

标准 C 语言

实用全书

C Unleashed

[美] Richard Heathfield 等著
Lawrence Kirby

张晓晖 张晓昕 等译

专家的洞察力

本书对 C 语言的复杂性和核心特征进行了详细的论述，包括：

- 优化和性能
- 处理 I/O 流
- 管理内存
- 调试和诊断
- 树和高级数据结构
- 矩阵运算
- 数字信号处理
- 遗传算法
- 自然语言处理技术
- 加密

“实用全书”系列丛书

标准 C 语言实用全书

C Unleashed

[美] Richard Heathfield Lawrence Kirby 等著

张晓晖 张晓昕 等译

电子工业出版社
Publishing House of Electronics Industry
北京·Beijing

内 容 提 要

本书详细叙述了C语言不同层次的知识。全书共分三个部分：C语言基础知识、数据结构和高级专题。在第一部分C语言基础知识中，考察了C语言编程中一些显而易见但又常常被忽略的领域。除此之外，这一部分还包含文件处理、调试、内存管理、仿真和递归等项目的内容。在第二部分数据结构中，讲述了数组、链表、堆栈、队列、双端队列、多叉树（一种特殊的树）、稀疏矩阵和图，并提供了相应的函数库。另外本部分还另辟一章专门讲述排序。在第三部分高级专题中，讨论了如何在算术上处理矩阵，了解了数字信号处理系统，以及其利用ANSI C编写CGI程序进行互联网编程等高级实用专题。

本书内容丰富、叙述清晰、循序渐进，同时每章自成体系。本书主要面向已有初级C语言编程经验的读者，并可帮助读者由中级C程序员进阶为高级C程序员。同时本书以知识单元为基本构件，便于拆卸和重组，可适合不同层次的教学需要。既可作为大专院校计算机专业的教材，也可作为从事计算机工作的科技人员的自学参考书。

Authorized translation from the English language edition published by Sams Publishing, Copyright © 2000. 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 the Publisher.

Simplified Chinese language edition published by Publishing House of Electronics Industry, Copyright © 2001.

本书中文简体版专有翻译出版版权由 Pearson 教育集团所属的 Sams Publishing 授予电子工业出版社。其原文版权及中文翻译出版版权受法律保护。未经许可，不得以任何形式或手段复制或抄袭本书内容。

图书在版编目(CIP)数据

标准C语言实用全书/(美)希思菲尔德(Heathfield, R.)著;张晓晖等译. —北京:电子工业出版社, 2001.6

(“实用全书”系列丛书)

书名原文: C Unleashed

ISBN 7-5053-6751-X

I. 标… II. ①希… ②张… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2001)第033167号

丛 书 名: “实用全书”系列丛书

书 名: 标准C语言实用全书

原 书 名: C Unleashed

著 者: [美] Richard Heathfield Lawrence Kirby 等

译 者: 张晓晖 张晓昕 等

责任编辑: 赵宏英

排版制作: 电子工业出版社计算机排版室监制

印 刷 者: 北京天竺颖华印刷厂

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 58 字数: 1448千字

版 次: 2001年6月第1版 2001年6月第1次印刷

书 号: ISBN 7-5053-6751-X
TP·3781

定 价: 88.00元(含光盘一张)

版权贸易合同登记号 图字: 01-2000-4285

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。若书店售缺,请与本社发行部联系调换。电话 68279077

译 者 序

C 语言是目前软件开发者使用的最主要的编程语言，这也是它得到广泛传播并且成为标准的原因。现在也有新的编程语言出现，但这些语言仍在演化中，C 语言仍然是进行强大的、可移植编程的首选语言。市面上有许多介绍 C 语言的书，可谓“人人自握龙蛇之珠”。但是本书是一本将 C 语言以全面而且实用的方式介绍给读者的书籍。

