

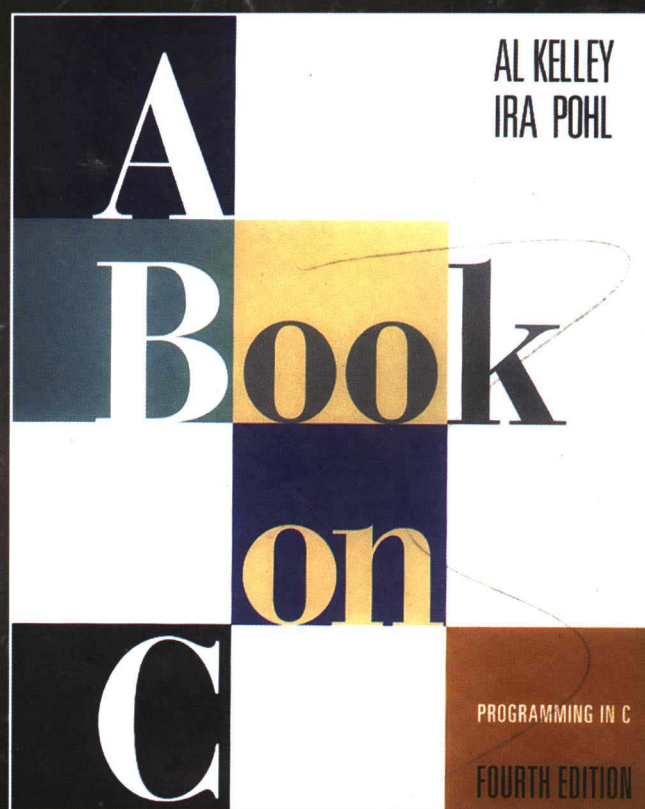


计 算 机 科 学 丛 书

原书第4版

C语言教程

(美) Al Kelley Ira Pohl 著 徐波 译



A Book on C
Programming in C, Fourth Edition



机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

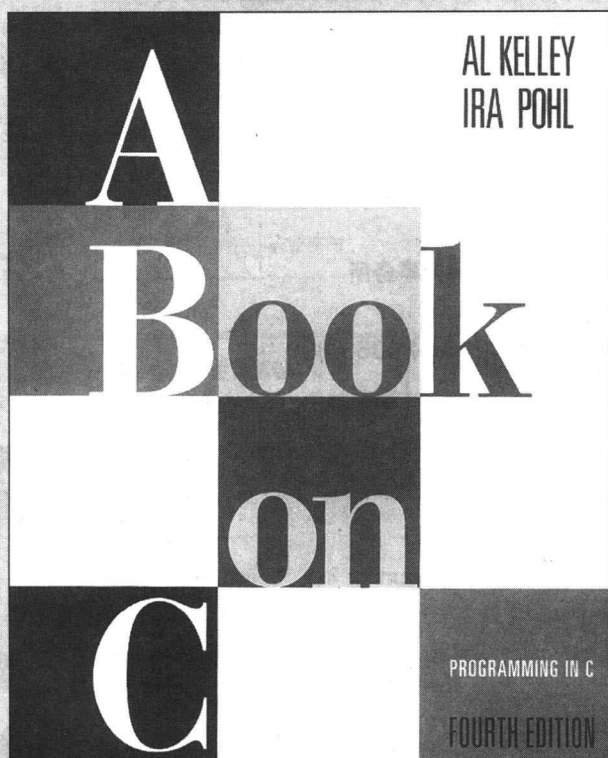
原书第4版

TP312
2183

2007

C语言教程

(美) Al Kelley Ira Pohl 著 徐波 译



A Book on C
Programming in C, Fourth Edition



机械工业出版社
China Machine Press

本书是一本优秀的C程序设计语言教材，完整描述了ANSI C语言及其语法特性，并对C语言的高级特性和应用作了深入阐述，介绍了从C到C++和Java过渡的相关知识。本书的一个鲜明特色就是结合大量示例描述C语言的重要特征，并对很多工作代码给出了逐步的分析，以这种独特的教学方法向读者解释新接触的编程元素及一些惯用法。

