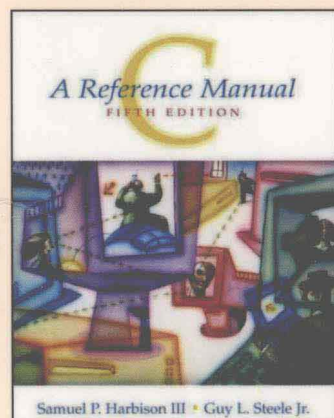


原书第5版

C 语言 参考手册



C: A Reference Manual
(Fifth Edition)

Samuel P. Harbison III 著
Guy L. Steele Jr.

徐波 等译



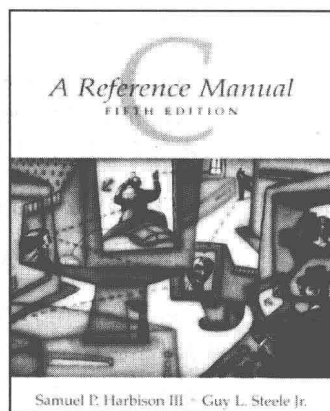
机械工业出版社
China Machine Press



C语言经典译丛

原书第5版

C语言 参考手册



C: A Reference Manual (Fifth Edition)

美) Samuel P. Harbison III 著
Guy L. Steele Jr.
徐波 等译



机械工业出版社
China Machine Press

本书是C语言的详尽参考手册，分为两部分：第一部分讨论了C语言的所有语言特征，包括词法、预处理机制、声明、类型、表达式、语句以及函数等基本语言特征；第二部分讨论了C语言的标准库，针对它们不同的功能分别进行详细介绍。为了帮助读者理解相关概念，本书在讨论C语言及其标准库的细节时，提供了许多实例和解释。第一部分的各章还提供了练习题和主要练习的解答，这些练习可帮助读者加深理解C语言的基本功能和特性。

本书可以作为高等院校计算机专业C语言课程的教材和参考书，也可供程序开发人员参考。

Authorized translation from the English language edition, entitled C: A Reference Manual, 5th Edition, 0-13-089592-X by Samuel P. Harbison III, Guy L. Steele Jr., published by Pearson Education, Inc., publishing as Addison Wesley Professional, Copyright © 2002 .

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 PEARSON EDUCATION ASIA LTD., and CHINA MACHINE PRESS Copyright © 2011.

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。

封底无防伪标均为盗版

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：01-2002-3650

图书在版编目（CIP）数据

C语言参考手册（原书第5版）/（美）哈比森（Harbison, S. P.）等著；徐波等译. —北京：机械工业出版社，2011.6

（C语言经典译丛）

书名原文：C: A Reference Manual, Fifth Edition

ISBN 978-7-111-34684-5

I. C… II. ①哈… ②徐… III. C语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆CIP数据核字（2011）第088147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：秦 健

北京市荣盛彩色印刷有限公司印刷

2011年6月第1版第1次印刷

186mm×240mm·26印张

标准书号：ISBN 978-7-111-34684-5

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；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com

译者序



在众多讲述C语言的书籍中，本书是一部非常优秀的作品。它详细、精确地描述了C语言所有方面的特性。说它是C程序员案头必备的参考手册，绝对不是夸张。

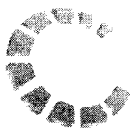
本书的作者都是享有盛誉的C语言专家，并且都曾担任C/C++语言标准委员会的成员，对C语言有着深入、独到的理解。本书正是他们殚精虑竭之作。这本手册技术细节描述精确，组织条理清晰，内容完备详尽而又简明扼要，在广度、深度和精度方面都是出类拔萃的。并且经过5个版本的完善之后，本书已经是一部非常成熟的作品。

本书并不是一本指导初学者从头学习C语言的教程，而是一本供专业程序员在开发程序时随时查阅的参考书。它对C语言各个方面的描述既精确又扼要，对于C语言有关的任何问题，几乎都可以在本书中找到答案。它所涵盖的C语言版本包括传统C、C89、C89修正案1和C99，尤其是对C99的详细介绍，是其他C语言名作所不具备的。因此，不管读者所使用的是什么类型的C编译器，都能够从本书中受益。另外，本书还专门注意了C和C++所存在的兼容性问题，对于想开发C/C++兼容的应用程序的开发人员而言，尤其具有价值。难怪有专家评论，除了Brian W. Kernighan和Dennis M. Ritchie的经典名作《The C Programming Language》^①之外，唯一离不开的就是这本书。

本书分为两个部分，第一部分是C语言本身的描述，第二部分则介绍了标准函数库。尽管本书介绍了C语言的完整知识，但它的篇幅却不是非常庞大，这一方面是由于C语言本身比较简洁，另一方面也是由于本书所使用的语言非常精练。对于专业程序员而言，最好的参考手册就是能够准确而精练地描述问题，并提供适当的例子。本书就是一本完全符合这个条件的参考手册。