本书说明了用 C 语言进行编程不仅仅需要在程序中使用正确的用法，而且编程的风格、程序的调试在编写程序过程中占有相当大的分量，所以本书不仅仅向读者介绍编程的机制，同时也告诉读者如何创建易于阅读、调试和更改的程序。此外，本书包含了许多人从未遇到过或从未应用于自己程序中的技术和技巧。现代计算机发展快速，功能更加强大，但是，计算机并不是快得足以掩盖程序中的不足。因此“经典”的数据结构和巧妙的算法仍然大有所为。本书介绍了从中级到高级 C 语言编程各领域中的专家经验，致力于通过分享专家的技巧和窍门使读者成为更好的 C 语言程序员。如果读者的兴趣在于使 C 程序运行得更快、可移植性更好，或希望采用 C 语言编制现代、快速和有效的算法，那么，阅读本书是大有裨益的。

本书几乎完全是在讨论 ANSI C。这是因为通过坚持使用标准 C，可以在最大程度上使代码具有可移植性。本书并不是只面向 Windows C 程序员、DOS C 程序员或 UNIX C 程序员，而是面向每个人的。本书的大部分程序都采用 ANSI C（确切地说是 ANSI/ISO C）编写，因此，它面向所有的平台。在本书中仅有极少数源代码用于特殊平台。

本书做到了理论与实践的完美结合，循序渐进、由浅入深地给出了大量的实例。同时本书取材新颖，并给出了大量源代码，是一本不可多得的学习 C 语言精华的好书和 C 语言程序员案头必备的参考，同时其他领域的研究人员也能从本书中得到有益的启发。

本书的翻译是集体劳动的结晶。主要翻译工作由张晓晖、张晓昕和王艳斌完成，由王辉先生校稿。参与本书翻译及文字录入等工作的人员还有：刘宇键、杨龙、戴伟、李想、韩庆军、李莉、兰朝利、李铁胜、刘建军、王子文、陈海燕、张君等。

因译者水平有限，错误之处在所难免，敬请读者和同行批评指正。

译 者

关于作者

Richard Heathfield 是 Eton 计算机系统有限公司的软件开发人员。他曾经为几家保险公司、卫生部门、银行和航空公司编写程序。目前，他与他的妻子和三个可爱的孩子住在 Northamptonshire (英国) 的 Potterspury 附近。Richard 在 Yahoo! 的聊天室消磨了大量时间。他目前的兴趣是 TCP/IP 编程、自动 C 代码生成和 C99。他的业余爱好是演奏键盘乐器和电吉他。可以通过以下网址与 Richard 联系：<http://users.powernet.co.uk/eton>。

Lawrence Kirby 毕业于英国剑桥大学，有十几年的 C 语言编程经验。他是专门从事通信和金融市场信息工作的 Netactive 系统有限公司的创始人之一。他曾在英国电信任职数年，从事网络服务和信息系统方面的工作。他非常乐于担任 Usenet 新闻组如 comp.lang.c 的常任撰稿人，并且认为帮助他人解决 C 语言的问题以及与其他程序员（从新手到 C 语言标准委员会成员）分享 C 语言编程经验都是十分有益的。

Dann Corbit 从华盛顿大学获得数值分析专业的学位，并从 1976 年起就担任计算机程序员。与计算机有关的部分工作清单可以参阅：ftp://38.168.214.175/pub/C.A.P.%20Biographies/DannCorbit.htm#_Toc441048186。

Dann 在 20 世纪 80 年代中期开始采用 C 语言编程，并在华盛顿 Bremerton 的奥林匹克学院教授 C 程序设计语言。他曾经为 Dr.Dobb's Journal 杂志撰稿，并与他人合作发表关于 Bowhead Whale 的科学论文。他是国际象棋分析计划（该计划用计算机程序分析棋盘位置并把结果编入数据库中）的创始人和管理者。象棋计划常见问题见：<ftp://38.168.214.175/pub/Chess%20Analysis%20Project%20FAQ.htm>。

作为 comp.lang.c 的常任撰稿人，Dann 的电子文稿常出现在这个新闻组中。他对数学及科学一直抱有浓厚的兴趣，他的项目——固溶体分析获得中哥伦比亚科学博览会的第一名，并进入国际科学博览会的决赛。

