晰，再逐步建立完整的程序。我没有使用过长的程序，因为根据我个人的经验，授课者没有时间介绍这些，而学生们也不会有耐心去阅读。但是，我没有忽视创作大程序所引发的问题，这些在第15章和第19章中进行了详细的介绍。

源代码

本书中所有程序的源代码都可以从图灵网站 (<http://www.turingbook.com>) 下载。有关本书原版的校正、修改和最新消息可以从<http://www.gsu.edu/~matknk/cbook>获取。

C++语言的介绍

本书从一开始就考虑到要完全兼容C++语言，因此读者不会被培养出那些稍后必须忘掉的习惯。本书通过下面3种方式为读者继续学习C++语言做了铺垫：

- 强化现代设计规则，比如信息隐藏。
- 一些分散在书中的C++语言的简要论述。其中，每次论述都用特殊的**C++**符号标记。
- 第19章提供的详细的C++语言概述。

读者

本书是为大学本科阶段的C语言课程编写的教材。具有其他高级语言或汇编语言的编程经验会很有帮助，不过这些经验对于会用计算机的读者（我的编辑称他们为“熟练的初学者”）来说并不是必需的。

因为本书内容齐备，自成一体，并且既可用于学习又可作为参考，所以它非常适合作为其他一些课程的辅助读物，如数据结构、编译器设计、操作系统、计算机图形学及其他要用C语言进行项目设计的课程。

“问与答”部分以及对实际问题的强调，使得本书也可以引起另外一些读者的兴趣，如培训班学员，或是自学C语言的人。

组织结构

本书分为4个部分：

- **C语言的基础特性。**第1章~第10章包含的C语言内容足够帮助读者编写出使用数组和函数的单个文件程序。
- **C语言的高级特性。**第11章~第20章建立在前面各章的内容上。这些章的内容有一定的难度，深入介绍了指针、字符串、预处理器、结构、联合、枚举以及C语言的低级特性。此外，第15章和第19章提供了程序设计方面的指导。
- **C语言标准库。**第21章~第26章集中介绍C语言库。这个庞大的函数集合是与每种编译器一起提供的。虽然一部分内容适合讲解，但这些章更适合作为参考资料来使用。
- **参考资料。**附录A介绍了C语言的完整语法，并用注解来解释一些较为模糊的论点。附录B给出了C语言运算符的完整列表。附录C描述了标准C和经典C之间的差异。附录D按字母顺序列出了C标准库中的全部函数，每个都予以充分说明。附录E列出了ASCII字符集。还有一个带注解的参考文献为读者指明其他信息的来源。

C语言的全面课程应该按顺序讲授前20章的内容，并且根据需要选取第21章~第26章中的某些主题。短期课程可以忽略以下一些主题而不会失去内容的连贯性：9.6节（递归函数）、12.4节（指针和多维数组）、14.5节（其他指令）、17.7节（指向函数的指针）、第19章（程序设计）以及20.2节（结构中的位域）和20.3节（其他低级技术）。

练习

作为教材，拥有大量多样化的精选习题显然是非常必要的。我准备了300多道难度各异的习

题。有些习题是简短的训练题，虽然它们不是最激动人心的（事实上，它们可能是完全无趣的），但是我认为这些题对培养C语言开发技巧是必要的，就如同词汇训练是外语教材必备的，或者在代数教材中数学问题也是必要的一样。除了训练题，我还加入了大量的简答题和编程练习。虽然简答题的答案通常都很简短，但是此类习题比训练题需要更多的思考。编程练习要求读者编写小程序或大程序中的某个片段。

少数练习没有明确的答案（有些特别挑剔的人称这些是“刁钻问题”）。因为C语言程序经常包含大量这类代码案例，所以我认为有必要提供一些这样的练习，并用星号(*)进行了标注。对于有星号的练习一定要小心：要么格外小心，认真考虑，要么干脆绕开它。

反馈

为了本书能够精确，我付出了极大的努力。然而，任何书籍都不可避免地会有一些错误。如果读者发现了错误，请通过knking@gsu.edu这个电子邮箱或以下地址与我联系。

