

计 算 机 科 学 丛 书

C程序设计 语 言

The C
Programming
Language
Second Edition

(美) Brian W. Kernighan 著
Dennis M. Ritchie
徐宝文 等译



机械工业出版社
China Machine Press

Prentice Hall

计算机科学丛书

C 程序设计语言

(第2版)

(美) Brian W. Kernighan 著
Dennis M. Ritchie

徐宝文 李帮清 刘杰 刘圆 等译



机械工业出版社
China Machine Press

本书是由C语言研制者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。全书共分8章及4个附录,全面、系统地讲述了C语言的各个方面及其程序设计的基本方法,内容深入浅出,配合例证,通俗易懂。适合于大专院校师生以及从事计算机软硬件研究、开发与应用的人员使用。

Brian W. Kernighan, Dennis M. Ritchie: The C Programming Language, Second Edition.
Authorized translation from the English language edition published by Prentice Hall PTR.
Copyright © 1988 by Prentice Hall, PTR.
All rights reserved.
Chinese simplified language edition published by China Machine Press.
Copyright © 2001 by China Machine Press.

本书中文简体字版由美国Prentice Hall PTR公司授权机械工业出版社独家出版。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。

本书版权登记号:图字:01-1999-2347

图书在版编目(CIP)数据

C 程序设计语言(第2版)/(美)柯尼汉(Kernighan, B. W.), (美)里奇(Ritchie, D. M.)著;徐宝文等译. -北京:机械工业出版社,2001.3

(计算机科学丛书)

书名原文: The C Programming Language, Second Edition

ISBN 7-111-07589-7

I. C… II. ①柯… ②里… ③徐… III. C语言-程序设计 IV. TP312

中国版本图书馆CIP数据核字(2001)第03455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:陈 谊 赵阿玲

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2001年3月第1版第1次印刷

787mm × 1092mm 1/16 · 16.5印张

印数:0 001-8 000册

定价:28.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

C Preface to the Chinese Edition

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.

Brian W. Kernighan
Dennis M. Ritchie

中文版序

C PROGRAMMING LANGUAGE

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语言已经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语言指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需改变。这个标准同时也被国际标准化组织（ISO）接受为国际标准，使世界各地的用户团体都受益于这一标准。

标准委员会会考虑到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——具有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的权威译本，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《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华人民共和国的同行更容易地理解C语言。

Brian W. Kernighan

Dennis M. Ritchie

译者序 C PROGRAMMING LANGUAGE

《C程序设计语言》(第2版)与《C++程序设计语言》(第3版)这两部书不仅在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。这主要是基于以下几点理由:

其一,这两部著作自第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++著作的作者都把这两部著作作为书末的基本参考文献。我国也早就翻译出版过这两部著作的第1版,其中,《C程序设计语言》(第1版)早在20年前就在我国翻译出版了,《C++程序设计语言》(第1版)1988年由清华大学出版社出版。

其二,这两部著作的原作者Dennis Ritchie与Bjarne Stroustrup分别是C与C++语言的研制者,这样就保证了在著作中能完整、准确地体现与描述C与C++语言的设计思想、程序设计方法以及各种语言成分的细节与用法,具有原创性与权威性,这很有利于读者把握C与C++语言的精髓。

《C程序设计语言》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来,由此可见该著作内容的稳定性。《C++程序设计语言》的第1版、第2版与第3版分别于1985、1991与1997年出版,据称已拥有超过一百万的读者。

这两部著作虽然经典,但叙述得深入浅出、条理清楚,加之有例证配合,非常通俗易懂。不仅适用于计算机专业人员,而且也适用于非计算机专业人员;不仅适合学术研究,而且也适合于作为C与C++语言的入门著作;不仅适合作为参考书,而且也适合作为教科书。这也许就是这两部著作自第1版问世以来长期畅销不衰的原因之一。

参加这部著作翻译的还有周毓明、陈振强、李慎之、周晓宇、戚晓芳、聂长海、张卫丰、戴桂兰、张颖等同志。最后,我们要向把这两部著作首先引入中国的第1版的译校者们表示衷心的感谢。

由于水平有限,翻译不当之处,恳请读者批评指正。

徐宝文
2000国庆期间于南京

译者简介

C PROGRAMMING LANGUAGE



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，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，负责承担过十多项国家级、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。