Chad Dixon 在过去 7 年一直担任网络工程师，从事着计算领域中范围广泛的技术性工作。他为客户解决的问题范围甚广，从客户机/服务器应用开发和数据库到网络监测系统。他曾为 NASA、Lyondell、Olin、Arco Chemical 和英国石油这样的公司或组织工作过，这使他在跨平台网络应用开发方面富有相当的经验。Chad 与他可爱的妻子住在德克萨斯的休斯顿。可以通过 <http://www.loopy.org> 与他联系。

William Fishburne 于 1989 年毕业于 Maryland 大学，获计算机科学的学士学位。他目前是 Alphatech 公司的高级顾问。William 也执教于 Maryland 的 Prince George's Community 学院，同时是 C 与 C++ 用户协会 (ACCU) 的成员。

William 自毕业起就采用 C 语言编程，并且为不同的领域工作，如卫星通信、文件成像和金融计算系统。他曾经使用过各种各样的 UNIX 平台，并较早把 Microsoft Windows NT 引入多处理器平台中。William 为流行编程杂志如 Dr.Dobb's Journal 撰稿。另外，他也参加了一些志愿者行动，如 Gutenberg 计划（该计划收集不再有版权的书籍，并免费以电子版的形式发送）。

可以用 wfishburne@atnet.com 与 William 联系，要了解 Gutenberg 计划可以访问：<http://www.gutenberg.net>。

Scott Fluhner 自 1979 年起用 C 语言编程。他目前是 Cisco Systems 的开发工程师。他的其他兴趣是科幻小说和密码学。

Sam Hobbs 在 1966 年就开始从事计算机工作，那时他使用的还是 25 位计算机和磁鼓存储器，只能使用机器语言。从那时开始，他先后使用过 FORTRAN、Basic、FOCAL、C、C++ 以及 AWK 和 Perl，还有一些数据库语言。他从 1985 年开始使用 C 语言。他主要在电力工业（特别是核电方面）的技术、监督和管理方面富有经验。目前他担任核电工业的工程和管理顾问。

Ian Kelly 是计算机系统顾问，他在 1963 年就开始专业编程。他从事的工作先后有：为英国的电力公司提供编程技术帮助，设计和编写操作系统（ADAM），构造了许多编译器，并为欧洲委员会提供软件移植性的报告。他拥有数学学士学位，掌握并应用了不同环境下的 20 多种计算机语言。目前，他在发展混合语言系统方面提供指导。

由于（错误地）相信法语和 Cobol 一样容易，在 20 世纪 70 年代末，他开始对机器翻译（MT）感兴趣，在这方面出版了两本书，发表了数篇论文，并且他在 BCS 中担任了 15 年的 MT 方面的专家组主席。

Jack Klein 从 1980 年起就进行桌上电脑的编程，他主要从事嵌入系统的设计和编程。他曾经为使用不同的 8、16 和 32 位处理器的产品编程，最初采用汇编语言，从 1983 年起，采用 C 语言。

Jack 专攻过程控制、动作控制和通信，他设计和编程的嵌入系统用于工业、医药、消费者和办公设备。

Jack 目前是 Siemens Medical Systems 核医学小组的高级首席工程师，从事嵌入系统线路设计、动作控制、患者安全性以及为医学成像设备提供成像软件的工作。

Michael Lee 从 1979 年开始编程，在 PL/1 和 C 语言方面具有多年的经验。他的工作主要是用户数据库系统开发。最近的工作是把 SQL 界面接到由 FORTRAN 实现的数据库上。

他最出色的工作可能是查询优化器，这个优化器代码分析收入数据库查询，并把它转换为一种等价查询，后者更好地利用数据库的链接和索引，因而更有效。其他工作包括一个内容广泛的 Web 主页，其内容是优化技巧、技术以及为 Usenet 新闻组生成前 10 个目录的软件。

在业余时间，Mike 喜欢骑山地自行车。

Ben Pfaff 是密歇根州立大学电气工程系的学生。他希望能在毕业后从事数字集成电路的设计工作。Ben 有 14 年的使用计算机的历史，有 10 年的 C 语言编程经验。除计算机外，他的其他兴趣包括：法语、日语、科学幻想小说和环游世界。他经常出现在网上，回答 comp.lang.c 中的问题。Ben 也是 Debian GNU/Linux 的维护人员（www.debian.org），同时是免费软件基金会（www.fsf.org）的软件作者。可以通过 E-mail: pfaffben@msu.edu 与 Ben 联系。