K. N. King

Department of Mathematics and Computer Science

Georgia State University

University Plaza

Atlanta, GA 30303-3083

我也同样期望听到读者的其他反馈，比如，哪些内容你最认可，需要删除哪些内容，又或希望添加哪些内容等。

致谢

首先，我要感谢本书的编辑，Norton出版社的Joe Wisonvsky，非常欣赏他的聪明才智和良好的鉴赏力。同时，还要感谢我的前任编辑Jim Jordan，他认为这世界需要一本新的C语言书。还有一位要感谢的是Norton出版社的Deborah Gerish，感谢她对书稿所做的文字编辑工作。

本书的前身是我为TopSpeed C编译器编写的指导手册。我要感谢Karin Ellison和Niels Jensen，正是他们让我有机会编写那本手册，并且允许我在本书中再次使用其中的资料。

我还要感谢下面这些对书稿进行部分或全部审稿的人：

Susan Anderson-Freed和Lisa J. Brown，伊利诺伊韦思利安大学

Manuel E. Bermudez，佛罗里达大学

Steven C. Cater，佐治亚大学

Patrick Harrison，美国海军学院

Brian Harvey，加利福尼亚大学伯克利分校

Henry H. Leitner，哈佛大学

Darrell Long，加利福尼亚大学圣克鲁兹分校

Arthur B. Maccabe，新墨西哥大学

Carolyn Rosner，威斯康星大学

Patrick Terry，罗德斯大学

在这里需要特别提到的是Brian Harvey和Patrick Terry，他们提供了超出职责范围的协助，对他们所做的详尽注释我感激不尽，同时我保证会原谅他们对我偶尔严厉的评论。

我也从亚特兰大的朋友和同事那里得到了大量有价值的反馈信息。他们是：一直乐于讨论

C和C++语言细节的Geoff George，在最终书稿中发现不少错误的Marge Hicks，以及对早期书稿发表很多有用意见的Scott Owen。衷心感谢我的部门主管Fred Turner给予我的支持与鼓励。感谢许多学生提供的反馈，包括用敏锐的眼光阅读了早期书稿的Terry Turner，以及Nina Dalal、Jay Schneider和Michael Sigmond。

感谢Susan Cole不仅仔细地阅读了全部书稿，还为我创造了便于写作的良好家庭氛围。没有她的爱和理解，我是不可能完成本书的。我还必须感谢我的猫咪Bronco和Dennis的帮助，在写作本书的过程中，它们一直陪伴着我。

最后，我还要感谢已故的Alan J. Perlis^①。他的警句出现在本书每一章的开始。在20世纪70年代中期我在耶鲁大学期间，曾经有幸在Alan的指导下进行了短暂学习。我想当他发现他的警句出现在一本C语言书中时一定会非常高兴的。

^① Alan J. Perlis (1922—1990) 是计算机科学先驱，1966年首届图灵奖得主。——编者注

目 录

第1章 C语言概述	1	3.1.1 转换说明	24
1.1 C语言的历史	1	3.1.2 程序: 用printf函数格式化数	25
1.1.1 起源	1	3.1.3 转义序列	25
1.1.2 标准化	1	3.2 scanf函数	26
1.1.3 C++语言	2	3.2.1 scanf函数的工作方法	27
1.2 C语言的优缺点	3	3.2.2 格式串中的普通字符	28
1.2.1 C语言的优点	3	3.2.3 混淆printf函数和scanf函数	29
1.2.2 C语言的缺点	4	3.2.4 程序: 计算持有的股票的价值	29
1.2.3 高效地使用C语言	5	问与答	30
问与答	5	练习	31
第2章 C语言基本概念	7	第4章 表达式	33
2.1 编写一个简单的C程序	7	4.1 算术运算符	33
2.1.1 程序: 显示双关语	7	4.1.1 运算符的优先级和结合性	34
2.1.2 编译和链接	8	4.1.2 程序: 计算通用产品代码的 校验位	35
2.2 简单程序的通用形式	8	4.2 赋值运算符	36
2.2.1 指令	9	4.2.1 简单赋值	36
2.2.2 函数	9	4.2.2 左值	37
2.2.3 语句	10	4.2.3 复合赋值	37
2.2.4 显示字符串	10	4.3 自增运算符和自减运算符	38
2.3 注释	11	4.4 表达式求值	39
2.4 变量和赋值	12	4.5 表达式语句	41
2.4.1 类型	12	问与答	42
2.4.2 声明	12	练习	43
2.4.3 赋值	13	第5章 选择语句	45
2.4.4 显示变量的值	13	5.1 逻辑表达式	45
2.4.5 程序: 计算箱子的空间重量	13	5.1.1 关系运算符	45
2.4.6 初始化	14	5.1.2 判等运算符	46
2.4.7 显示表达式的值	15	5.1.3 逻辑运算符	46
2.5 读入输入	15	5.2 if语句	47
2.6 定义常量	16	5.2.1 复合语句	48
2.7 标识符	17	5.2.2 else子句	48
2.8 C语言程序的布局	18	5.2.3 级联式if语句	49
问与答	20	5.2.4 程序: 计算股票经纪人的佣金	50
练习	21	5.2.5 “悬空else”的问题	51
第3章 格式化的输入/输出	23	5.2.6 条件表达式	51
3.1 printf函数	23	5.2.7 布尔值	52