自从1978年《C程序设计语言》出版以来，计算机界经历了一场革命。大型计算机变得更大；而个人计算机的能力可以和十年前的主流计算机相媲美。在这段时间中，C语言也在悄悄地改变，并且早已超出了仅仅作为UNIX操作系统的语言的范畴。

C的适用范围的扩大、在这些年中语言的改变和各个组织开发的超出其预定内容的编译器，所有这一切要求对C语言有一个比本书第1版更精确和更新的定义。在1983年，美国国家标准协会（ANSI）成立了一个委员会，它的目标是产生“一个无二义性的、独立于机器的C语言的定义”，同时仍保持其精髓。其结果就是C的ANSI标准。

此标准规范了一些在本书第1版中提示过但没有描述的结构，特别是结构赋值和枚举。它提供了一种新的函数声明的形式，允许在使用中对函数的定义进行交叉检查。它说明了一个标准库和一个完成输入输出、内存管理和字符串操作等类似任务的函数集扩充。此标准明确地说明了原始定义没有指出的一些特性的行为。同时，此标准还明确地说明了语言中的哪些部分依然依赖于机器。

《C程序设计语言》的第2版描述的是ANSI标准定义的C语言。尽管我们已经指出语言中的多种革新，但我们还是决定不用新的形式来写。对于大部分内容并没有多大的差别，最明显的改变是新形式的函数声明和定义。现代编译器已能支持此标准的大部分特性。

我们尽力保持本书第1版的简洁性。C不是一个大型语言，也不需要一本很厚的书来描述。我们改进了对典型特性的阐述，如指针，它是C程序设计的中心。我们对以前的例子进行了提炼，并在几章中增加了新的例子。例如，我们用程序来处理复杂的声明，这些程序将声明转换为单词或反之。像以前一样，所有例子的文本都以机器可读的形式直接经过测试。

附录A是参考手册，不是标准，但我们的目的是希望用较少的篇幅表述标准的要点。它是要对于程序员来说容易理解，而不是提供给编译器实现者的定义——这正是标准承担的角色。附录B是对标准库提供的功能的总结。附录C是对以前版本的变动的小结。

就像我们在第1版序中所说的，“当对C的经验增加时它显得很好用”。经过十几年的实践，我们仍然这么认为。我们希望这本书能帮助你学习并使用好C语言。

非常感谢那些帮助我们完成本书这一版的朋友们。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，他们仔细地阅读了本书。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的有益的建议。Dave Prosser回答了很多关于ANSI标准的细节问题。我们广泛地使用了Bjarne Stroustrup的C++的翻译程序来部分测试我们的程序。Dave Kristol为我们提供了一个ANSI C编译器进行最终测试。Rich Drechsler帮助我们进行了排版。

诚挚地感谢每一个人。

Brian W. Kernighan

Dennis M. Ritchie

第1版序

C PROGRAMMING LANGUAGE

C是一种通用的程序设计语言，它包含了紧凑的表达式、现代控制流和数据结构以及一个丰富的运算符集合。C不是一种“很高级”的语言，也不“大”，它不特定于某一个应用领域。但是C的限制少，通用性强，这使得它比一些被认为功能强大的语言更方便、效率更高。

C最初是由Dennis Ritchie为UNIX系统设计的，并在DEC PDP-11计算机上实现。此操作系统、C编译器和基本上所有的UNIX应用程序（包括本书在编写中所使用到的所有软件）都是用C编写的。同时存在可适用于其他机器的编译器，这些机器包括IBM System/370、Honeywell 6000和Interdata 8/32。但是，C不捆绑于任何特定的机器或系统，很容易用C编写出不用修改就可以运行在所有支持C的机器上的程序。

本书的目的是帮助读者学习用C进行程序设计。它包含一个引导式的介绍，以使新用户能尽可能快地开始；还包含介绍C的每一主要特性的各章和一个参考手册。内容的安排并不仅仅是对规则的描述，而且基于阅读、书写和修改例子的顺序。对大部分例子来说，它们都是完整的可运行的程序，而不仅仅是孤立的程序段。所有例子的文本都以机器可读的形式直接经过测试。除了演示如何有效地使用语言外，我们还尽力在可能的时候例示有效的算法以及良好的程序设计风格和正确设计的原则。