本书系统、完整，可作为C语言的参考手册，也非常适合作为学习C语言的入门和高级课程教材。

Simplified Chinese edition copyright © 2006 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *A Book on C: Programming in C, Fourth Edition* (ISBN 0-201-18399-4) by Al Kelley and Ira Pohl, Copyright © 1998.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

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

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2006-6890

图书在版编目（CIP）数据

C语言教程（原书第4版）／（美）凯利（Kelley, A.），（美）波尔（Pohl, I.）著；徐波译．—北京：机械工业出版社，2007.1

（计算机科学丛书）

书名原文：A Book on C: Programming in C, Fourth Edition

ISBN 7-111-20213-9

I. C… II. ①凯…②波…③徐… III. C语言—程序设计—教材 IV. TP312

中国版本图书馆CIP数据核字（2006）第127227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：罗媛媛 刘立卿

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2007年1月第1版第1次印刷

184mm×260mm·27.5印张

定价：45.00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

出版者的话

文艺复兴以降，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取得了垄断性的优势；也正是这样的传统，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，还揭橥了学术的源变，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、从业人员较少的现状下，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图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann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Tanenbaum, Stroustrup, Kernighan, Jim Gray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、研究及度藏。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襄助，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，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，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，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步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步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业

的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程,而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下,读者必将在计算机科学的宫殿中由登堂而入室。

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,但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。教材的出版只是我们的后续服务的起点。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:

电子邮件: hzjsj@hzbook.com

联系电话: (010) 68995264

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元
石教英
张立昂
邵维忠
周克定
郑国梁
高传善
裘宗燕

王 珊
吕 建
李伟琴
陆丽娜
周傲英
施伯乐
梅 宏
戴 葵

冯博琴
孙玉芳
李师贤
陆鑫达
孟小峰
钟玉琢
程 旭

史忠植
吴世忠
李建中
陈向群
岳丽华
唐世渭
程时端

史美林
吴时霖
杨冬青
周伯生
范 明
袁崇义
谢希仁

前言

本书详细介绍了C这种通用的编程语言，既显示了C“简单即是美”的优雅特性，又展现了它“深入底层”的强大功能。本书通过详细探讨许多来自应用领域的交互式运行程序，深入描述了ANSI版本的C语言。本书一步一步地对C语言进行了完整的描述；另外，还提供了许多完整的工作程序。

在适当的时候，我们会讨论传统C和ANSI C的不同之处（传统C仍然被广泛使用）。本书通过大量的例子，对每个重要的语言特性进行阐述。此外，还以表格形式对C的关键信息进行总结，便于读者以后参考。每一章在结束的时候都有总结和练习部分。总结部分对该章所讨论的一些关键概念进行概括，而练习部分则是对正文的论证和延伸。

本书从通用编程语言的角度对C语言进行描述。它可以作为读者学习编程语言的入门课程；也可以作为进一步学习C语言的高级课程；还可以与比较式编程语言、计算语言学、数据结构、数据库系统、分形几何、图形学、数值分析、操作系统、编程方法学以及科学应用程序等课程一起学习。C适用于编写上述领域的应用程序，本书解释了编写这些应用程序所需要的所有特性。本书也可以作为数据结构课程的教材，因为它讨论了枚举类型、联合、自引用等结构以及不规则数组等高级数据结构特性。作为与UNIX或Windows 95/NT相关的操作系统课程，本书探讨了有关文件结构和系统函数的一些知识，使程序员能对原有的系统函数库进行添加，并理解与操作系统相关的底层C代码。对于应用程序和科学应用编程，本书介绍了如何编写示例函数库。本书所提供的工作代码还涉及统计、求方程式的根、排序、文本操纵、文件处理和游戏等内容。