5.3 switch语句	53	练习	95
5.3.1 break语句的作用	55		
5.3.2 程序: 显示法定格式的日期	55		
问与答	56		
练习	58		
第6章 循环	61	第8章 数组	98
6.1 while语句	61	8.1 一维数组	98
6.1.1 无限循环	62	8.1.1 数组下标	98
6.1.2 程序: 显示平方值的表格	63	8.1.2 程序: 数列反向	100
6.1.3 程序: 数列求和	63	8.1.3 数组初始化	100
6.2 do语句	64	8.1.4 程序: 检查数中重复出现的数字	101
6.3 for语句	65	8.1.5 对数组使用sizeof运算符	101
6.3.1 for语句的惯用法	66	8.1.6 程序: 计算利息	102
6.3.2 在for语句中省略表达式	67	8.2 多维数组	103
6.3.3 逗号运算符	67	8.2.1 多维数组初始化	104
6.3.4 程序: 显示平方值的表格(改进版)	68	8.2.2 常量数组	105
6.4 退出循环	69	8.2.3 程序: 发牌	105
6.4.1 break语句	69	问与答	106
6.4.2 continue语句	70	练习	107
6.4.3 goto语句	71	第9章 函数	110
6.4.4 程序: 账目簿结算	71	9.1 函数的定义和调用	110
6.5 空语句	73	9.1.1 程序: 计算平均值	110
问与答	74	9.1.2 程序: 显示倒数计数	111
练习	76	9.1.3 程序: 显示双关语(改进版)	112
第7章 基本类型	78	9.1.4 函数定义	113
7.1 整型	78	9.1.5 函数调用	114
7.1.1 整型常量	79	9.1.6 程序: 判定素数	115
7.1.2 读/写整数	80	9.2 函数声明	116
7.1.3 程序: 数列求和(改进版)	81	9.3 实际参数	117
7.2 浮点型	81	9.3.1 实际参数的转换	118
7.2.1 浮点常量	82	9.3.2 数组型实际参数	119
7.2.2 读/写浮点数	83	9.4 return语句	120
7.3 字符型	83	9.5 程序终止	121
7.3.1 转义序列	84	9.6 递归函数	122
7.3.2 字符处理函数	85	9.6.1 快速排序算法	123
7.3.3 读/写字符	86	9.6.2 程序: 快速排序	124
7.3.4 程序: 确定消息的长度	87	问与答	125
7.4 sizeof运算符	88	练习	128
7.5 类型转换	89	第10章 程序结构	131
7.5.1 常用算术转换	89	10.1 局部变量	131
7.5.2 赋值中的转换	90	10.2 外部变量	132
7.5.3 强制类型转换	91	10.2.1 程序: 用外部变量实现栈	132
7.6 类型定义	92	10.2.2 外部变量的利与弊	133
问与答	93	10.2.3 程序: 猜数	134
		10.3 程序块	137
		10.4 作用域	137
		10.5 构建C程序	138

