



C 和 C++ 实务精选



# C

# 陷阱与缺陷

# C Traps and Pit falls falls

Andrew Koenig 著  
高 巍 译  
王 昕 审校

人民邮电出版社  
POSTS & TELECOMMUNICATIONS PRESS

C和C++实务精选

# C 陷阱与缺陷

---

Andrew Koenig 著

高 巍 译

王 昕 审校

人民邮电出版社

C 和 C++ 实务精选  
C 陷阱与缺陷

---

- ◆ 著 Andrew Koenig  
译 高 巍  
审 校 王 昕  
责任编辑 陈冀康
- ◆ 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>  
读者热线 010-67132705  
北京汉魂图文设计有限公司制作  
北京朝阳展望印刷厂印刷  
新华书店总店北京发行所经销
- ◆ 开本: 720×980 1/16  
印张: 12  
字数: 224 千字 2002 年 11 月第 1 版  
印数: 1-5 000 册 2002 年 11 月北京第 1 次印刷

著作权合同登记 图字: 01 - 2002 - 2444 号

ISBN 7-115-10623-1/TP · 3080

---

定价: 30.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

## 图书在版编目 (CIP) 数据

C 陷阱与缺陷 / (美) 沃尔尼格 (Vornig, A.) 著; 高巍译.

—北京: 人民邮电出版社, 2002.11

(C 和 C++ 实务精选)

ISBN 7-115-10623-1

I. C... II. ①沃...②高... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 082203 号

## 版 权 声 明

Andrew Koenig: C Traps and Pitfalls

Copyright ©1989 by AT&T Bell Laboratories, Inc.

ISBN: 0201179288

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise without the prior consent of Addison-Wesley.

Published by arrangement with Addison Wesley Longman, Inc. All Rights Reserved.

版权所有。未经出版者书面许可, 对本书任何部分不得以任何方式或任何手段复制和传播  
人民邮电出版社经 Addison Wesley Longman 公司授权出版。版权所有, 侵权必究。

---

# 内 容 提 要

作者以自己 1985 年在 Bell 实验室时发表的一篇文章为基础，结合自己的工作  
经验扩展成为这本对 C 程序员具有珍贵价值的经典著作。写作本书的出发点不  
是要批判 C 语言，而是要帮助 C 程序员绕过编程过程中的陷阱和障碍。

全书分为 8 章，分别从词法分析、语法规义、连接、库函数、预处理器、可  
移植性缺陷等几个方面分析了 C 编程中可能遇到的问题。最后，作者用一章的篇  
幅给出了若干具有实用价值的建议。

本书适合有一定经验的 C 程序员阅读学习，即便你是 C 编程高手，本书也应  
该成为你的案头必备书籍。

---

# 作者简介

## Andrew Koenig

AT&T 大规模程序研发部（前贝尔实验室）成员。他从 1986 年开始从事 C 语言的研究，1977 年加入贝尔实验室。他编写了一些早期的类库，并在 1988 年组织召开了第一个相当规模的 C++ 会议。在 ISO/ANSI C++ 委员会成立的 1989 年，他就加入了该委员会，并一直担任项目编辑。他已经发表了 C++ 方面的 100 多篇论文，在 Addison-Wesley 出版了 *C Trap and Pitfalls*，《C 陷阱与缺陷》和 *Ruminations on C++*（《C++ 沉思录》，人民邮电出版社）还应邀到世界各地演讲。

Andrew Koenig 不仅有着多年的 C++ 开发、研究和教学经验，而且还亲身参与了 C++ 的演化和变革，对 C++ 的变化和发展起到重要的影响。



# Preface to the Chinese Edition

When I first wrote *C Traps and Pitfalls*, I never dreamed that it would still be in print 14 years later! I believe that the reason for this book's longevity is that it teaches some important lessons about C programming that are still not widely understood.

The aspects of C that invite mistakes are the same aspects that make it attractive for expert programmers. Accordingly, most people who set out to become C experts will make the same mistakes along the way—mistakes that will be there to be made as long as C continues to attract new programmers.

If you read a typical programming book, you will probably find that the author thinks that the most important part of becoming a good programmer is to learn as many details as possible about a particular language, library, or system. There is some truth in this notion of course, but it tells only part of the story. Details are easy to learn: All one needs is a reference book with a good index, and perhaps a more experienced colleague to point one in the right direction once in a while. It is much harder to understand the best ways of using what one already knows.