Peter Seebach 是 C 语言的业余爱好者。他成为 ANSI/ISO C 标准委员会成员只是“因为听起来很有趣”。他编程主要是因为这是在电子游戏中作弊的最好方法。

Steve Summit 是一个熟练的 C 语言程序员、作者和老师。20 年来，他几乎一直专门用 C 语言编程。他是 comp.lang.c FAQ 的维护人员。他的编程范围十分广泛，但所有程序都十分干净利索，可用性强，可移植性好，并且是正确的。

Mathew Watson 在 10 岁时，编写了他的第一个机器代码（因为高级语言在第一代 8 位微机上运行速度非常慢），并且自此之后迷上了计算机。他在伦敦大学学习计算机科学，并为不同的计算机公司工作。Mathew 目前正在等待着他第一个孩子的诞生，这样他/她可以向他展示 21 世纪的编程——正如 Mathew 为他父亲做的那样。

Stephan Wilms 是专业软件开发人员，他在工业图像处理以及心脏病研究与治疗医学系统方面富有经验。在工作中，他使用高度并行嵌入系统，并开发复杂的终端用户 Windows 应用软件。

Ian Woods 具有 6 年专业编程经验，目前他为一家研究公司编写实时嵌入软件。他专门使用 C 语言及一些汇编语言。他所做的许多工作需要离散数学及信号处理的专门知识。他目前在学习一种称为 Lojban 的口头语言，这是一种具有正式语法的设计逻辑语言，能够大大简化语音生成和处理。他的大部分兴趣集中在他的创新性想法上，涉及软件开发的许多领域。

Mike Wright 是 Innovative Systems Architect 的主要技术人员，这家公司位于犹他州，由于为美国国防部的检测和训练靶场提供技术支持而闻名。Mike 指导了高级任务分析及适用于内华达州“黑色”靶场的可视化工具的设计与开发。Mike 对他工作的挑战性十分满意，同时他很关注软件工程中的最新进展。可以通过 mike_wisdom@yahoo.com 与他联系。

献 辞

本书献给世界上最美丽的女士 Nicky。

——Richard Heathfield

致 谢

需要感谢许多人！首先自然要感谢合著作者。没有他们，就没有本书。需要特别提到的是 Dann Corbit 和 Ben Pfaff，他们详细剖析了作者编写的源代码，解决了其中的许多疑问。感谢 Chad Dixon，我每逢困境他都能及时提供帮助。感谢 Ian Kelly，他是英国的计算机权威。

非常感谢 Sams 公司的 Carol Ackerman，是他的大力资助才促使本书的完成。感谢 Gus Miklos 一贯沉稳的领导。感谢 Sams 公司的其他人员，正是他们的辛勤努力才使本书从设想变成现实。

此外尚有许多人毫无保留地付出了他们的时间、精力、想法和资源。感谢 Ola Angelsmark、Nick Cramer、Mark Dean、Jonathan George、Matt Gessner、Stephen Heathfield、John Hixson、Leibish Mermelstein、Edmund Stephen-Smith 和 Bryan Williams。

最后，需要提及的是世界上最好的一群黑客，他们通过在 comp.lang.c 上回答其他人的问题，为他人提供一些专业的技术帮助，以此省去了作者亲自请教的麻烦。

——Richard Heathfield

· 请向我们反馈信息

作为本书的读者，你的批评和评论对我们很重要。我们将重视你的观点，我们想知道我们应该做些什么，以及如何能做得更好；同时也想了解你希望我们出版什么领域的书籍，以及其他有益的见解。

作为 Sams 公司的合作者，我欢迎你的评论。你能通过传真、电子邮件或直接给我写信让我知道你喜欢本书的哪些部分或不喜欢本书的哪些部分——同样我们想知道为了使本书更好我们需要做些什么。