新增的Java内容

在第14章中，我们讨论了C程序员可以很自然和很方便地用Java进行编程的原因。Java是一种适合在Internet上运行的语言，它借鉴了C和C++的许多思想，并按照一种独立于机器和系统的方式运行。这使它特别适合在Internet上运行，例如编写由浏览器所使用的在Web页面上运行的Applet。由于Java是对C和C++的扩展，因此C程序员很乐意学习Java语言。

完整的ANSI C语言

计算机专业人员可以在本书中找到C语言的所有主题，包括枚举类型、列表处理和操作系统接口。第1章对这门语言进行了总览性描述。在阅读了这一章之后，有编程基础的读者就可以编写C代码了。由于本书各章的内容相对独立，因此编程基础较扎实的读者可以根据需要阅读特定的章节。第11章详细讨论了C语言与操作系统的连接，这些内容可以帮助系统程序员在MS-DOS和UNIX下用C进行编程。

交互性环境

本书充分利用了现代计算机的交互性环境。我们在本书中始终鼓励读者进行试验。键盘

和屏幕是常规的输入/输出设备，但本书也对其他的输入输出途径进行了解释。因此，本书既适用于小型的家用和商用计算机用户，也适用于大型交互性系统的用户。我们假定读者可以使用交互性的ANSI C系统。在编写本书期间，我们使用了一些不同的C系统：在IBM兼容的奔腾机器上运行的各种Borland和Microsoft编译器，GNU的gcc编译器，运行于DEC、SGI、Sun等工作站的本地编译器以及运行于San Diego的Cray大型机上的C编译器。

工作代码

我们通过示例程序、解释和语法来描述C语言。本书使用了大量的工作代码，通过许多小而有用的例子对一些重要的概念进行描述。之所以使用小型代码，是为了更容易让读者理解。之所以说这些小例子非常有用，是因为总是可以将程序分解为层次式的基本构件。本书所描述的程序和函数可以在实际的系统中使用。作者所信奉的哲学是读者应该不断进行试验，并从中享受乐趣。

分析

本书采用“分析”段落，对所提供的许多程序和函数进行详细的分析。这种分析方法是作者最初于1984年创造的，这是一种独一无二的教学方法，用于说明工作代码的一些关键特性。这种分析方法类似于对代码进行结构化的走查，目的是向读者解释新接触的编程元素以及工作代码中所出现的惯用法。

灵活的组织

本书的组织形式非常灵活。第1章分为两个部分：第一部分解释了实现交互性输入/输出所需要的关键编程技巧，这些内容是必须理解的；第1章的第二部分是对整个语言的总览性介绍。对于那些已经熟悉其他类似编程语言的读者，在阅读了这些内容之后，很容易了解C语言的概貌。对于初学编程的读者，第二部分可以推迟阅读。**警告：**初学者应该推迟第1章第二部分的学习。

第2章在词法层次上对C语法进行描述，讨论了C的语法规则。该章选择性地使用了一些语法规则来显示C语言的结构。对于第2章所描述的Backus-Naur-Form（BNF巴克斯-诺尔形式）记法，读者可以选择完全将其略过，这并不会对学习的连续性造成任何影响。本书使用了BNF风格的语法描述，使读者可以了解标准的编程语言描述形式。另外，本书通过示例程序和解释对语言的成分进行详细的描述。

参考作用

本书可以作为C语言的参考手册。本书提供了许多表格，简洁地总结了C语言的一些关键概念。附录A描述了完整的ANSI C标准函数库以及相关的头文件。附录A的内容是根据标准头文件组织的，例如`ctype.h`、`stdio.h`和`string.h`等。在适当的时候，我们会提供示例代码说明某个特定结构或特定函数的用法。