问与答	144	13.3.1 用printf函数和puts函数 写字符串	173
练习	144	13.3.2 用scanf函数 和gets函数 读字符串	173
第 11 章 指针	146	13.3.3 逐个字符读字符串	174
11.1 指针变量	146	13.4 访问字符串中的字符	175
11.2 取地址运算符和间接寻址运算符	147	13.5 使用C语言的字符串库	176
11.2.1 取地址运算符	147	13.5.1 strcpy函数	177
11.2.2 间接寻址运算符	148	13.5.2 strcat函数	177
11.3 指针赋值	148	13.5.3 strcmp函数	178
11.4 指针作为实际参数	149	13.5.4 strlen函数	178
11.4.1 程序: 找出数组中的最大元素 和最小元素	151	13.5.5 程序: 显示一个月的提示 列表	179
11.4.2 用const保护实际参数	152	13.6 字符串惯用法	181
11.5 指针作为返回值	153	13.6.1 搜索字符串的结尾	181
问与答	153	13.6.2 复制字符串	182
练习	155	13.7 字符串数组	184
第 12 章 指针和数组	156	13.7.1 命令行参数	185
12.1 指针的算术运算	156	13.7.2 程序: 核对行星的名字	186
12.1.1 指针加上整数	157	问与答	187
12.1.2 指针减去整数	157	练习	189
12.1.3 指针相减	158	第 14 章 预处理器	192
12.1.4 指针比较	158	14.1 预处理器的工作方式	192
12.2 指针用于数组处理	158	14.2 预处理指令	194
12.3 用数组名作为指针	160	14.3 宏定义	194
12.3.1 程序: 数列反向 (改进版)	161	14.3.1 简单的宏	194
12.3.2 数组型实际参数 (改进版)	161	14.3.2 带参数的宏	196
12.3.3 用指针作为数组名	162	14.3.3 #运算符	197
12.4 指针和多维数组	163	14.3.4 ##运算符	198
12.4.1 处理多维数组的元素	163	14.3.5 宏的通用属性	199
12.4.2 处理多维数组的行	163	14.3.6 宏定义中的圆括号	199
12.4.3 用多维数组名作为指针	164	14.3.7 创建较长的宏	200
问与答	164	14.3.8 预定义宏	201
练习	166	14.4 条件编译	202
第 13 章 字符串	168	14.4.1 #if指令和#endif指令	202
13.1 字符串字面量	168	14.4.2 defined运算符	203
13.1.1 字符串字面量中的转义序列	168	14.4.3 #ifdef指令和#ifndef指令	203
13.1.2 延续字符串字面量	169	14.4.4 #elif指令和#else指令	203
13.1.3 如何存储字符串字面量	169	14.4.5 使用条件编译	204
13.1.4 字符串字面量的操作	170	14.5 其他指令	205
13.1.5 字符串字面量与字符常量	170	14.5.1 #error指令	205
13.2 字符串变量	170	14.5.2 #line指令	205
13.2.1 初始化字符串变量	171	14.5.3 #pragma指令	206
13.2.2 字符数组与字符指针	172	问与答	206
13.3 字符串的读/写	173	练习	208