参加本书翻译工作的人员有徐波、陈永军、李福军、毛玉山、冯毅晓、陈丽娜、杨洁、张瑜、应巧敏、顾沂、董志杰、许朝阳、龚建峰等。

① 该书的中译本和影印版均已由机械工业出版社引进出版，书名为《C程序设计语言（英文版·第2版）》（978-7-111-19626-0），《C程序设计语言（第2版·新版）》（978-7-111-12806-0）。——编辑注



前言

本书是C编程语言的参考手册。本书的目标是对C语言的基本概念和运行时函数库提供完整而准确的描述，同时还强调了以正确性、可移植性和可维护性为基本出发点的良好编程风格。

本书预期读者已经理解基本的编程概念，并希望很多读者已经可以用C语言熟练编程。为了保持参考手册的格式，本书从头到尾介绍C语言的词法结构、预处理器、声明、类型、表达式、语句、函数和运行时函数库。书中包括了许多交叉引用，使读者可以从任何地方入手。

本版完整地介绍了最新的C语言国际标准ISO/IEC 9899:1999 (C99)，明确地指出了语言本身和库函数的哪些特性是C99新增的，并指出了C99与原有的C89标准的不同之处。这是目前唯一适用于所有主流C语言版本的参考手册，涵盖了传统C、1989年C标准、1995年对C89的修改和补充以及最新的C99标准。本书还介绍了标准C和标准C++的公共子集Clean C。尽管C99增加了许多新信息，但本书并没有对章节结构进行很大的改动，这样熟悉旧版的读者可以顺利地找到他所需要的材料。

本书最初源于我们在Tartan公司的工作，为从微机到大型机的一系列计算机开发C语言编译器系列。我们要求编译器文档齐全，提供精确而有用的错误诊断信息，并能产生性能卓越的目标代码。经过某个编译器正确编译的C程序应该能够在硬件差别允许的情况下，在其他所有编译器中也能顺利通过编译。

1984年，尽管C语言已经非常普及，但还没有一本书对C语言的介绍精确到足以指导我们设计新的编译器。另外，当时的语言描述对于那些想使用编译器对C程序进行深入分析的编程人员和客户而言仍然不够精确。在本书中，我们特别注重那些影响程序的清晰度、目标代码的有效性以及不同环境中程序移植性的语言特性。

Web站点

读者可以访问本书的Web站点CReferenceManual.com，其中包括本书的示例代码、更深入的讨论、澄清的问题以及C语言资源的更多链接。

致谢

在准备本书的第5版时，特别感谢原NCITS JII主席Rex Jaeschke、芬兰赫尔辛基的Antoine Trux以及爱迪生设计集团创始人Steve Adamczyk的帮助。

对于本书以前的版本提供过帮助的人员包括Jeffrey Esakov、Alan J.Filipski、Frank J.Wagner、Debra Martin、P.J.Plauger和Steve Vinoski。其他提供过帮助的人员包括Aurelio Bignoli、Steve

Clamage、Arthur Evans、Jr.、Roy J. Fuller、Morris M.Kessan、George V.Reilly、Mark Lan、Mike Hewett、Charles Fischer、Kevin Rodgers、Tom Gibb、David Lim、Stavros Macrakis、Steve Vegdahl、Christopher Vickery、Peter van der Linden和Dave Wilson。还要感谢Machael Angus、Mady Bauer、Larry Breed、Sue Broughton、Alex Czajkowski、Robert Firth、David Gaffney、Steve Gorman、Dennis Hamilton、Chris Hanna、Ken Harrenstien、Rex Jaeschke、Don Lindsay、Tom MacDonald、Peter Nelson、Joe Newcomer、Kevin Nolish、David Notkin、Peter Plamondon、Roger Ray、Larry Rosler、David Spencer和Barbara Steele。

本书最初的一些示例程序参考了下列著作中的算法：

- Beeler, Michael, Gosper, R.William和Schroepel, Richard, 《HAKMEM》, AI Memo 239 (麻省理工学院人工智能实验室, 1972年2月)。
- Bentley, Jon Louis, 《Writting Efficient Programs》(Prentice-Hall, 1982)。
- Bentley, Jon Louis, 《Programming Pearls》(1983年8月起出现在《Communications of the ACM》的每月专栏)。
- Kernighan, Brian W.和Ritchie, Dennis M. 《The C Programming Language》(Prentice-Hall, 1978)。
- Knuth, Donald E. 《The Art of Computer Programming》^①第1-3卷 (Addison-Wesley, 1968、1969、1973、1981)。
- Sedgewick, Robert, 《Algorithms》^② (Addison-Wesley, 1983)。