在附录B中，我们提供了C语言的完整语法。在附录C中，我们讨论了ANSI C和传统C的一些主要区别。

完整的语法特性

第3章到第10章对C语言的各个特性进行了完整的介绍。其中包括了许多高级话题，读者在

第一遍阅读时可以将它们省略，这样并不会对接下来的学习产生影响。例如，读者在第一遍阅读本书时可以忽略枚举类型的介绍。本书强调了像字长和浮点表示形式等依赖于机器的特性，但初学者可以忽略这些细节。

预处理器

第8章完整地讲述了预处理器，它用于对C语言的功能和概念进行扩展。宏可以用于生成内联代码，可以代替函数调用。使用宏可以减少程序的执行时间。该章对预处理器进行了详细的描述，包括ANSI C委员会所增加的一些新特性。在传统C中，预处理器在不同的编译器中可能具有显著的差别。在ANSI C中，预处理器的功能完全是预先指定的。

递归和列表处理

第5章对递归进行了详细讨论。递归这个概念对于初学者而言常常显得颇为神秘。第8章在讨论快速排序算法时以及第10章在讨论基本的列表处理技巧时再次说明了递归的用法。要完整地理解列表处理技巧，读者需要学习高级编程和数据结构课程。

与操作系统连接

第11章讨论了C语言和操作系统的连接。在该章中，我们解释了如何进行文件处理，并详细介绍了标准函数库中的各个输入/输出函数。我们还解释了如何在C程序中执行系统命令，如何设置文件访问权限以及如何使用环境变量。我们提供了一些例子，说明了性能评估程序、函数库管理工具以及*make*工具的用法。

高级应用

在第12章中，我们讨论了C语言的一些高级应用。我们讨论了诸如并发进程的创建、叠加进程、进程间通信以及信号等话题，并提供了一些工作代码。另外，我们还讨论了向量和矩阵的动态内存分配，这些知识工程师和科学家会用到。读者可以根据需要选择性地使用这些高级技巧，它们可以作为高级编程实践的基础。本书也可以作为以C为实现语言的高级计算机科学课程的辅导书。

表格、总结和练习

在本书中，我们用许多表格对C语言的一些关键概念进行总结。这些表格可以帮助读者加深对C语言的理解。例如，C具有大量的操作符，并且几乎允许操作符进行各种类型的混合使用。理解这些操作符的优先级和结合性是非常重要的，它有助于理解各种混合表达式的求值顺序。本书中的很多表格就是对各种混合表达式的求值顺序进行分析。作为参考手册，这些表格和代码很容易寻找。

练习部分是对语言的基本元素进行测试，并讨论了一些高级的以及因系统而异的特性。许多练习的目的是为了解决问题，还有些练习则是测试读者对C的语法或语义的理解。有些练习所讨论的主题是对正文的补充，有些读者可能会对它们产生兴趣。本书的练习提供了难易程度各异的问题，适合各个层次的读者。

目 录

出版者的话
专家指导委员会
前言

第0章 从零开始	1
0.1 为什么要用C	1
0.2 ANSI C标准	2
0.3 从C到C++	2
0.4 从C和C++到Java	3
第1章 C语言概述	5
1.1 编程和预备知识	5
1.2 程序输出	5
1.3 变量、表达式和赋值	8
1.4 使用#define和#include	10
1.5 使用printf()和scanf()	12
1.6 控制流	14
1.7 函数	19
1.8 数组、字符串和指针	23
1.8.1 数组	23
1.8.2 字符串	25
1.8.3 指针	27
1.9 文件	29
1.10 与操作系统有关的内容	33
1.10.1 编写和运行C程序	33
1.10.2 中断程序	35
1.10.3 输入文件尾标志	35
1.10.4 输入和输出的重定向	35
1.11 总结	36
1.12 练习	37
第2章 词法元素、操作符和C系统	43
2.1 字符和词法元素	43
2.2 语法规则	45
2.3 注释	46
2.4 关键字	47