第 15 章 编写大规模程序211	第 17 章 指针的高级应用250
15.1 源文件.....211	17.1 动态存储分配.....250
15.2 头文件.....212	17.1.1 内存分配函数.....250
15.2.1 #include指令.....212	17.1.2 空指针.....251
15.2.2 共享宏定义和类型定义.....213	17.2 动态分配字符串.....251
15.2.3 共享函数原型.....214	17.2.1 使用malloc函数为字符串
15.2.4 共享变量声明.....215	分配内存.....252
15.2.5 嵌套包含.....216	17.2.2 在字符串函数中使用动态
15.2.6 保护头文件.....216	存储分配.....252
15.2.7 头文件中的#error指令.....217	17.2.3 动态分配字符串的数组.....253
15.3 把程序划分成多个文件.....217	17.2.4 程序: 显示一个月的提示
15.4 构建多文件程序.....223	列表(改进版).....253
15.4.1 makefile.....223	17.3 动态分配数组.....254
15.4.2 链接期间的错误.....224	17.3.1 使用malloc函数为数组
15.4.3 重新构建程序.....225	分配存储空间.....255
15.4.4 在程序外定义宏.....226	17.3.2 calloc函数.....255
问与答.....227	17.3.3 realloc函数.....256
练习.....228	17.4 释放存储.....256
第 16 章 结构、联合和枚举229	17.4.1 free函数.....257
16.1 结构变量.....229	17.4.2 “悬空指针”问题.....257
16.1.1 结构变量的声明.....229	17.5 链表.....257
16.1.2 结构变量的初始化.....230	17.5.1 声明结点类型.....258
16.1.3 对结构的操作.....231	17.5.2 创建结点.....258
16.2 结构类型.....232	17.5.3 ->运算符.....259
16.2.1 结构标记的声明.....232	17.5.4 在链表的开始处插入结点.....259
16.2.2 结构类型的定义.....233	17.5.5 搜索链表.....261
16.2.3 结构类型的实际参数和	17.5.6 从链表中删除结点.....262
返回值.....233	17.5.7 链表排序.....264
16.3 数组和结构的嵌套.....234	17.5.8 程序: 维护零件数据库
16.3.1 嵌套的结构.....234	(改进版).....264
16.3.2 结构数组.....235	17.6 指向指针的指针.....268
16.3.3 结构数组的初始化.....235	17.7 指向函数的指针.....269
16.3.4 程序: 维护零件数据库.....236	17.7.1 函数指针作为实际参数.....269
16.4 联合.....241	17.7.2 qsort函数.....270
16.4.1 使用联合来节省空间.....242	17.7.3 函数指针的其他用途.....271
16.4.2 使用联合来构造混合的	17.7.4 程序: 列三角函数表.....272
数据结构.....243	问与答.....273
16.4.3 为联合添加“标记字段”.....243	练习.....276
16.5 枚举.....244	第 18 章 声明278
16.5.1 枚举标记和枚举类型.....245	18.1 声明的语法.....278
16.5.2 枚举作为整数.....245	18.2 存储类型.....279
16.5.3 用枚举声明“标记字段”.....246	18.2.1 变量的特性.....279
问与答.....246	18.2.2 auto存储类型.....280
练习.....247	18.2.3 static存储类型.....280

18.2.4	extern存储类型	281	20.3.1	定义依赖机器的类型	320
18.2.5	register存储类型	282	20.3.2	用联合从多个视角看待数据	320
18.2.6	函数的存储类型	282	20.3.3	将指针作为地址使用	321
18.2.7	小结	283	20.3.4	程序: 设置Num Lock键	322
18.3	类型限定符	284	20.3.5	volatile类型限定符	323
18.4	声明符	284	问与答		323
18.4.1	解释复杂声明	285	练习		323
18.4.2	使用类型定义来简化声明	286	第 21 章 标准库		325
18.5	初始化式	287	21.1	标准库的使用	325
问与答		288	21.1.1	对标准库中使用的名字	
练习		289	一些限制		325
第 19 章 程序设计		291	21.1.2	使用宏隐藏函数	325
19.1	模块	291	21.2	标准库概述	326
19.1.1	内聚性与耦合性	293	21.3	<stddef.h>: 常用定义	327
19.1.2	模块的类型	293	问与答		328
19.2	信息隐藏	293	练习		328
19.3	抽象数据类型	296	第 22 章 输入/输出		329
19.4	C++语言	297	22.1	流	329
19.4.1	C语言与C++语言之间的		22.1.1	文件指针	330
差异		297	22.1.2	标准流和重定向	330
19.4.2	类	299	22.1.3	文本文件与二进制文件	330
19.4.3	类定义	300	22.2	文件操作	331
19.4.4	成员函数	300	22.2.1	打开文件	332
19.4.5	构造函数	302	22.2.2	模式	332
19.4.6	构造函数和动态存储分配	303	22.2.3	关闭文件	333
19.4.7	析构函数	304	22.2.4	为流附加文件	333
19.4.8	重载	304	22.2.5	从命令行获取文件名	334
19.4.9	面向对象编程	306	22.2.6	程序: 检查文件是否可以	
19.4.10	派生	306	打开		334
19.4.11	虚函数	308	22.2.7	临时文件	335
19.4.12	模板	310	22.2.8	文件缓冲	336
19.4.13	异常处理	310	22.2.9	其他文件操作	337
问与答		311	22.3	格式化的输入/输出	337
练习		312	22.3.1	...printf类函数	337
第 20 章 低级程序设计		314	22.3.2	...printf类函数的转换说明	338
20.1	按位运算符	314	22.3.3	...printf类函数的转换说明	
20.1.1	移位运算符	314	示例		339
20.1.2	按位求反运算符、按位与		22.3.4	...scanf类函数	341
运算符、按位异或运算符		315	22.3.5	...scanf类函数的格式化字	
和按位或运算符		315	字符串		342
20.1.3	用按位运算符访问位	316	22.3.6	...scanf类函数的转换说明	342
20.1.4	用按位运算符访问位域	317	22.3.7	scanf函数的示例	343
20.1.5	程序: XOR加密	317	22.3.8	检测文件末尾和错误条件	344
20.2	结构中的位域	318	22.4	字符的输入/输出	346
20.3	其他低级技术	320	22.4.1	输出函数	346