本书不是一本介绍程序设计的手册，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。尽管更多的知识有助于学习，但一个初级程序员应能够阅读本书并学会C语言。

从我们的经验来看，C语言是一种令人愉快的、高效的和多样化的语言，适合于编写各种程序。它容易学习，并且当对它的经验增加时显得很好用。我们希望本书能帮助你很好地使用它。

我们非常高兴收到了来自很多朋友和同事的批评和建议，帮助我们为本书添加了很多内容。特别感谢Mike Bianchi、Jim Blue、Stu Feldman、Doug McIlroy、Bill Roome、Bob Rosin和Larry Rosler仔细地阅读了本书的多次修改稿。也要感谢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在不同阶段提出有益的意见，并感谢Mike Lesk和Joe Ossanna在排版上所给予的宝贵的协助。

Brian W. Kernighan

Dennis M. Ritchie



前

言

C PROGRAMMING LANGUAGE

C是一种通用的程序设计语言。它是为UNIX开发的，并与之紧密相关——UNIX系统和运行于其上的软件都是用C编写的。然而，它不特定于任何一个操作系统或机器。由于它很适合用来编写编译器和操作系统，因而被称为“系统程序设计语言”。尽管如此，它同样被很好地用来编写不同应用领域的主要程序。

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

BCPL和B都是“无类型”的语言。相比而言，C提供了很多数据类型。其基本类型为几种大小的字符、整数和浮点数类型。另外还有从指针、数组、结构和联合产生的派生数据类型的层次。表达式由运算符和操作数形成。任何一个表达式，包括赋值或函数调用表达式，都可以是一个语句。指针提供了独立于机器的地址算术。

C为实现结构良好的程序提供了基本的控制流结构：语句组、判断（if-else）、选择一个可能的路径（switch）、终止测试在顶端进行（while、for）和底端进行（do）的循环和提前跳出循环（break）。

一个函数可以返回基本类型的值、结构、联合和指针。函数可以递归调用。局部变量通常是“自动的”，即每次函数调用时重新创建变量。函数定义可能不是嵌套的，但可以用块结构的方式声明变量。C程序的函数能以单独的源文件形式存在，这些文件可以分开编译。变量对于函数可以是内部的，也可以是外部的，但外部变量只在一个源文件内有效，或者对整个程序是可见的。

编译的预处理阶段对程序文本进行宏替换，包含进其他源文件，并进行条件编译。

C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可以处理大部分计算机可以处理的对象，如字符、数字和地址。这些对象可以用由机器执行的算术和逻辑运算符组合在一起。

C不提供直接处理诸如字符串、集合、列表和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，但不存在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语言没有任何存储分配工具；也不提供堆和无用内存回收。最后，C本身没有提供输入、输出工具，没有READ或WRITE语句，也没有内部文件访问方法。所有这些高级机制必须由显式调用的函数提供。C的大部分实

现已包含了这类函数的标准集合。

类似地，C只提供单线程的控制流：测试、循环、组合和子程序。它不提供多程序设计、并行操作、同步和协同例程。

缺少这些特性可能看上去会降低效率（“你的意思是我必须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大小会有很多益处。由于C相对较小，可以用一个小的篇幅将它描述出来，也可以很快学会。一个程序员有理由期望学会和理解并实际使用整个语言。

很多年来，C的定义就是《C程序设计语言》第1版中的参考手册。在1983年，美国国家标准协会（ANSI）建立了一个委员会以提供一个C的成熟的定义。所得的定义就是1988年完成的ANSI标准，即“ANSI C”。这个标准的大部分特性已由现代编译器所支持。

这个标准基于以前的参考手册。语言有一点小的改变。这个标准的目的之一就是确保现存的程序仍然有效，或者当程序失败时，编译器会产生对新动作的警告信息。

对大部分程序员来说，最重要的改变是新的函数声明和定义的语法。现在一个函数声明可以包含描述参数的信息；为了匹配，函数的定义也做了相应的改变。这些附加的信息使编译器很容易检测到参数不匹配所造成的错误。根据我们的经验，这个扩充对语言非常有用。

