C和C++ Practices

C和C++实务精选

PEARSON Addison Wesley

C和指针

POINTERS ON C

Kenneth A. Reek 著

徐 波 译

人民邮电出版社
POSTS & TELECOM PRESS

C和C++实务精选

C和指针

Kenneth A.Reek 著
徐 波 译

人民邮电出版社

C 和 C++实务精选

C 和指针

◆ 著　　Kenneth A.Reek
译　　徐　波
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
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：800×1000　1/16
印张：29.75
字数：779 千字　　2003 年 9 月第 1 版
印数：1-5 000 册　　2003 年 9 月北京第 1 次印刷

著作权合同登记　图字：01 - 2003 - 4079 号
ISBN 7-115-11456-0/TP • 3528

定价：55.00 元

图书在版编目（CIP）数据

C 和指针/（美）里克（Reek,K.A.）编著；徐波译. 北京：人民邮电出版社，2003.9
ISBN 7-115-11456-0

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

中国版本图书馆 CIP 数据核字（2003）第 056412 号

版 权 声 明

内容提要

本书提供与C语言编程相关的全面资源和深入讨论。本书通过对指针的基础知识和高级特性的探讨，帮助程序员把指针的强大功能融入到自己的程序中去。

全书共18章，覆盖了数据、语句、操作符和表达式、指针、函数、数组、字符串、结构和联合等几乎所有重要的C编程话题。书中给出了很多编程技巧和提示，每章后面有针对性很强的练习，附录部分则给出了部分练习的解答。

本书适合C语言初学者和初级C程序员阅读，也可作为计算机专业学生学习C语言的参考。

前言

为什么需要这本书

市面上已经有了许多优秀的讲述 C 语言的书籍，为什么我们还需要这一本呢？我在大学里教授 C 语言编程已有 10 个年头，但至今尚未发现一本书是按照我所喜欢的方式来讲述指针的。许多书籍用一章的篇幅专门讲述指针，而且往往出现在全书的后半部分。但是，仅仅描述指针的语法、并用一些简单的例子展示其用法是远远不够的。我在授课时，很早便开始讲授指针，而且在以后的授课过程中也经常讨论指针。我描述它们在各种不同的上下文环境中的有效用法，展示使用指针的编程惯用法(programming idiom)。我还讨论了一些相关的课题如编程效率和程序可维护性之间的权衡。指针是本书的线索所在，融会贯通于全书之中。

指针为什么如此重要？我的信念是：正是指针使 C 威力无穷。有些任务用其他语言也可以实现，但 C 能够更有效地实现；有些任务无法用其他语言实现，如直接访问硬件，但 C 却可以。要想成为一名优秀的 C 程序员，对指针有一个深入而完整的理解是先决条件。

然而，指针虽然很强大，与之相伴的风险却也不小。跟指甲锉相比，链锯可以更快地切割木材，但链锯更容易使你受伤，而且伤害常常来得极快，后果也非常严重。指针就像链锯一样，如果使用得当，它们可以简化算法的实现，并使其更富效率；如果使用不当，它们就会引起错误，导致细微而令人困惑的症状，并且极难发现原因。对指针只是略知一二便放手使用是件非常危险的事。如果那样的话，它给你带来的总是痛苦而不是欢乐。本书提供了你所需要的深入而完整的关于指针的知识，足以使你避开指针可能带来的痛苦。

为什么要学习 C 语言

为什么 C 语言依然如此流行？历史上，由于种种原因，业界选择了 C，其中最主要的原因就在于它的效率。优秀 C 程序的效率几乎和汇编语言程序一样高，但 C 程序明显比汇编语言程序更易于开发。和许多其他语言相比，C 给予程序员更多的控制权，如控制数据的存储位置和初始化过程等。C 缺乏“安全网”特性，这虽有助于提高它的效率，但也增加了出错的可能性。例如，C 对数组下标引用和指针访问并不进行有效性检查，这可以节省时间，但你在使用这些特性时就必须特别小心。如果你在使用 C 语言时能够严格遵守相关规定，就可以避免这些潜在的问题。