感谢这些作者的灵感。

还有一点需要说明, Guy Steele先生由于工作繁忙, 无法在新版本中参与更多工作。虽然书中仍然处处体现出他对C语言的精辟分析, 但新版本所存在的任何问题, 都不应该再由他负责。

《C语言参考手册》一书已经出版了很多年, 感谢所有读者多年来对它的关爱。

Sam Harbison

于宾夕法尼亚州匹兹堡市

harbison@CAREferenceManual.com

① 该书影印版已由机械工业出版社引进出版, 书名为《计算机程序设计艺术 第1卷 基本算法 (英文版·第3版)》(978-7-111-22709-0)、《计算机程序设计艺术 第2卷 半数值算法 (英文版·第3版)》(978-7-111-22718-2)和《计算机程序设计艺术 第3卷 排序和查找 (英文版·第2版)》(978-7-111-22717-5)。——编辑注

② 该书影印版已由机械工业出版社引进出版, 书名为《算法: C语言实现 (第1~4部分) 基础知识、数据结构、排序及搜索 (英文版·第3版)》(978-7-111-19764-X)和《算法: C语言实现 (第5部分) 图算法 (英文版·第3版)》(978-7-111-19769-0)。——编辑注



目 录

译者序
前 言

第一部分 C语言

第1章 概述	2
1.1 C语言的演变	2
1.1.1 传统C	2
1.1.2 标准C (1989)	2
1.1.3 标准C (1995)	3
1.1.4 标准C (1999)	3
1.1.5 标准C++	4
1.1.6 本书内容	4
1.2 应该使用哪种C语言	4
1.3 C编程概述	5
1.4 一致性	6
1.5 语约定	7
第2章 词法元素	8
2.1 字符集	8
2.1.1 执行字符集	9
2.1.2 空白字符和行终止符	9
2.1.3 字符编码	10
2.1.4 三字符组	10
2.1.5 多字节字符和宽字符	11
2.2 注释	13
2.3 标记	14
2.4 操作符和分隔符	15
2.5 标识符	15
2.6 关键字	17
2.7 常量	18
2.7.1 整数常量	18
2.7.2 浮点数常量	22
2.7.3 字符常量	23

2.7.4 字符串常量	25
2.7.5 转义字符	27
2.7.6 字符转义码	28
2.7.7 数值转义码	29
2.8 C++兼容性	29
2.8.1 字符集	30
2.8.2 注释	30
2.8.3 操作符	30
2.8.4 标识符和关键字	30
2.8.5 字符常量	30
2.9 关于字符集、指令集和编码	30
2.10 练习	32
第3章 C预处理器	34
3.1 预处理器命令	34
3.2 预处理器词法约定	35
3.3 定义和替换	36
3.3.1 类似对象的宏定义	36
3.3.2 定义带参数的宏	37
3.3.3 宏表达式的重新扫描	39
3.3.4 预定义的宏	40
3.3.5 取消宏定义和重新定义宏	42
3.3.6 宏展开的优先级错误	42
3.3.7 宏参数的副作用	43
3.3.8 把标记转换为字符串	43
3.3.9 宏展开中的标记合并	44
3.3.10 宏的可变参数列表	45
3.3.11 其他问题	46
3.4 文件包含	46
3.5 条件编译	48
3.5.1 #if、#else和#endif命令	48
3.5.2 #elif命令	49
3.5.3 #ifdef和#ifndef命令	50

3.5.4 条件命令中的常量表达式	51	4.5.4 函数声明器	78
3.5.5 defined操作符	52	4.5.5 声明器的组合	80
3.6 显式的行号	52	4.6 初始化值	81
3.7 pragma指令	53	4.6.1 整数	82
3.7.1 标准pragma命令	53	4.6.2 浮点数	83
3.7.2 标准pragma指令的位置	54	4.6.3 指针	83
3.7.3 _Pragma操作符	54	4.6.4 数组	84
3.8 错误指令	54	4.6.5 枚举	86
3.9 C++兼容性	55	4.6.6 结构	86
3.10 练习	55	4.6.7 联合	87
第4章 声明	57	4.6.8 省略花括号	88
4.1 声明的组织形式	57	4.6.9 指定成员的初始化值	88
4.2 术语	58	4.7 隐式声明	89
4.2.1 作用域	58	4.8 外部名称	90
4.2.2 可见性	59	4.8.1 初始化值模型	90
4.2.3 前向引用	59	4.8.2 省略的存储类别模型	90
4.2.4 名称的重载	60	4.8.3 公共模型	90
4.2.5 重复声明	61	4.8.4 混合公共模型	91
4.2.6 重复可见性	62	4.8.5 总结和建议	91
4.2.7 范围	62	4.8.6 未引用的外部声明	92
4.2.8 初始值	63	4.9 C++兼容性	92
4.2.9 外部名称	64	4.9.1 作用域	92
4.2.10 编译时名称	65	4.9.2 标签和typedef名称	92
4.3 存储类别和函数指定符	65	4.9.3 用于类型的存储类别指定符	93
4.3.1 默认存储类别指定符	66	4.9.4 const类型限定符	93
4.3.2 存储类别指定符的例子	66	4.9.5 初始化值	93
4.3.3 函数指定符	67	4.9.6 隐式声明	93
4.4 类型指定符和限定符	67	4.9.7 定义和引用声明	94
4.4.1 默认类型指定符	68	4.9.8 函数链接	94
4.4.2 缺失的声明器	69	4.9.9 无参函数	94
4.4.3 类型限定符	70	4.10 练习	94
4.4.4 const	70	第5章 类型	97
4.4.5 volatile和序列点	72	5.1 整数类型	98
4.4.6 restrict	74	5.1.1 有符号整数类型	98
4.5 声明器	75	5.1.2 无符号整数类型	101
4.5.1 简单声明器	75	5.1.3 字符类型	102
4.5.2 指针声明器	76	5.1.4 扩展整数类型	104
4.5.3 数组声明器	77	5.1.5 布尔类型	104