One way to gain such understanding is to learn what not to do. Programming languages, such as C, that are intended to be convenient for experts to use often invite misuse in ways that someone with enough experience can predict. By studying the mistakes that programmers make most often in such a language, one can not only avoid those mistakes, but one can also understand more deeply how the language works.

I am particularly pleased to learn about the Chinese translation of this book because the translation makes it available for the first time to such a large audience. If you read this book, I hope that it will help turn your frustration into happiness.

Andrew Koenig  
Gillette, New Jersey, USA  
October, 2002

# 中文版序

我动笔写作《C 缺陷与陷阱》时，可没想到 14 年后这本书仍然在印行！它之所以历久不衰，我想，可能是书中道出了 C 语言编程中一些重要的经验教训。就是到今天，这些教训也还没有广为人知。

C 语言中那些容易导致人犯错误的特性，往往也正是编程老手们为之吸引的特性。因此，大多数程序员在成长为 C 编程高手的道路上，犯过的错误真是惊人地相似！只要 C 语言还能继续感召新的程序员投身其中，这些错误就还会一犯再犯。

大家通常读到的程序设计书籍中，那些作者总是认为，要成为一个优秀的程序员，最重要的无非是学习一种特定程序语言、函数库或者操作系统的细节，而且多多益善。当然，这种观念不无道理，但也有偏颇之处。其实，掌握细节并不难，一本索引丰富完备的参考书就已经足矣；最多，可能还需要一位稍有经验的同事不时从旁点拨，指明方向。难的是那些我们已经了解的东西，如何“运用之妙，存乎一心”。

学习哪些是不应该做的，倒不失为一条领悟运用之道的路子。程序设计语言，就比如说 C 吧，其中那些让精于编程者觉得称心应手之处，也格外容易误用；而经验丰富的老手，甚至可以如有“先见之明”般，指出他们误用的方式。研究一种语言中程序员容易犯错之处，不但可以“前车之覆，后车之鉴”，还能使我们更谙熟这种语言的深层运作机制。

知悉本书中文版即出，将面对更为广大的中国读者，我尤为欣喜。如果您正在读这本书，我真挚地希望，它能对您有所裨益，为您释疑解惑，让您体会编程之乐。

Andrew Koenig

美国新泽西州吉列

2002 年 10 月



# 18 次印刷的奇迹

——经典 C 语言书籍 *C Traps and Pitfalls* 简介

如果有人问我，要想学好一门编程语言，应该阅读什么样的书籍呢？毫无疑问，在大多数场合下我都会向他推荐市面上最新出版的书籍。原因就是：以现在计算机领域内技术的发展速度，几乎是每隔一段时间，我们就需要对自己现有的知识进行更新，这样看来，使用一本比较新的书籍，里面的内容将会比较贴近当前技术的发展，因而也就能让你更容易掌握你所要学的东西。

但有一本讲述 C 语言的书，自出版以来，历经 14 载，它一直都被各个书评站点（或书评人）列入“重点推荐”的清单中。尤为夸张的是，14 年来，在它的 18 次印刷版本中，除去第二次印刷稍微修改过一些问题，以后的 16 次印刷，我们居然发现它的内容没有丝毫的变更!!! 对于技术书籍，我想其精确性与权威性也算是奇迹了吧。

这就是 Andrew Koenig 给我们带来的 *C Traps and Pitfalls*（中文书名：《C 陷阱与缺陷》）。在 C/C++ 领域中，Andy（Andrew 的呢称）的名字对于每个人来说绝对是如雷贯耳。作为一个知名的专栏作者，Andy（和他那位同样大名鼎鼎的妻子——Barbara Moo）已经在各类杂志上面发表了上百篇的杂志文章，给很多人在技术进步的道路上带来了极大的帮助。ACCU 的 Francis Glassborow 对他的评价是：Andy 是世界上最出色的几位 C++ 专家之一。

本书是 Andy 的第一本技术书籍，其原始素材来自于 Andy 在 1986 所提交的同名的技术报告。在书中，作者针对 C 程序在编译、链接的过程中可能碰到的种种问题以及编译、运行环境对程序可能带来的影响等，列出了许多值得我们注意的地方。按照作者本人的观点，前人碰到过的问题来现身说法，可以帮助你避免那些一而再，再而三出现在你的程序中的问题。由于是以实例来描述作者（以及他人）所碰到过的具体问题，因此，此书少去了许多空洞无味的说教，虽然本书篇幅不大（原书正文只有区区 147 页），但实际上，它的每个小节，每一段都蕴含