22.4.2	输入函数	346	练习	381
22.4.3	程序: 复制文件	347		
22.5	行的输入/输出	348	第 25 章 国际化特性	383
22.5.1	输出函数	348	25.1 <locale.h>: 本地化	383
22.5.2	输入函数	348	25.1.1 类别	383
22.6	块的输入/输出	349	25.1.2 setlocale函数	384
22.7	文件的定位	350	25.1.3 localeconv函数	384
22.8	字符串的输入/输出	352	25.2 多字节字符和宽字符	386
问与答		353	25.2.1 多字节字符	387
练习		356	25.2.2 宽字符	387
第 23 章 库对数值和字符数据的支持		360	25.2.3 多字节字符函数	388
23.1	<float.h>: 浮点型的特性	360	25.2.4 多字节字符串函数	389
23.2	<limits.h>: 整数类型的大小	361	25.3 三字符序列	389
23.3	<math.h>: 数学计算	362	问与答	390
23.3.1	错误	362	练习	391
23.3.2	三角函数	363	第 26 章 其他库函数	392
23.3.3	双曲函数	363	26.1 <stdarg.h>: 可变长度实参	392
23.3.4	指数函数和对数函数	363	26.1.1 调用带有可变实参列表的 函数	393
23.3.5	幂函数	364	26.1.2 v...printf类函数	393
23.3.6	就近取整函数、绝对值函数 和取余函数	364	26.2 <stdlib.h>: 通用的实用工具	394
23.4	<ctype.h>: 字符处理	365	26.2.1 字符串转换函数	394
23.4.1	字符测试函数	365	26.2.2 程序: 测试字符串转换 函数	395
23.4.2	程序: 测试字符测试函数	366	26.2.3 伪随机序列生成函数	397
23.4.3	字符大小写转换函数	367	26.2.4 程序: 测试伪随机序列生成 函数	397
23.4.4	程序: 测试大小写转换函数	367	26.2.5 与环境的通信	398
23.5	<string.h>: 字符串处理	367	26.2.6 搜索和排序实用工具	399
23.5.1	复制函数	368	26.2.7 程序: 确定航空里程	399
23.5.2	拼接函数	368	26.2.8 整数算术运算函数	400
23.5.3	比较函数	369	26.3 <time.h>: 日期和时间	401
23.5.4	搜索函数	370	26.3.1 时间处理函数	401
23.5.5	其他函数	372	26.3.2 时间转换函数	403
问与答		372	26.3.3 程序: 显示日期和时间	405
练习		372	问与答	406
第 24 章 错误处理		374	练习	407
24.1	<assert.h>: 诊断	374	附录 A C 语言语法 (图灵网站下载)	
24.2	<errno.h>: 错误	375	附录 B C 语言运算符 (图灵网站下载)	
24.3	<signal.h>: 信号处理	376	附录 C 标准 C 与经典 C 的比较 (图灵网站 下载)	
24.3.1	信号宏	376	附录 D 标准库函数 (图灵网站下载)	
24.3.2	signal函数	377	附录 E ASCII 字符集 (图灵网站下载)	
24.3.3	预定义的信号处理函数	377	参考文献 (图灵网站下载)	
24.3.4	raise函数	378	索引 (图灵网站下载)	
24.3.5	程序: 测试信号	378		
24.4	<setjmp.h>: 非局部跳转	379		
问与答		380		