5.2 浮点类型	104	5.11.6 指针兼容性	139
5.3 指针类型	108	5.11.7 源文件之间的兼容性	139
5.3.1 通用指针	109	5.12 类型名称和抽象声明器	139
5.3.2 null指针和非法指针	109	5.13 C++兼容性	141
5.3.3 使用指针的一些警告	110	5.13.1 枚举类型	141
5.4 数组类型	111	5.13.2 typedef名称	141
5.4.1 数组和指针	111	5.13.3 类型兼容性	141
5.4.2 多维数组	112	5.14 练习	141
5.4.3 数组边界	112	第6章 转换和表示形式	144
5.4.4 操作	113	6.1 表示形式	144
5.4.5 可变长度的数组	113	6.1.1 存储单位和数据长度	144
5.5 枚举类型	115	6.1.2 字节顺序	145
5.6 结构类型	117	6.1.3 对齐限制	146
5.6.1 结构类型引用	119	6.1.4 指针的长度	147
5.6.2 结构上的操作	120	6.1.5 地址模型的效果	147
5.6.3 成员	120	6.1.6 类型表示形式	149
5.6.4 结构成员的布局	122	6.2 转换	150
5.6.5 位段	122	6.2.1 表示形式的变化	150
5.6.6 移植性问题	124	6.2.2 细微的转换	150
5.6.7 结构的大小	125	6.2.3 转换为整数类型	150
5.6.8 灵活数组成员	126	6.2.4 转换为浮点类型	152
5.7 联合类型	127	6.2.5 转换为结构和联合类型	152
5.7.1 联合成员的布局	127	6.2.6 转换为枚举类型	152
5.7.2 联合的大小	128	6.2.7 转换为指针类型	152
5.7.3 使用联合类型	129	6.2.8 转换为数组和函数类型	153
5.7.4 (其他)使用联合类型	130	6.2.9 转换为void类型	153
5.8 函数类型	131	6.3 寻常转换	154
5.9 void类型	133	6.3.1 类型转换	154
5.10 typedef名称	133	6.3.2 赋值转换	154
5.10.1 用于函数类型的typedef名称	135	6.3.3 寻常单目转换	155
5.10.2 重定义typedef名称	135	6.3.4 寻常双目转换	157
5.10.3 编译器的说明	136	6.3.5 默认的函数实参转换	158
5.11 类型兼容性	136	6.3.6 其他函数转换	159
5.11.1 等价的类型	136	6.4 C++兼容性	159
5.11.2 枚举兼容性	137	6.5 练习	159
5.11.3 数组兼容性	138	第7章 表达式	161
5.11.4 函数兼容性	138	7.1 对象、左值和指示符	161
5.11.5 结构和联合兼容性	139	7.2 表达式和优先级	162