请注意对有关本书主题的一些技术问题，我不能提供帮助；同时由于我收到的信件太多，我可能不能回复每封信件。

写信时，一定要写清楚本书的题目和作者以及你的姓名、电话或传真号码。我将仔细检查你的评论，并与本书的作者和编辑共同讨论。

Fax: 317-581-4770

Email: opsys@mcp.com

Mail: Michael Stephens

Sams Publishing

201 West 103rd Street

Indianapolis, IN 46290 USA

目 录

第一部分 C 语言基础知识

第 1 章 欢迎阅读《标准 C 语言实用全书》	(2)
1.1 谁应当阅读本书	(3)
1.2 本书需要的基本知识	(3)
1.3 如何最好地学习本书	(5)
1.4 为什么还要写一本关于 C 语言的书	(6)
1.5 为什么要使用 ANSI C	(6)
1.6 本书面向何种平台	(7)
1.7 关于本书的源代码	(7)
1.8 代码的质量	(7)
1.9 本书的内容编排	(8)
1.10 版权	(9)
1.11 关于 CD	(9)
1.12 C 语言编程共同体	(10)
1.13 小结	(12)
第 2 章 编程标准:原因和对策	(13)
2.1 花括号的格式	(14)
2.2 空白的使用	(16)
2.3 结构化编程	(19)
2.4 初始化	(23)
2.5 静态变量和全局变量	(25)
2.6 标识符命名	(26)
2.7 编写有用的注释	(32)
2.8 一般的错误和误解	(36)
2.9 理解说明、定义和原型	(43)
2.10 可移植性的重要性	(44)
2.11 宏	(49)
2.12 C 与 C++	(49)
2.13 小结	(50)
第 3 章 优化	(51)
3.1 优化的要求和古老的历史	(51)
3.2 考虑性能	(52)
3.3 剖面测试系统	(57)

3.4	优化的算法策略	(61)
3.5	协作和并行优化	(71)
3.6	何时不优化	(74)
3.7	小结	(76)
第 4 章	处理日期和时间	(77)
4.1	日期和时间函数	(77)
4.2	千年虫:与日期相关的问题	(85)
4.3	有用的一点说明	(86)
4.4	小结	(89)
第 5 章	处理位和字节	(90)
5.1	C 语言中值的表达	(90)
5.2	整数值的表示	(91)
5.3	在位处理中使用无符号类型	(93)
5.4	位移	(93)
5.5	其他位操作符	(94)
5.6	位数组(位图)	(96)
5.7	位的计数	(100)
5.8	位反射	(102)
5.9	位字段	(105)
5.10	关于可移植性	(106)
5.11	小结	(110)
第 6 章	脱机数据存储和检索	(111)
6.1	目标和应用	(111)
6.2	基本技术	(114)
6.3	通用格式	(129)
6.4	高级技术	(136)
6.5	相关主题	(139)
6.6	小结	(140)
第 7 章	程序出错时:代码修补	(141)
7.1	处理诊断	(141)
7.2	调试常见错误	(146)
7.3	调试软件	(158)
7.4	预见失败,争取成功	(167)
7.5	调试过程	(171)
7.6	编程者的恶梦	(176)
7.7	小结	(178)

第 8 章 管理内存	(179)
8.1 什么是内存管理.....	(179)
8.2 常见的内存用法错误.....	(179)
8.3 内存分配函数出错时.....	(185)
8.4 Calloc 成功时做什么.....	(189)
8.5 保持控制.....	(189)
8.6 小结.....	(205)
第 9 章 模拟和控制器	(206)
9.1 有限状态机概述.....	(206)
9.2 灯开关范例.....	(207)
9.3 将 FSM 转变成代码.....	(208)
9.4 模拟和控制器应用程序.....	(209)
9.5 控制器的关键安全方面.....	(209)
9.6 模拟简单计算机.....	(211)
9.7 小结.....	(216)
第 10 章 递 归	(217)
10.1 什么是递归.....	(217)
10.2 怎样使用递归.....	(220)
10.3 熟悉递归.....	(224)
10.4 递归应用程序.....	(233)
10.5 小结.....	(234)

第二部分 数据结构

第 11 章 简单抽象数据结构	(236)
11.1 数组.....	(237)
11.2 单链表.....	(262)
11.3 双链表.....	(272)
11.4 循环链表.....	(285)
11.5 栈.....	(293)
11.6 队列.....	(301)
11.7 优先队列.....	(304)
11.8 双端队列.....	(311)
11.9 异型结构和对象树.....	(318)
11.10 小结.....	(318)
第 12 章 二叉搜索树	(320)
12.1 检查搜索算法.....	(320)
12.2 二叉树.....	(321)