2.5 标识符	48
2.6 常量	49
2.7 字符串常量	49
2.8 操作符和标点符号	50
2.9 操作符的优先级和结合性	51
2.10 增值操作符和减值操作符	52
2.11 赋值操作符	53
2.12 例子: 计算2的乘方	54
2.13 C系统	55
2.13.1 预处理器	55
2.13.2 标准函数库	56
2.14 总结	58
2.15 练习	59
第3章 基本数据类型	65
3.1 声明、表达式和赋值	65
3.2 基本数据类型	66
3.3 字符和char数据类型	67
3.4 int数据类型	70
3.5 整数类型short、long和unsigned	71
3.6 浮点类型	72
3.7 typedef的用法	74
3.8 sizeof操作符	74
3.9 使用getchar()和putchar()	75
3.10 数学函数	77
3.10.1 使用abs()和fabs()	78
3.10.2 UNIX和数学函数库	79
3.11 隐式类型转换和强制类型转换	79
3.11.1 整型提升	79
3.11.2 寻常算术转换	79
3.11.3 强制类型转换	81
3.12 十六进制和八进制常量	81
3.13 总结	83
3.14 练习	84

第4章 控制流	89	5.13 默认初始化	134
4.1 关系操作符、相等操作符和逻辑操作符	89	5.14 递归	135
4.2 关系操作符和表达式	90	5.15 例子: 汉诺塔	138
4.3 相等操作符和表达式	92	5.16 总结	141
4.4 逻辑操作符和表达式	93	5.17 练习	142
4.5 复合语句	95	第6章 数组、指针和字符串	149
4.6 表达式和空语句	95	6.1 一维数组	149
4.7 if和if-else语句	96	6.1.1 初始化	150
4.8 while语句	98	6.1.2 下标	150
4.9 for语句	100	6.2 指针	151
4.10 例子: 布尔变量	102	6.3 传引用调用	153
4.11 逗号操作符	102	6.4 数组和指针之间的关系	154
4.12 do语句	103	6.5 指针运算和元素的大小	155
4.13 例子: 斐波那契数	105	6.6 数组作为函数的实参	156
4.14 goto语句	107	6.7 例子: 冒泡排序	156
4.15 break和continue语句	108	6.8 用calloc()和malloc()进行动态内存分配	157
4.16 switch语句	108	6.9 例子: 归并和归并排序	160
4.17 条件操作符	109	6.10 字符串	164
4.18 总结	110	6.11 标准函数库中的字符串处理函数	165
4.19 练习	111	6.12 多维数组	168
第5章 函数	119	6.12.1 二维数组	168
5.1 函数定义	119	6.12.2 存储映射函数	169
5.2 return语句	121	6.12.3 形式参数声明	169
5.3 函数原型	121	6.12.4 三维数组	170
5.4 例子: 创建乘方表	122	6.12.5 初始化	170
5.5 从编译器的角度观察函数原型	123	6.12.6 使用typedef	171
5.6 函数定义顺序的另一种风格	124	6.13 指针数组	172
5.7 函数调用和传值调用	125	6.14 main()函数的参数	176
5.8 开发大型程序	126	6.15 不规则数组	176
5.9 使用断言	128	6.16 函数作为参数	178
5.10 作用域规则	129	6.17 例子: 使用二分法寻找函数的根	180
5.10.1 平行和嵌套代码块	130	6.18 函数指针数组	183
5.10.2 以调试为目的使用代码块	130	6.19 类型限定符const和volatile	186
5.11 存储类型	130	6.20 总结	187
5.11.1 auto存储类型	130	6.21 练习	188
5.11.2 extern存储类型	131	第7章 位操作符和枚举类型	201
5.11.3 register存储类型	132	7.1 位操作符和表达式	201
5.11.4 static存储类型	133	7.1.1 按位求反	202
5.12 静态外部变量	133	7.1.2 补码	202