语言的其他改变都比较小。一直被广泛使用的结构赋值和枚举现在已成了语言的正式部分。可以进行单精度的浮点运算。已澄清了算术运算，特别是无符号类型的运算的属性。预处理器得到了更详尽的说明。这些改变对大部分程序员的影响都比较小。

这个标准的第二个突出贡献是定义了一个C的函数库。它说明了诸如访问操作系统（如读写文件）、格式化的输入输出、内存分配和字符串操作等的函数。一个标准头文件的集合提供了统一的访问函数和数据类型的声明的方法。那些使用这个库与宿主系统交互的程序保证有兼容的动作。这个库的大部分与UNIX系统的“标准I/O库”相似。这个库已在本书第1版中有所描述，并用在很多其他系统中。同样，大部分程序员看不到有多大的变化。

由于大多数计算机支持C所提供的数据类型和控制结构，因此只需要一个很小的库就可实现自包含的程序。标准库函数总是被显式地调用，因此如果没有必要，可以避免这些函数调用。除了涉及到操作系统的细节，大部分库函数可以用C来编写，并可以移植。

虽然C与大部分计算机的能力相匹配，但是它独立于任何特定的机器结构。只要稍加细心就可以写出可移植的程序，即可以不加改变地运行于很多硬件上。这个标准明

确地提出了可移植性问题，并预设了一个常量的集合来说明运行程序的机器的特性。

C不是一种强类型的语言，但是随着它的发展，其类型检查机制已得到了加强。C的初始定义虽然不赞成将指针和整数混用，但却允许这样使用。现在已不再允许，这个标准要求有正确的声明和显式的强制转换。实际上有一些好的编译器已有这样的要求。在这个方向上，新的函数声明方式是做出的另一个步骤。编译器将对大部分的类型错误产生警告，并不进行不相容的数据类型之间的自动转换。然而，C保留了其初始的设计思想，即程序员知道他们在做什么，只是要求他们明确地表达他们的意图。

像其他语言一样，C也有其不完美的地方。有一些运算符的优先级不合适；语法的一些部分还可以做得更好。然而，对于很多程序设计的应用来说，C已被证明是一个非常高效和表达能力强的语言。

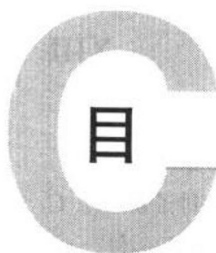
本书安排如下：第1章是对C的中心部分的一个简要介绍。其目的是让读者尽可能快地开始，因为我们深切地相信学习一种新语言的方法是用它写程序。这个介绍假定读者对程序设计的基本元素有一定的了解。在此没有解释计算机、编译，也没有解释诸如 $n=n+1$ 这样的表达式。虽然我们已尽力在可能的地方展示有用的程序设计技术，但是本书的意图并不是介绍数据结构和算法。当不得不做出选择时，我们将专注于语言本身。第2章到第6章更详细地讨论C的不同特性，且比第1章更形式化。其重点是作为完整程序的例子，而不是孤立的程序段。

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、作用域规则和多个源文件等，并涉及到预处理器。第5章讨论指针和地址运算。第6章讨论结构和联合。

第7章讨论标准库，它提供了与操作系统的一个公用接口。这个库由ANSI标准定义，即意味着受到所有支持C的机器的支持，因此使用这个库进行输入、输出和其他操作系统访问的程序可以不加改变地在不同机器上运行。

第8章讨论C程序和UNIX操作系统之间的接口，集中于输入/输出、文件系统和存储分配。尽管这一章的一些部分特定于UNIX系统，但是使用其他系统的程序员仍然会从中找到有用的信息，包括深入了解如何实现标准库的一个版本和对可移植性的一些建议。

附录A包含了一个语言参考手册。C的正式的语法和语义定义是ANSI标准本身。但那个文档首先是给编译器的编写者看的。本书中的参考手册用一种不那么严格的方式更为简洁地介绍了C语言的定义。附录B是对标准库的总结，这同样是为用户而不是实现者准备的。附录C是对初始语言变动的小结。如果有疑问，标准和每个人自己用的编译器将是解释语言的最终权威。附录D提供了本书的索引。