C 提供了丰富的操作符集合，它们可以让程序员有效地执行一些底层的计算如移位和屏蔽等，而不必求助汇编语言。C 的这个特点使很多人把 C 称为“高层”的汇编语言。但是，当需要的时候，C 程序可以很方便地提供汇编语言的接口。这些特性使 C 成为实现操作系统和嵌入性控制器软件的良好选择。

C 流行的另一个原因是由于它的普遍存在。C 编译器在许多机器上实现。另外，ANSI 标准提高了 C 程序在不同机器之间的可移植性。

最后，C 是 C++的基础。C++提供了一种和 C 不同的程序设计和实现的观点。然而，如果你对 C 的知识和技巧，如指针和标准库等成竹在胸，将非常有助于你成为一名优秀的 C++程序员。

为什么应该阅读这本书

本书并不是一本关于编程的入门图书。它所面向的读者应该已经具备了一些编程经验，或者是一些想学习 C，但又不想被诸如为什么循环很重要以及何时需要使用 if 语句等肤浅问题耽误进程的人。

另一方面，我并不要求本书的读者以前学习过 C。我讲述了 C 语言所有方面的内容。这种内容的广泛覆盖性使本书不仅适用于学生，也适用于专业人员。也就是说，适用于首次学习 C 的读者和那些经验更丰富的希望进一步提高语言使用技巧的用户。

优秀的 C++书籍把精力集中于与面向对象模型有关的课题上（如类的设计）而不是专注于基本的 C 技巧，这样做是对的。但 C++是建立在 C 的基础之上的，C 的基本技巧依然非常重要，特别是那些能够实现可复用类的技巧。诚然，C++程序员在阅读本书时可以跳过一些他们所熟悉的内容，但他们会在本书中找到许多有用的 C 工具和技巧。

本书的组织形式

本书按照教程的形式组织，它所面向的读者是先前具有编程经验的人。它的编写风格类似于导师在你的身后注视着你的工作，不时给你一些提示和忠告。我的目标是把通常需要多年实践才能获得的知识和观点传授给读者。这种组织形式也影响到材料的顺序——我通常在一个地方引入一个话题，并进行完整的讲解。因此，本书也可以当作参考手册。

在这种组织形式中，存在两个显著的例外之处。首先是指针，它贯穿全书，将在许多不同的上下文环境中进行讨论。其次就是第 1 章，它对语言的基础知识提供了一个快速的介绍。这种介绍有助于你很快掌握编写简单程序的技巧。第 1 章所涉及的主题将在后续章节中深入讲解。

较之其他书籍，本书在许多领域着墨更多，主要是为了让每个主题更具深度，向读者传授通常只有实践才能获得的经验。另外，我使用了一些在现实编程中不太常见的例子，虽然有些不太容易理解，但这些例子显示了 C 在某些方面的趣味所在。

ANSI C

本书描述 ANSI C，是由 ANSI/ISO 9899-1990[ANSI 90]进行定义并由[KERN 89]进行描述的。我之所以选择这个版本的 C 是基于两个原因：首先，它是旧式 C（有时称作 Kernighan 和 Ritchie[KERN 78]，或称 K&R C）的后继者，并已在根本上取代了后者；其次，ANSI C 是 C++的基础。本书中的所有例子都是用 ANSI C 编写的。我常常把“ANSI C 标准文档”简称为“标准”。

排版说明

语法描述格式如下

```
if( expression )
        statement
else
        statement
```

我在语法描述中使用了 4 种字体，其中必须的代码（如此例中的关键字 if）将如上所示设置为 Courier New 字体。必要代码的抽象描述（如上例中的 *expression*）用 *Courier New* 表示。有些语句具有可选部分，如果我决定使用可选部分（如此例中的 **else** 关键字），它将严格按上面的例子以**粗体 Courier New** 表示。可选部分的抽象描述（如第二个 ***statement***）将以***粗斜体 Courier New*** 表示。每次引入新术语时，我将以**黑体**表示。

完整的程序将标上号码，以“程序 0.1”这样的格式显示。标题给出了程序的名称，包含

源代码的文件名则显示在右下角——这些文件都可以从 Addison Wesley Longman 的网站上找到。

文中有“**提示**”部分。这些提示中的许多内容都是对良好编程技巧的讨论——就是使程序更易编写、更易阅读并在以后更易理解。当一个程序初次写成时，稍微多做些努力就可能节约以后修改程序的大量时间。其他一些提示能帮助你把代码写得更加紧凑或更有效率。