7.2.1 操作符的优先级和结合性	162	7.11.3 初始化值常量表达式	200
7.2.2 溢出和其他算术异常	163	7.12 求值的顺序	201
7.3 基本表达式	164	7.13 被丢弃的值	202
7.3.1 名称	164	7.14 内存访问的优化	203
7.3.2 字面值	165	7.15 C++兼容性	204
7.3.3 带括号的表达式	165	7.16 练习	204
7.4 后缀表达式	166	第8章 语句	206
7.4.1 下标表达式	166	8.1 语句的基本语法规则	206
7.4.2 成员选择	168	8.2 表达式语句	207
7.4.3 函数调用	169	8.3 标签语句	207
7.4.4 后缀增值和减值操作符	171	8.4 复合语句	208
7.4.5 复合字面值	172	8.5 条件语句	210
7.5 单目表达式	173	8.5.1 多路条件语句	210
7.5.1 类型转换	174	8.5.2 悬疑的else问题	211
7.5.2 sizeof操作符	174	8.6 迭代式语句	212
7.5.3 单目负号和正号	176	8.6.1 while语句	212
7.5.4 逻辑反	176	8.6.2 do语句	213
7.5.5 位反	177	8.6.3 for语句	214
7.5.6 取地址操作符	177	8.6.4 使用for语句	215
7.5.7 间接访问	178	8.6.5 多个控制变量	217
7.5.8 前缀增值和减值操作符	179	8.7 switch语句	218
7.6 双目操作符表达式	180	8.8 break和continue语句	220
7.6.1 乘除操作符	180	8.9 return语句	222
7.6.2 加减操作符	182	8.10 goto语句	223
7.6.3 移位操作符	183	8.11 空语句	224
7.6.4 关系操作符	185	8.12 C++兼容性	224
7.6.5 相等操作符	186	8.12.1 复合语句	224
7.6.6 位操作符	188	8.12.2 循环中的声明	224
7.6.7 整数集合的例子	189	8.13 练习	225
7.7 逻辑操作符表达式	192	第9章 函数	226
7.8 条件表达式	194	9.1 函数定义	226
7.9 赋值表达式	195	9.2 函数原型	228
7.9.1 简单赋值	196	9.2.1 原型出现在什么时候	230
7.9.2 复合赋值	197	9.2.2 混合使用原型和非原型声明	230
7.10 线性表达式	197	9.2.3 合理地使用原型	231
7.11 常量表达式	198	9.2.4 原型和调用约定	231
7.11.1 预处理器常量表达式	199	9.2.5 标准C和传统C的兼容性	233
7.11.2 整型常量表达式	200	9.3 形式参数声明	233

9.4 形参类型的调整	235
9.5 参数传递约定	236
9.6 形参的一致	237
9.7 函数的返回类型	238
9.8 返回类型的一致	239
9.9 main函数	239
9.10 内联函数	240
9.11 C++兼容性	242
9.11.1 原型	242
9.11.2 形参和返回类型中的类型声明	242
9.11.3 返回类型的一致	242
9.11.4 main	242
9.11.5 内联	243
9.12 练习	243

第二部分 C函数库

第10章 函数库简介	246
10.1 标准C工具	246
10.2 C++兼容性	247
10.3 库头文件和名称	249
10.3.1 assert.h	249
10.3.2 complex.h	249
10.3.3 ctype.h	249
10.3.4 errno.h	249
10.3.5 fenv.h	249
10.3.6 float.h	250
10.3.7 inttypes.h	250
10.3.8 iso64.h	250
10.3.9 limits.h	250
10.3.10 locale.h	250
10.3.11 math.h	251
10.3.12 setjmp.h	252
10.3.13 signal.h	252
10.3.14 stdarg.h	252
10.3.15 stdbool.h	252
10.3.16 stddef.h	252
10.3.17 stdint.h	252
10.3.18 stdio.h	252

10.3.19 stdlib.h	253
10.3.20 string.h	253
10.3.21 tgmath.h	253
10.3.22 time.h	254
10.3.23 wchar.h	254
10.3.24 wctype.h	254

第11章 标准语言附加

11.1 NULL、ptrdiff_t、size_t、offsetof	255
11.2 EDOM、ERANGE、EILSEQ、errno、 strerror、perror	256
11.3 bool、false、true	257
11.4 va_list、va_start、va_arg、va_end	258
11.5 标准C的操作符宏	261

第12章 字符处理

12.1 isalnum、isalpha、isctrl、iswalnum、 iswalphabet、iswcntrl	262
12.2 iscsym、iscsymf	264
12.3 isdigit、isodigit、isxdigit、iswdigit、 iswxdigit	264
12.4 isgraph、isprint、ispunct、iswgraph、 iswprint、iswpunct	265
12.5 islower、isupper、iswlower、iswupper	266
12.6 isblank、isspace、iswhite、iswspace	266
12.7 toascii	267
12.8 toint	267
12.9 tolower、toupper、tolower、 toupper	267
12.10 wctype_t、wctype、iswctype	268
12.11 wctrans_t、wctrans	269

第13章 字符串处理

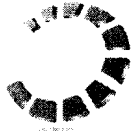
13.1 strcat、strncat、wscat、wcsncat	270
13.2 strcmp、strncmp、wcscmp、wcsncmp	271
13.3 strcpy、strncpy、wcscpy、wcsncpy	272
13.4 strlen、wcslen	273
13.5 strchr、strchr、wcschr、wcschr	273
13.6 strspn、strcspn、strpbrk、strpbrk、 wcspn、wcscspn、wcpbrk	274
13.7 strstr、strtok、wcsstr、wcstok	275