7.1.3 位逻辑操作符	203	9.12 练习	265
7.1.4 左移位和右移位操作符	203	第10章 结构和列表处理	271
7.2 掩码	204	10.1 自引用的结构	271
7.3 软件工具: 打印int值的二进制形式	205	10.2 线性链表	272
7.4 包装和解包	206	10.3 链表操作	273
7.5 枚举类型	209	10.4 一些链表处理函数	275
7.6 例子: “石头、剪刀、布” 游戏	211	10.4.1 插入	277
7.7 总结	215	10.4.2 删除	277
7.8 练习	216	10.5 堆栈	278
第8章 预处理器	221	10.6 例子: 波兰记法和堆栈求值	280
8.1 #include的使用	221	10.7 队列	285
8.2 使用#define	221	10.8 二叉树	287
8.3 带参数的宏	222	10.8.1 二叉树的遍历	288
8.4 stddef.h中的类型定义和宏	224	10.8.2 创建树	290
8.5 例子: 用qsort()进行排序	225	10.9 普通的树	290
8.6 例子: 带参数的宏	228	10.9.1 遍历	292
8.7 stdio.h和ctype.h中的宏	231	10.9.2 calloc()的用法以及树的创建	292
8.8 条件编译	232	10.10 总结	293
8.9 预定义的宏	234	10.11 练习	294
8.10 “#” 和 “##” 操作符	234	第11章 输入/输出和操作系统	299
8.11 assert()宏	235	11.1 输出函数printf()	299
8.12 使用#error和#pragma	235	11.2 输入函数scanf()	302
8.13 行号	236	11.3 fprintf()、fscanf()、sprintf() 和sscanf()函数	305
8.14 对应的函数	236	11.4 fopen()和fclose()函数	306
8.15 例子: 快速排序	236	11.5 例子: 对文件进行空间加倍	307
8.16 总结	238	11.6 使用临时文件和优雅函数	309
8.17 练习	239	11.7 随机访问文件	311
第9章 结构和联合	247	11.8 文件描述符输入/输出	312
9.1 结构	247	11.9 文件访问权限	313
9.2 访问结构成员	249	11.10 在C程序内部执行命令	314
9.3 操作符的优先级和结合性的总结	251	11.11 在C程序内部使用管道	315
9.4 在函数中使用结构	252	11.12 环境变量	316
9.5 结构的初始化	253	11.13 C编译器	316
9.6 例子: 玩扑克牌	254	11.14 使用性能评估程序	318
9.7 联合	257	11.15 函数库	319
9.8 位字段	259	11.16 对C代码进行计时	320
9.9 例子: 访问位和字节	260	11.17 使用make	323
9.10 ADT堆栈	261	11.18 使用touch	327
9.11 总结	264		

11.19 其他有用的工具	327	13.8 多态	368
11.20 总结	328	13.9 模板	370
11.21 练习	329	13.10 C++的异常	371
第12章 高级应用	337	13.11 面向对象编程的优点	371
12.1 用fork()创建并发进程	337	13.12 总结	372
12.2 进程的叠加: exec...()函数族系	338	13.13 练习	373
12.3 使用pipe()实现进程间的通信	340	第14章 从C到Java	377
12.4 信号	342	14.1 输出	377
12.5 例子: 哲学家用餐问题	344	14.2 变量和类型	378
12.6 矩阵的动态分配	346	14.3 类和抽象数据类型	379
12.6.1 为什么二维数组无法满足要求	346	14.4 重载	380
12.6.2 用指针数组创建矩阵	347	14.5 类的创建和销毁	380
12.6.3 调整下标范围	348	14.6 面向对象编程和继承	381
12.6.4 一次分配所有内存	350	14.7 多态和重写方法	381
12.7 返回状态	351	14.8 Applet	382
12.8 总结	354	14.9 Java的异常	383
12.9 练习	355	14.10 Java和OOP的优势	384
第13章 从C到C++	359	14.11 总结	384
13.1 输出	359	14.12 练习	385
13.2 输入	360	附录A 标准函数库	387
13.3 函数	362	附录B C的语法	413
13.4 类和抽象数据类型	363	附录C ANSI C与传统C的比较	419
13.5 重载	364	附录D ASCII字符码	424
13.6 构造函数和析构函数	366	附录E 操作符的优先级和结合性	425
13.7 面向对象编程和继承	367		