另外还有一些提示涉及软件工程的话题。C 的诞生远早于现代软件工程原则的形成。因此，有些语言特性和通用技巧不为这些原则所提倡。这些话题通常涉及到某种特定结构的效率和代码的可读性和可维护性之间的利弊权衡。这方面的讨论将向你提供一些背景知识，帮助你判断效率上的收益是否抵得上其他质量上的损失。

当你看到“**警告**”时就要特别小心：我将要指出的是 C 程序员新手（有时甚至是老手）经常出现的错误之一，或者代码将不会如你所预想的那样运行。这个警告标志将使提示内容不易被忘记，而且以后回过头来寻找也更容易一些。

“K&R C”表示我正在讨论 ANSI C 和 K&R C 之间的重要区别。尽管绝大多数以 K&R C 写成的程序仅需极微小的修改即可在 ANSI C 环境运行，但有时你仍可能碰到一个 ANSI 之前的编译器，或者遇到一个更老式的程序。如此一来，两者的区别便至关重要。

每章问题和编程练习

本书每章的最后一节是问题和编程练习。问题难简不一，从简单的语法问题到更为复杂的问题诸如效率和可维护性之间的权衡等。编程练习按等级区分难度：★ 的练习最为简单，★★★★★ 的练习难度最大。这些练习有许多作为课堂测验已沿用多年。问题或编程练习前如果有一个 ✍ 符号，表示在附录中可以找到它的参考答案。

补充材料

Addison Wesley Longman 专门为本书维护了一个 World Wide Web 站点。该站点的 URL 是 http://www.awl.com/cseng/titles/0-673-99986-6/（或可直接访问作者主页 www.cs.rit.edu/~kar/）。这个站点包含本书所有程序的源代码，以章为单位分类。你还可以在上面看到本书的最新勘误表。你还可以联系附近的 Addison Wesley Longman 代表，获取 *Instructor's Guide*，它包含了书上未给出答案的问题和编程练习的所有答案。

如果你是一位教育工作者，也可以免费获取 UNIX 系统上自动递交和测试学生程序的软件[REEK 89, REEK 96]，通过匿名 FTP：ftp.cs.rit.edu，目录是 pub/kar/try。

致谢

我无法列出所有对本书做出贡献的人们，但我将感谢他们中的所有人。我的妻子 Margaret 对我的写作鼓励有加，为我提供精神上的支持，而且她默默承受着由于我写作本书而带给她的生活上的孤独。

我要感谢 Warren Caithers 教授，他是我在 RIT 的同事，阅读并审校了本书的初稿。他真诚的批评帮助我从一大堆讲课稿和例子中生成了一份清晰、连贯的手稿。

我非常感谢我的 C 语言编程课程的学生们，他们帮助我发现录入错误，提出改进意见，并在教学过程中忍受着草稿形式的教材。你们对我的作品的反应向我提供了有益的反馈，帮助我进一步改进本书的质量。

我还要感谢 Steve Allan, Bill Appelbe, Richard C.Detmer, Roger Eggen, Joanne Goldenberg, Dan Hinton, Dan Hirschberg, Keith E.Jolly, Joseph F.Kent, Masoud Milani, Steve Summit 和 Kanupriya Tewary，他们在本书出版前对它作了评价。他们的建议和观点对我进一步改进本书的表达形式助益颇多。

最后，我要向我在 Addison-Wesley 的编辑 Deborah Lafferty 女士、产品编辑 Amy Willcutt 女士表示感谢。正是由于她们的帮助，才使这本书从一本书手稿成为一本正式的书籍。她们不仅给了我很多有价值的建议，而且鼓励我改进我原先自我感觉良好的排版。现在我已经看到了结果，她们的意见是正确的。

现在是开始学习的时候了，我预祝大家在学习 C 语言的过程中找到快乐！

Kenneth A. Reek

kar@cs.rit.edu

Churchville, 纽约

目录