着作者（以及他人）大量的经验教训，都值得我们去仔细琢磨，经常温习。为此，Francis Glassborow 说到：从我了解 C 语言开始，我就将它时时放在手边，经常翻阅。作者自己也在书中毫不谦虚地说：“如果你是一个程序员在开发中经常用到 C 语言，这本书应该成为你案头必备书籍。即使你已经是一个 C 语言的专家级程序员，仍然有必要拥有一本”。事实上，Andy 并没有吹嘘，就书中所列出的种种问题，我本人也不止一次在自己的程序（也包括别人的程序）中发现它们的踪迹，而且有些问题出现得还极为频繁。这使我不禁就想到，要是我们能够早一些看到这本书上提及的问题，那岂不是可以省去很多开发时的除错时间……

可能有人会有疑问：从书名来看，它是一本讲述 C 语言的书籍，那么对于 C++ 的学习者来说，它是不是也同样有价值呢？另外，现在 C 语言的最新 ISO/ANSI 标准文档 C99 都已经制订出来了，而作为一本在 C89 之前出版的 C 语言书籍，它的作用是否还和以前一样大呢？答案是肯定的。本书英文版连续 18 次印刷的事实就是有力的证明。实际上，C++ 在那个层面上和 C 的区别并不大，在 C 程序中常犯的错误通常在 C++ 程序中也经常得以重现，因此，从这个角度来说，C 语言中的陷阱也常常就是 C++ 语言中的陷阱。此外，虽然最新的 C99 对于以前的 K & R C 有了一些变化，但在较低层次（如词法、语法）上，它们几乎是没有什么差别的。因此，对于本书中所有问题的讨论，几乎都可以适用于最新的 ISO/ANSI C。

现在，人民邮电出版社翻译出版 *C Traps and Pitfalls* 一书，无疑是献给 C 和 C++ 程序员的一份厚礼。我本人很荣幸能够担任此书的技术审校，为这本书中文版的出版尽一点绵薄的心力。感谢译者的辛勤劳动，也感谢出版社能够给我这样的机会！希望本书能够为你的学习带来一些帮助。

王昕

2002 年 8 月

对于经验丰富的行家而言，得心应手的工具在初学时的困难程度往往要超过那些容易上手的工具。刚刚接触飞机驾驶的学员，初航时总是谨小慎微，只敢沿着海岸线来回飞行，等他们稍有经验就会明白这样的飞行其实是一件多么轻松的事。初学骑自行车的新手，可能觉得后轮两侧的辅助轮很有帮助，但一旦熟练过后，就会发现它们很是碍手碍脚。

这种情况对程序设计语言也是一样。任何一种程序设计语言，总存在一些语言特性，很可能会给还没有完全熟悉它们的人带来麻烦。令人吃惊的是，这些特性虽然因程序设计语言的不同而异，但对于特定的一种语言，几乎每个程序员都是在同样的一些特性上犯过错误、吃过苦头！因此，作者也就萌生了将这些程序员易犯错误的特性加以收集、整理的最初念头。

我第一次尝试收集这类问题是在 1977 年。当时，在华盛顿特区举行的一次 SHARE（IBM 大型机用户组）会议上，我作了一次题为“PL/I 中的问题与‘陷阱’”的发言。作此发言时，我刚从哥伦比亚大学调至 AT&T 的贝尔实验室，在哥伦比亚大学我们主要的开发语言是 PL/I，而贝尔实验室中主要的开发语言却是 C。在贝尔实验室工作的 10 年间，我积累了丰富的经验，深谙 C 程序员（也包括我本人）在开发时如果一知半解将会遇到多少麻烦。

1985 年，我开始收集有关 C 语言的此类问题，并在年底将结果整理后作为一

篇内部论文发表。这篇论文所引发的回应却大大出乎我的意料，共有 2 000 多人向贝尔实验室的图书馆索取该论文的副本！我由此确信有必要将该论文的内容进一步扩充，于是就写成了现在读者所看到的这本书。

### 本书是什么

本书力图通过揭示一般程序员，甚至是经验老道的职业程序员，如何在编程中犯错误、摔跟头，以提倡和鼓励预防性的程序设计。这些错误实际上一旦被程序员真正认识和理解，并不难避免。因此，本书阐述的重点不是一般原则，而是一个个具体的例子。