12.3 平衡二叉树	(337)
12.4 小结	(358)
第 13 章 快速排序技术	(359)
13.1 分类数据	(359)
13.2 排序基础	(360)
13.3 有用的排序技术	(365)
13.4 归并策略	(380)
13.5 小结	(404)
第 14 章 多叉树	(405)
14.1 多叉树数据结构	(405)
14.2 如何使用多叉树	(406)
14.3 多叉树与二叉树及散列表的区别	(418)
14.4 小结	(418)
第 15 章 稀疏矩阵	(419)
15.1 什么是稀疏矩阵	(419)
15.2 为什么只用于稀疏数据	(424)
15.3 什么时候使用稀疏矩阵	(426)
15.4 构造稀疏矩阵	(430)
15.5 遍历稀疏矩阵	(452)
15.6 小结	(453)
第 16 章 图	(454)
16.1 定义图	(454)
16.2 环图和无环图	(458)
16.3 表示图	(458)
16.4 最短路径算法	(480)
16.5 最小生成树	(496)
16.6 最优化:最后的注解	(504)
16.7 小结	(505)

第三部分 高级专题

第 17 章 矩阵运算	(508)
17.1 矩阵是什么	(508)
17.2 简单的矩阵算术运算	(509)
17.3 使用 C 语言实现矩阵结构	(510)
17.4 从数组初始化	(517)
17.5 从文件获取矩阵	(518)
17.6 向 stdout 或文件写入 MATRIX_T 对象	(521)

17.7 相加和转置的完整实现	(522)
17.8 复杂矩阵概念	(524)
17.9 求解线性方程组	(537)
17.10 未来开发目标	(541)
17.11 其他方法	(542)
17.12 小结	(542)
第 18 章 数字信号处理	(543)
18.1 C 语言满足现实世界	(543)
18.2 数据压缩	(544)
18.3 错误检查和校正	(574)
18.4 PID 控制算法	(587)
18.5 小结	(601)
第 19 章 表达式分析和评价	(602)
19.1 问题陈述	(602)
19.2 答案陈述	(603)
19.3 波兰表示法	(612)
19.4 从普通形式转换到逆波兰形式	(614)
19.5 从波兰形式到评价的转换	(650)
19.6 分析错误的输入	(655)
19.7 小结	(656)
第 20 章 制作软件工具	(657)
20.1 优秀软件工具的特点	(658)
20.2 代码库	(660)
20.3 通用过滤器和工具	(661)
20.4 自动测试数据生成	(672)
20.5 代码生成器	(677)
20.6 小结	(704)
第 21 章 遗传算法	(705)
21.1 理解遗传算法	(705)
21.2 遗传的结构	(706)
21.3 突变操作	(708)
21.4 重组	(708)
21.5 选择	(709)
21.6 打开黑匣子	(711)
21.7 优化	(711)
21.8 应用实例:基因股票顾问	(713)
21.9 小结	(719)

第 22 章 跨平台开发:通信编程	(720)
22.1 规划移植	(720)
22.2 抽象层	(721)
22.3 套接字	(722)
22.4 跨平台应用程序	(736)
22.5 小结	(757)
第 23 章 使用 C 语言编写公网网关接口应用程序	(758)
23.1 什么是 CGI	(758)
23.2 CGI 入门	(759)
23.3 HTML 输入技术	(760)
23.4 CGI 环境	(764)
23.5 检索输入	(768)
23.6 分析查询字符串	(771)
23.7 样本应用程序:函数原型查寻.....	(776)
23.8 安全性考虑	(781)
23.9 小结	(783)
第 24 章 随机精度运算	(784)
24.1 误差怎样通过算术运算传播	(785)
24.2 溢出、下溢以及除零.....	(787)
24.3 计算次序	(787)
24.4 整数的长度	(788)
24.5 浮点数的长度	(791)
24.6 不要假设环境	(797)
24.7 极高和极低精度范例	(798)
24.8 表示极高精度的不同方法	(799)
24.9 字符串运算的样本程序	(800)
24.10 分数	(804)
24.11 通用运算例程	(805)
24.12 使用标准类型	(807)
24.13 选择自己的表示法	(815)
24.14 计算 e 的一万位	(816)
24.15 小结	(820)
第 25 章 自然语言处理技术	(821)
25.1 自然语言的语法和语义	(821)
25.2 自然语言输入的复杂性	(828)
25.3 自然语言输出的复杂性	(840)
25.4 小结	(842)