目

录

C PROGRAMMING LANGUAGE

中文版序	
译者序	
序	
第1版序	
前言	
第1章 基本概念	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	13
1.6 数组	15
1.7 函数	17
1.8 变元——按值调用	20
1.9 字符数组	20
1.10 外部变量与作用域	23
第2章 类型、运算符与表达式	27
2.1 变量名	27
2.2 数据类型与大小	27
2.3 常量	28
2.4 说明	30
2.5 算术运算符	31
2.6 关系运算符与逻辑运算符	32
2.7 类型转换	33
2.8 加一与减一运算符	36
2.9 按位运算符	38
2.10 赋值运算符与赋值表达式	39
2.11 条件表达式	41
2.12 运算符优先级与表达式求值次序	42
第3章 控制流	44
3.1 语句与分程序	44
3.2 if-else语句	44
3.3 else-if语句	45
3.4 switch语句	47
3.5 while与for循环语句	48
3.6 do-while循环语句	51
3.7 break语句与continue语句	52
3.8 goto语句与标号	53
第4章 函数与程序结构	55
4.1 函数的基本知识	55
4.2 返回非整数值的函数	58
4.3 外部变量	60
4.4 作用域规则	66
4.5 头文件	67
4.6 静态变量	69
4.7 寄存器变量	70
4.8 分程序结构	70
4.9 初始化	71
4.10 递归	72
4.11 C预处理程序	74
4.11.1 文件包含	74
4.11.2 宏替换	74
4.11.3 条件包含	76
第5章 指针与数组	78

5.1 指针与地址	78	7.8.2 字符类测试和转换函数	141
5.2 指针与函数变元	80	7.8.3 ungetc函数	141
5.3 指针与数组	81	7.8.4 命令执行函数	141
5.4 地址算术运算	84	7.8.5 存储管理函数	141
5.5 字符指针与函数	87	7.8.6 数学函数	142
5.6 指针数组与指向指针的指针	90	7.8.7 随机数发生器函数	142
5.7 多维数组	93	第8章 UNIX系统界面	144
5.8 指针数组的初始化	95	8.1 文件描述符	144
5.9 指针与多维数组	96	8.2 低级I/O——read和write系统调用	145
5.10 命令行变元	96	8.3 open、creat、close和unlink系统 调用	146
5.11 指向函数的指针	100	8.4 随机访问——lseek系统调用	148
5.12 复杂说明	103	8.5 实例——fopen和getc函数的一种 实现方法	149
第6章 结构	108	8.6 实例——目录显示	153
6.1 结构的基本知识	108	8.7 实例——存储分配程序	158
6.2 结构与函数	109	附录A 参考手册	163
6.3 结构数组	112	A.1 引言	163
6.4 结构指针	115	A.2 词法规则	163
6.5 自引用结构	117	A.3 语法符号	166
6.6 查找表	121	A.4 标识符的含义	166
6.7 类型定义	123	A.5 对象和左值	168
6.8 联合	125	A.6 转换	168
6.9 位字段	126	A.7 表达式	171
第7章 输入与输出	128	A.8 说明	180
7.1 标准输入输出	128	A.9 语句	192
7.2 格式化输出——printf函数	130	A.10 外部说明	195
7.3 变长变元表	131	A.11 作用域与连接	198
7.4 格式化输入——scanf函数	133	A.12 预处理	199
7.5 文件访问	135	A.13 语法	204
7.6 错误处理——stderr和exit函数	138	附录B 标准库	212
7.7 行输入输出	139	B.1 输入与输出: <stdio.h>	212
7.8 其他函数	140		
7.8.1 字符串处理函数	140		

B.2 字符类测试: <ctype.h>	219	B.9 信号处理: <signal.h>	226
B.3 字符串函数: <string.h>	220	B.10 日期与时间函数: <time.h>	226
B.4 数学函数: <math.h>	221	B.11 由实现定义的限制: <limits.h> 和<float.h>	228
B.5 实用函数: <stdlib.h>	222	附录C 变更小结	230
B.6 诊断: <assert.h>	225	附录D 索引	233
B.7 变长变元表: <stdarg.h>	225		
B.8 非局部跳转: <setjmp.h>	225		