第0章 从零开始

在C编程语言中，0是最为自然的起点。C是从0开始计数的；C用0表示假，用非0表示真；C数组的下标从0开始；C的字符串用0作为结束标志；C的指针用0表示null值；C的外部变量和静态变量在默认情况下被初始化为0。本书解释了C的所有概念，并带你领略用C进行编程的快乐。

C是一种通用的编程语言，最初是在1972年由贝尔实验室的Dennis Ritchie所设计，并在一台PDP-11机器上实现。一开始，它是作为UNIX操作系统的系统语言使用的。早在1970年，UNIX的开发者Ken Thompson曾经用汇编语言和一种称为B的语言编写了早期版本的UNIX。C的发明正是为了克服B的一些限制。

B是一种基于BCPL的编程语言，后者是Martin Richards于1967年所开发的一种无类型系统编程语言。它的基本数据类型就是机器的字，并大量使用了指针和地址运算。这恰好与结构化编程的精神背道而驰。结构化编程的特点就是使用强类型语言，例如与ALGOL类似的语言。C是从B和BCPL发展而来，并吸收了它们的精髓。

在上世纪80年代早期，最初的C语言在增加了void类型、枚举类型以及其他一些改进之后，发展成为我们所熟悉的传统C。上世纪80年代晚期，美国国家标准学会（ANSI）X3J11委员会起草了C的标准，使C进一步发展成为ANSI C（或标准C）。这个委员会在传统C的基础之上增加了void*类型、函数原型和一种新的函数定义语法，并增强了预处理器的功能，使语言的定义更为精确。今天，ANSI C是一种成熟的、通用的编程语言，可以在许多机器和操作系统中使用。它是当今业界所使用的主流编程语言之一，到处都可以看到它的踪影。另外，ANSI C还是C++的基础（C++是一种增加了面向对象特性的编程语言）。本书描述ANSI版本的C语言，同时还对C++和Java的一些话题进行了介绍。

0.1 为什么要用C

C是一种小型语言。在编程中，小即是美。C的关键字要少于Pascal（在这种语言中称为保留字），但它却是一种功能更为强大的语言。C的威力来自于它包含了正确的控制结构和数据类型，并近乎不设限制地允许程序员做他们想做的事情。C的所有威力都是由它的最基本功能拼建而成的。

C是UNIX的本地语言，而UNIX是工作站、服务器和大型机上最主要的交互性操作系统。另外，C也是个人电脑的标准开发语言，MS-DOS和OS/2的许多代码就是用C编写的。许多窗口工具包、数据库程序、图形函数库以及其他类型的大型应用程序都是用C编写的。

C具有可移植性。在一种机器上所编写的代码能够很方便地移植到另一种机器上。C为程序员提供了一个标准的函数库，它们可以在所有类型的机器上使用。另外，C还提供了内置的预处理指令，能够帮助程序员隔离一些依赖于系统的代码。

C非常简洁。C提供了一组功能强大的操作符，有些操作符允许程序员在二进制位的层次上对机器进行访问。许多机器的机器语言都存在和C的自增操作符“++”直接类似的指令，

因此这种操作具有非常高的效率。程序员可以在1条语句或1个表达式中组合使用间接访问和地址运算,实现其他语言需要多条语句才能实现的功能。对于许多程序员而言,这种方法既优雅又高效。软件生产率调查表明,程序员平均每天所编写的有效代码数量并不多。因此,简洁的语言可以从根本上提高程序员的效率。