第 26 章 加密	(843)
26.1 评价安全性风险	(843)
26.2 为什么不生成新加密算法	(845)
26.3 选择加密算法	(847)
26.4 实现加密	(855)
26.5 小结	(867)
第 27 章 嵌入系统	(868)
27.1 使用 C 语言进行嵌入系统编程	(869)
27.2 嵌入 C 编程和 ANSI C	(874)
27.3 实时操作系统	(875)
27.4 典型范例 RTOS: RTEMS	(875)
27.5 小结	(876)
第 28 章 并行处理	(877)
28.1 基本概念	(877)
28.2 计算机和并行处理	(877)
28.3 使用 C 语言进行并行处理	(880)
28.4 小结	(888)
第 29 章 未来展望: C99	(889)
29.1 增加的新内容	(889)
29.2 我们失去了什么	(897)
29.3 改变之处	(897)
29.4 小结	(898)

附 录

附录 A GNU 通用公共许可证	(902)
附录 B 参考书目	(906)
附录 C CD 的内容	(909)

第一部分

C 语言基础知识

- 第 1 章 欢迎阅读《标准 C 语言实用全书》
- 第 2 章 编程标准：原因和对策
- 第 3 章 优化
- 第 4 章 处理日期和时间
- 第 5 章 处理位和字节
- 第 6 章 脱机数据存储和检索
- 第 7 章 程序出错时：代码修补
- 第 8 章 管理内存
- 第 9 章 模拟和控制器
- 第 10 章 递归

第 1 章 欢迎阅读《标准 C 语言实用全书》

本章要点

- 谁应当阅读本书
- 本书需要的基本知识
- 如何最好地学习本书
- 为什么还要写一本关于 C 语言的书
- 为什么要使用 ANSI C
- 本书面向何种平台
- 关于本书的源代码
- 代码的质量
- 本书的内容编排
- 版权
- 关于 CD
- C 语言编程共同体

到目前为止，Sams 公司出版了一系列实用全书，几乎覆盖了所有语言和编译程序。但是仍然有一个缺憾，即在这一系列中缺少完全关于 C 语言的图书。现在这一缺憾消除了，读者手中正是这样一本关于 C 语言的书。

关于 C 语言已经出版了许多书，C 语言的每一方面都被包含在内。那么，关于 C 语言还有哪些内容没有包含在内，需要我们去告诉大家的呢？

很明显，没有了。C 语言是一种简单但功能非常强大的语言，很多操作系统、视频游戏、微波炉、文字处理系统以及其他的电子产品都采用 C 语言来编制程序。如果 C 语言具有任何严重的缺陷，那么它就不会如此流行，而且本书的内容也要少得多。本书沿用传统，称为《标准 C 语言实用全书》，但我更想称它为《C 语言程序员实用全书》。我们并不是要对 C 语言本身进行详细探讨，而是要对 C 程序员讲述编程技巧。

为什么要对程序员讲述编程技巧呢？原因如下：

问：我正在进行一些精度要求非常高的计算，需要 500 位的精度，但我只能得到大约 15 位的精度，怎样才能得到更高的精度？

答：在 C 语言中无法做到。双精度型的数字位数由 `<float.h>` 中的 `DBL_DIG` 确定，在现代系统中一般为 15 位。你真的需要 500 位吗？

尽管这个问题和回答是我杜撰的，但是如果在 Usenet 上提出这个问题的话，很有可能得到这类回答。长期以来，人们一直在说：“这在 C 语言中做不到，根本别想”。在本书中，我们将说明这些是可以做到的。虽然实现这些需要进行大量的工作和思考，但这确实是可以