13.8	strtod, strtod, strtold, strtol, strtoll, strtoul, strtoull	276	15.11.5	精度	303
13.9	atof, atoi, atol, atoll	277	15.11.6	长度规范	303
13.10	strcoll, strxfrm, wcscoll, wcsxfrm	277	15.11.7	转换操作	304
第14章	内存函数	279	15.12	v[x]printf, v[x]scanf	311
14.1	memchr, wmemchr	279	15.13	fread, fwrite	312
14.2	memcmp, wmemcmp	279	15.14	feof, ferror, clearerr	313
14.3	memcpy, memccpy, memmove, wmemcpy, wmemmove	280	15.15	remove, rename	314
14.4	memset, wmemset	281	15.16	tmpfile, tmpnam, mktemp	314
第15章	输入/输出工具	282	第16章	基本工具	316
15.1	FILE, EOF, wchar_t, WEOF	283	16.1	malloc, calloc, malloc, calloc, free, cfree	316
15.2	fopen, fclose, fflush, freopen, fwide	284	16.2	rand, srand, RAND_MAX	318
15.2.1	文件模式	285	16.3	atof, atoi, atol, atoll	319
15.2.2	文件的定向	286	16.4	strtod, strtod, strtold, strtol, strtoll, strtoul, strtoull	319
15.3	setbuf, setvbuf	287	16.5	abort, atexit, exit, _Exit, EXIT_FAILURE, EXIT_SUCCESS	321
15.4	stdin, stdout, stderr	287	16.6	getenv	322
15.5	fseek, ftell, rewind, fgetpos, fsetpos	288	16.7	system	323
15.5.1	fseek和ftell	287	16.8	bsearch, qsort	323
15.5.2	fgetpos和fsetpos	290	16.9	abs, labs, llabs, div, ldiv, lldiv	325
15.6	fgetc, fgetwc,getc, getwc, getchar, getwchar, ungetc, ungetwc	290	16.10	mblen, mbtowc, wctomb	326
15.7	fgets, fgets, gets	291	16.10.1	编码和转换状态	327
15.8	fscanf, fwscanf, scanf, wscanf, sscanf, swscanf	292	16.10.2	长度函数	327
15.8.1	控制字符串	293	16.10.3	转换为宽字符	327
15.8.2	转换规范	293	16.10.4	宽字符转换为多字节字符	328
15.9	fputc, fputwc, putc, putwc, putchar, putwchar	298	16.11	mbstowcs, wcstombs	328
15.10	fputs, fputws, puts	299	16.11.1	转换为宽字符串	328
15.11	fprintf, printf, sprintf, snprintf, fwprintf, wprintf, swprintf	299	16.11.2	从宽字符串转换为多字节字符	329
15.11.1	输出格式	300	第17章	数学函数	331
15.11.2	转换规范	301	17.1	abs, labs, llabs, div, ldiv, lldiv	331
15.11.3	转换标志	301	17.2	fabs	332
15.11.4	最小字段宽度	302	17.3	ceil, floor, lrint, llrint, lround, llround, nearbyint, round, rint, trunc	332
			17.4	fmod, remainder, remquo	333
			17.5	frexp, ldexp, modf, scalbn	334
			17.6	exp, exp2, expm1, ilogb, log, log10, log1p, log2, logb	334

17.7	cbrt、fma、hypot、pow、sqrt	335	21.1.4	PRI...和SCN...格式字符串宏	362
17.8	rand、srand、RAND_MAX	336	21.2	固定长度的整数类型	363
17.9	cos、sin、tan、cosh、sinh、tanh	336	21.3	具有最小宽度的最小长度类型	364
17.10	acos、asin、atan、atan2、acosh、 asinh、atanh	337	21.4	具有最小宽度的快速类型	364
17.11	fdim、fmax、fmin	338	21.5	指针长度和最大长度整数类型	365
17.12	类型通用的宏	338	21.6	ptrdiff_t、size_t、wchar_t、wint_t和 sig_atomic_t的范围	366
17.13	erf、erfc、lgamma、tgamma	341	21.7	imaxabs、imaxdiv、imaxdiv_t	366
17.14	fpclassify、isfinite、isinf、isnan、 isnormal、signbit	342	21.8	strtoimax、strtouimax	367
17.15	copysign、nan、nextafter、 nexttoward	343	21.9	wcstouimax、wcstoumax	367
17.16	isgreater、isgreaterequal、isless、 islessequal、islessgreater、isunordered	343	第22章 浮点环境	368	
第18章 时间和日期函数	345	22.1	介绍	368	
18.1	clock、clock_t、CLOCKS_PER_SEC、 times	345	22.2	浮点环境	369
18.2	time、time_t	346	22.3	浮点异常	369
18.3	asctime、ctime	346	22.4	浮点四舍五入模式	370
18.4	gmtime、localtime、mktime	347	22.5	浮点表达式收缩	371
18.5	difftime	348	第23章 复数运算	372	
18.6	strtime、wcsftime	349	23.1	复数函数库约定	372
第19章 控制函数	352	23.2	complex、_Complex_I、imaginary、 _Imaginary_I、I	372	
19.1	assert、NDEBUG	352	23.3	CX_LIMITED_RANGE	372
19.2	system、exec	352	23.4	cacos、casin、catan、ccos、csin、 ctan	373
19.3	exit、abort	353	23.5	cacosh、casinh、catanh、ccosh、csinh、 ctanh	374
19.4	setjmp、longjmp、jmp_buf	353	23.6	cexp、clog、cabs、cpow、csqrt	374
19.5	atexit	354	23.7	carg、cimag、creal、conj、cproj	375
19.6	signal、raise、gsignal、ssignal、 psignal	354	第24章 宽字符和多字节字符工具	377	
19.7	sleep、alarm	356	24.1	基本类型和宏	377
第20章 区域	357	24.2	宽字符和多字节字符之间的转换	377	
20.1	setlocale	357	24.3	宽字符串和多字节字符串之间的转换	379
20.2	localeconv	358	24.4	转换为算术类型	380
第21章 扩展整数类型	361	24.5	输入和输出函数	380	
21.1	基本规则	361	24.6	字符串函数	381
21.1.1	类型的种类	361	24.7	日期和时间转换	382
21.1.2	要么都定义，要么都不定义	361	24.8	宽字符分类和映射函数	382
21.1.3	MIN和MAX限制	361	附录A ASCII字符集	383	
			附录B 语法	384	
			附录C 练习答案	396	