C是模块化的。C只支持一种风格的函数,即外部函数,函数的参数是通过传值方式传递的。C不允许函数的嵌套,它允许程序员通过使用static存储类型实现有限形式的私有性(即限制在文件内部)。这些特性以及操作系统所提供的工具,能够稳定地支持用户定义的函数库和模块化编程。

C是C++和Java的基础。这意味着C程序员平常所使用的结构和方法也可以由C++和Java程序员所使用。因此,学习C也是学习C++或Java的第一步。

C在大多数机器上都具有非常高的效率。由于C的有些结构非常贴近于机器,因此C可以用一种对机器的体系结构而言非常自然的方式来实现。由于这个缘故,经过编译的C代码往往具有极高的效率。当然,程序员必须注意那些依赖于机器的代码。

C也不是没有缺点。它的语法非常复杂,并且它不会自动执行数组的边界检查。有些符号(例如*和=)在C中具有多种不同的含义。例如,一个常见的编程错误就是误用操作符“=”来表示操作符“==”。然而,C是一种优雅的语言,它不会限制程序员对机器的访问。与它的优秀特性相比,C的这些缺点是完全可以接受的。

C的吸引力来自于它所提供的功能强大的操作符以及它不受束缚的本性。C程序员需要努力实现功能的模块化和代码的最简化,并需要进行大量的练习和交互。事实上,练习和交互一直贯穿于本书之中。

0.2 ANSI C标准

ANSI这个缩写代表“美国国家标准学会”(American National Standards Institute)。这个组织致力于为各种系统(包括编程语言)制订标准。ANSI的X3J11委员会专门负责制订C编程语言的标准。上世纪80年代晚期,这个委员会创立了一份标准草案,这就是大家所熟知的ANSI C(或标准C)。1990年,这个委员会完成了它的使命,国际标准化组织(ISO)也批准了ANSI C作为C的标准。因此,ANSI C(或称ANSI/ISO C)便成为广受承认的国际标准。

这个标准指定了用C所编写的程序的形式,并确定了如何对C程序进行解释。这个标准的目的是提高C程序的可移植性、可靠性和可维护性,并保证C程序在各种机器上执行时都能达到非常高的效率。当前,几乎所有的C编译器都遵循ANSI C标准。

0.3 从C到C++

当前,C的应用遍布全球。PC、工作站和大型机上都可以看到它的身影。同时,计算机和操作系统也在不断地发展和演化。对C进行扩展或者限制,让它适用于某个特定的领域显然与C的精神背道而驰。

尽管C语言本身没有进行扩展,但它常常作为其他更高级或更专门的编程语言的核心。Concurrent C通过增加并发元素对C进行了扩展,Objective C通过增加类似Smalltalk风格的对象对C进行了扩展。超级计算机上所使用的有些形式的C则充分利用了这些机器的并行能力。

在这些扩展中,最为重要的是C++,这是一种广为使用的面向对象编程语言。由于它是C

的扩展，因此程序员可以在大型软件项目中同时使用C和C++。C程序员也乐于学习C++语言（参见第13章）。

0.4 从C和C++到Java

Java适用于编写在Internet上运行的程序。它允许程序员编写安全、可移植的程序，这些程序可以从Internet下载并在本地机器上运行。这种新的编程语言借鉴于C和C++的一些思想，并致力于使程序以一种独立于机器和系统的方式运行。它的语义是根据虚拟机而设计的，这意味着Java程序在本质上就是可移植的，在各种不同的系统（例如在PC上运行的Windows 95系统以及在工作站上运行的各种UNIX系统）中能够保持一致的行为。

Java常常用于编写Web上由浏览器所使用的Applet。一般情况下，Applet用于提供代码的图形用户接口。由于Java是C和C++的扩展，因此C程序员也乐于学习Java（参见第14章）。