如果你是一个程序员并且开发中真正用到 C 语言来解决复杂问题，这本书应该成为你的案头必备书籍。即使你已经是一个 C 语言的专家级程序员，仍然有必要拥有这本书，很多读过本书早期手稿的专业 C 程序员常常感叹：“就在上星期我还遇到这样一个 Bug！”如果你正在教授 C 语言课程，本书毫无疑问应该成为你向学生推荐的首选补充阅读材料。

### 本书不是什么

本书不是对 C 语言的批评。程序员无论使用何种程序设计语言，都有可能遇到麻烦。本书浓缩了作者长达 10 年的 C 语言开发经验，集中阐述了 C 语言中各种问题和“陷阱”，目的是希望程序员读者能够从中吸取我本人以及我所见过的其他人所犯错误的经验教训。

本书不是一本“烹饪菜谱”。我们不能希望可以通过详尽的指导说明来完全避免错误。如果可行的话，那么所有的交通事故都可以通过在路旁刷上“小心驾驶”的标语来杜绝。对一般人而言最有效的学习方式是从感性的、活生生的事例中学习，比如自己的亲身经历或者他人的经验教训。而且，哪怕只是明白了一种特定的错误是如何可能发生的，就已经在将来避免该错误的路上迈了一大步。

本书并不打算教你如何用 C 语言编程（见 Kernighan 和 Ritchie: *The C Programming Language*, 第 2 版, Prentice-Hall, 1988），也不是一本 C 语言参考手册（见 Harbison 和 Steele: *C: A Reference Manual*, 第 2 版, Prentice-Hall, 1987）。本书未提及数据结构与算法（见 Van Wyk: *Data Structures And C Programs*, Addison-Wesley, 1988），仅仅简略介绍了可移植性（见 Horton: *How To Write*

*Portable Programs In C*, Prentice-Hall, 1989) 和操作系统接口 (见 Kernighan 和 Pike: *The Unix Programming Environment*, Prentice-Hall, 1984)。本书中所涉及的问题均来自编程实践, 适当作了简化 (如果希望读到一些“挖空心思”设计出来, 专门让你绞尽脑汁的 C 语言难题, 见 Feuer: *The C Puzzle Book*, Prentice-Hall, 1982)。本书既不是一本字典也不是一本百科全书, 我力图使其精简短小, 以鼓励读者能够阅读全书。

### 读者的参与和贡献

可以肯定, 我遗漏了某些值得注意的问题。如果你发现了一个 C 语言问题而本书又未提及, 请通过 Addison-Wesley 出版社与我联系。在本书的下一版中, 我很有可能引用你的发现, 并且向你致谢。

### 关于 ANSI C

在我写作本书时, ANSI C 标准尚未最后定案。严格地说, 在 ANSI 委员会完成其工作之前, “ANSI C”的提法从技术上而言是不正确的。而实际上, ANSI 标准化工作大体已经尘埃落定, 本书中提及的有关 ANSI C 标准内容基本上不可能有所变动。很多 C 编译器甚至已经实现了大部分 ANSI 委员会所考虑的对 C 语言的许多重大改进。

毋需担心你使用的 C 编译器并不支持书中出现的 ANSI 标准函数语法, 它并不会妨碍你理解例子中真正重要的内容, 而且书中提及的程序员易犯错误其实与何种版本的 C 编译器并无太大关系。

### 致谢

本书中问题的收集整理工作绝非一人之力可以完成。以下诸位都向我指出过 C 语言中的特定问题, 他们是 Steve Bellovin (6.3 节), Mark Brader (1.1 节), Luca Cardelli (4.4 节), Larry Cipriani (2.3 节), Guy Harris and Steve Johnson (2.2 节), Phil Karn (2.2 节), Dave Kristol (7.5 节), George W. Leach (1.1 节), Doug McIlroy (2.3 节), Barbara Moo (7.2 节), Rob Pike (1.1 节), Jim Reeds (3.6 节), Dennis Ritchie (2.2 节), Janet Sirkis (5.2 节), Richard Stevens (2.5 节), Bjarne Stroustrup (2.3 节), Ephraim Vishnaic (1.4 节), 以及一位自愿要求隐去姓名者 (2.3 节)。为简短起见, 对于同一个问题此处仅仅列出了第一位向我指出该问题的人。我认