第一部分

C 语言



第1章 概述

第2章 词法元素

第3章 C预处理器

第4章 声明

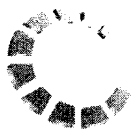
第5章 类型

第6章 转换和表示形式

第7章 表达式

第8章 语句

第9章 函数



第1章

概 述

20世纪70年代初，Dennis Ritchie在贝尔实验室设计了C语言。C语言的前身可以追溯到ALGOL（1960），历经剑桥的CPL（1963）、Martin Richards的BCPL（1967）以及Ken Thompson在贝尔实验室所开发的B语言（1970）发展而来。尽管C语言是一种通用用途的编程语言，但它在传统上用于系统编程。特别值得一提的是，著名的UNIX操作系统一开始就是用C语言编写的。

C语言流行的原因是多方面的。它小巧、高效，是一种功能强大的编程语言，并且具有丰富的运行时函数库。它提供了对计算机的精确控制，却没有采用太多的隐藏机制。由于C语言的标准化的早在十多年前就已完成，所以程序员们已经习惯了使用C语言。一般来说，用C语言能够非常容易地编写可以在不同国家、不同语言的不同计算机系统中进行移植的程序。另外，现在存在大量的遗留C代码，它们需要修改和扩展。

从20世纪90年代后期开始，C语言的地位逐渐受到它的“大哥”C++的挤压。但是，C语言仍然有大量忠实的追随者，并且依然非常流行，因为程序员在很多时候并不需要使用C++所提供的特性，或者不喜欢C++所带来的其他开销。

C语言经受住了时间的考验，经验丰富的程序员可以用C语言编写出快速高效的代码，数以百万行计的C程序代码充分证明了它的生命力。

1.1 C语言的演变

1984年，当我们编写本书的第1版时，C语言已经广泛地使用。但是，它还没有制订官方标准，也不存在语言的精确描述。程序员们事实上所使用的标准是当时的C编译器。C语言在1989年成为一种国际标准，1994年进行了一次修订，1999年又进行了一次重要的修订。

简单地修改语言的定义并不会自动更改世界上数以百万行计的C程序代码。我们努力使本书与时俱进，使程序员在遇到各种变体的C语言时都能把本书作为参考手册。

1.1.1 传统C

C语言的最初描述出现在Brian Kernighan和Dennis Ritchie所著的《The C Programming Language》第1版（Prentice Hall，1978），通常称为“K&R”。在这本书出版之后，C语言又经历了一些较小的变化，添加了一些新特性，并删除了一些不合时宜的特性。我们把20世纪80年代早期公认的C语言定义称为传统C，也就是C语言被标准化之前的一种语言变体。当然，众多C语言编译器开发商都对传统C进行了各自的扩展。

1.1.2 标准C（1989）

1982年，美国国家标准委员会（ANSI）认识到对C语言进行标准化有助于C语言在商业编程

领域得到更广泛的使用，于是成立了一个委员会，着手制订C语言及其运行时函数库的标准。这个称为X3J11的委员会（现在称为NCTIS J11）由Jim Brodie担任主席，成功地制订了一个标准，并于1989年被采纳为《美国国家标准X3.159-1989》，或称ANSI C。