为这些错误绝不是凭空臆造出来的，而且即使是，我想也没有人愿意承认。至少这些错误我本人几乎都犯过，而且有的还不止犯一次。

在书稿编辑方面许多有用的建议来自 Steve Bellovin, Jim Coplien, Marc Donner, Jon Forrest, Brian Kernighan, Doug McIlroy, Barbara Moo, Rob Murray, Bob Richton, Dennis Ritchie, Jonathan Shapiro, 以及一些未透露姓名的审阅者。Lee McMahon 与 Ed Sitar 为我指出了早期手稿中的许多录入错误，使我避免了一旦成书后将要遇到的很多尴尬。Dave Prosser 为我指明了许多 ANSI C 中的细微之处。Brian Kernighan 提供了极有价值的排版工具和帮助。

与 Addison-Wesley 出版社合作是一件愉快的事情，感谢 Jim DeWolf, Mary Dyer, Lorraine Ferrier, Katherine Harutunian, Marshall Henrichs, Debbie Lafferty, Keith Wollman, 和 Helen Wythe。当然，他们也从一些并不为我所知的人们那里得到了帮助，使本书最终得以出版，我在此也一并致谢。

我需要特别感谢 AT&T 贝尔实验室的管理层，他们开明的态度和支持使得我以写作本书，包括 Steve Chappell, Bob Factor, Wayne Hunt, Rob Murray, Will Smith, Dan Stanzione 和 Eric Sumner。

本书书名受到 Robert Sheckley 的科幻小说选集的启发，其书名是 *The People Trap and Other Pitfalls, Snares, Devices and Delusions (as well as Two Sniggles and a Contrivance)* (1968 年由 Dell Books 出版)。

---

# 目 录

---

第0章 导读	1
第1章 词法“陷阱”	5
1.1 =不同于==	6
1.2 & 和   不同于&& 和	8
1.3 词法分析中的“贪心法”	8
1.4 整型常量	10
1.5 字符与字符串	11
第2章 语法“陷阱”	15
2.1 理解函数声明	15
2.2 运算符的优先级问题	19
2.3 注意作为语句结束标志的分号	24
2.4 switch 语句	26
2.5 函数调用	28
2.6 “悬挂”else 引发的问题	29

第 3 章 语义“陷阱” .....	33
3.1 指针与数组 .....	33
3.2 非数组的指针 .....	39
3.3 作为参数的数组声明 .....	41
3.4 避免“举隅法” .....	43
3.5 空指针并非空字符串 .....	44
3.6 边界计算与不对称边界 .....	45
3.7 求值顺序 .....	57
3.8 运算符&&、   和 ! .....	59
3.9 整数溢出 .....	61
3.10 为函数 main 提供返回值 .....	62
第 4 章 连接 .....	65
4.1 什么是连接器 .....	65
4.2 声明与定义 .....	67
4.3 命名冲突与 static 修饰符 .....	69
4.4 形参、实参与返回值 .....	70
4.5 检查外部类型 .....	77
4.6 头文件 .....	80
第 5 章 库函数 .....	83
5.1 返回整数的 getchar 函数 .....	84
5.2 更新顺序文件 .....	85
5.3 缓冲输出与内存分配 .....	86
5.4 使用 errno 检测错误 .....	88
5.5 库函数 signal .....	89
第 6 章 预处理器 .....	93



---

6.1	不能忽视宏定义中的空格.....	94
6.2	宏并不是函数.....	94
6.3	宏并不是语句.....	99
6.4	宏并不是类型定义.....	101
<b>第 7 章</b>	<b>可移植性缺陷.....</b>	<b>103</b>
7.1	应对 C 语言标准变更.....	104
7.2	标识符名称的限制.....	106
7.3	整数的大小.....	107
7.4	字符是有符号整数还是无符号整数.....	108
7.5	移位运算符.....	109
7.6	内存位置 0.....	110
7.7	除法运算时发生的截断.....	111
7.8	随机数的大小.....	113
7.9	大小写转换.....	113
7.10	首先释放，然后重新分配.....	115
7.11	可移植性问题的一个例子.....	116
<b>第 8 章</b>	<b>建议与答案.....</b>	<b>121</b>
8.1	建议.....	122
8.2	答案.....	126
附录 A	PRINTF, VARARGS 与 STDARG.....	145
附录 B	Koenig 和 Moo 夫妇访谈.....	167