考虑到编程是一项国际性的活动，因此在完成了ANSI C标准之后，国际标准组织（ISO）又创立了一个称为ISO/IEC JTC1/SC22/WG14的国际标准化小组。这个小组由P.J.Plauger所领导，对ANSI C只进行了少量的编辑性修改，把它转换为国际标准ISO/IEC 9899:1990。随后，ANSI也采纳了ISO/IEC标准，人们把这种标准简单地称为“标准C”。由于C的标准最终仍然会被修改，所以我们把它称为标准C（1989），或简称为C89。

从传统C到C89的改变包括：

- 添加了真正的标准函数库。
- 新的预处理器命令和特性。
- 函数原型，允许程序员在函数声明中指定参数的类型。
- 增加了一些新的关键字，包括const、volatile和signed。
- 宽字符、宽字符串和多字节字符。
- 在转换规则、声明和类型检查方面的许多小改动和澄清。

1.1.3 标准C（1995）

作为对C语言标准的正常维护，WG14对C89进行了两处技术修订（缺陷修复）和一个补充（扩展）。总体来说，这次修订对C89的修改还是比较温和的，主要是增加了一些新的函数库，其结果就是我们所说的“C89修正案1”，或者“C95”。它对C89的修改包括：

- 3个新的标准库头文件：iso646.h、wctype.h和wchar.h。
- 几个新的标记和宏，用于替换有些国家的字符集中不存在的操作符和标点符号。
- printf/scanf函数家族的一些新的格式代码。
- 大量用于多字节字符和宽字符的新函数以及一些类型和常量。

1.1.4 标准C（1999）

ISO/IEC标准必须定期审查和更新。1995年，WG14开始对C语言标准进行更深入的修订，于1999年完成了这项任务并获得了批准。新标准称为ISO/IEC 9899:1999或“C99”，它代替了以前的标准（以及所有的修正和补充），现在已经成为C语言的官方标准。开发商们对它们的C语言编译器进行了更新，以遵循新标准。

C99在C89和C95的语言及函数库上添加了许多新特性，包括：

- 复数运算。
- 整数类型的扩展，包括更长的标准类型。
- 可变长度的数组。
- 布尔类型。
- 对非英语字符集提供了更好的支持。
- 对浮点类型提供了更好的支持，包括所有类型的数学函数。

- C++风格的注释 (//)。

和C95相比，C99是一次相当大的修订，因为它不仅对语言本身作了一些修改，而且对标准函数库进行了扩展。C99标准文档的篇幅要明显多于C89标准文档。但是，这些修改是“按照C的精神”进行的，它们并没有改变C语言的本质。

1.1.5 标准C++

C++语言是在20世纪80年代早期由Bjarne Stroustrup在AT&T的贝尔实验室设计的。现在，它在很大程度上已经取代C语言成为主流的编程语言。大多数C语言编译器实际上是C/C++编译器，程序员可以根据需要选择这两种语言之一。C++语言本身也被标准化，即ISO/IEC 14882:1998，或“标准C++”。C++语言在C语言的基础上进行了很多扩展，满足了程序员开发大型程序的需要，包括增强的类型检查以及对面向对象编程的支持。但是，C++也是世界上最为复杂的编程语言之一，存在许多意想不到的陷阱。

标准C++近乎是（但不完全是）标准C的超集。由于C语言和C++语言标准是根据不同的方案开发的，因此它们之间可能存在一些不协调之处。而且，C语言有意与C++语言保持距离。例如，C语言并没有试图增加C++语言中“类”这个特性的简化版本。

用标准C和标准C++的公共子集编写C代码也是可行的，有些人把这种代码称为Clean C。用这种代码编写的程序既可以编译为C程序，也可以编译为C++程序。由于C++一般具有比标准C更严格的规则，因此Clean C可以看成是一个更好、更具移植性的子集。如果使用Clean C编写程序，必须考虑下面这些情况：

- Clean C程序必须使用函数原型。C++不再允许旧式风格的声明。
- Clean C程序必须避免使用C++关键字作为标识符，例如class和virtual。

另外还存在一些规则和区别，但它们通常不会造成问题。在本书中，我们解释如何编写能够被C++编译器所接受的标准C语言代码。我们并不讨论标准C中并不存在的那些C++特性（正是这些特性奠定了C和C++的区别）。

1.1.6 本书内容

本书描述了C语言的3种主要变型：传统C、C89和C99。书中说明了哪些特性是C89修正案1所增加的，并描述了C和C++的公共子集Clean C。我们还将解释如何编写“良好”的C程序，即具有良好的可读性、可移植性和可维护性的程序。

官方的“标准C”是C99。但是，我们使用标准C这个术语表示从C89沿用到C99的那些特性和概念。我们会特意说明那些仅在C99中存在的特性，以便使用C89编译器的程序员可以避免这些特性。

1.2 应该使用哪种C语言

应该使用哪种C语言取决于读者所使用的是哪种C编译器，以及代码在移植性方面有什么要求。可供的选择有：

1. C99。这是标准C的当前版本。它具有所有最新的特性，但有些编译器可能还不支持它