TURING 图灵程序设计丛书

Understanding and Using C Pointers

C/C++程序员进阶必备，透彻理解指针与内存管理

深入理解C指针

[美] *Richard Reese* 著

陈晓亮 译

O'REILLY®

人民邮电出版社

POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

深入理解C指针

Understanding and Using C Pointers

[美] Richard Reese 著
陈晓亮 译

O'REILLY®
Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo
O'Reilly Media, Inc.授权人民邮电出版社出版

人民邮电出版社
北　京

图书在版编目（CIP）数据

深入理解C指针 /（美）里斯（Reese,R.）著 ；陈晓亮译. -- 北京 ：人民邮电出版社，2014.2
（图灵程序设计丛书）
书名原文：Understanding and using C pointers
ISBN 978-7-115-34448-9

Ⅰ. ①深… Ⅱ. ①里… ②陈… Ⅲ. ①C语言－程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2014)第008761号

内 容 提 要

指针是C语言的一项核心特性，对于指针的掌握程度是区分C语言新手与老手的重要标志。本书专门研究指针，旨在提供比其他图书更全面和深入的C指针和内存管理知识。全书巨细靡遗地为读者展现了C语言编程中最重要的话题：C的动态内存管理、指针和函数、指针和数组、指针和字符串、指针和结构体。作者还专门用一章篇幅讲解了安全问题和指针误用，让读者对指针的认识又深入一层。全书每章都穿插了使用指针的注意事项和潜在陷阱，及时贴心地提示读者在编程中避开此类问题。

本书适合C和C++程序员和开发人员阅读，也可作为计算机专业学生学习C语言的参考图书。

◆ 著 [美] Richard Reese
译 陈晓亮
责任编辑 刘美英
责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京天宇星印刷厂印刷

◆ 开本：800×1000 1/16
印张：12.75
字数：243千字 2014年2月第1版
印数：1－4 000册 2014年2月北京第1次印刷
著作权合同登记号 图字：01-2014-0409号

定价：45.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第0021号

版权声明

O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 Make 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar 博客有口皆碑。”

——Wired

“O'Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——CRN

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去 Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

前言

C 是一门重要的编程语言，一直以来备受关注。C 语言的核心是指针，这门语言的灵活性和超长之处很大一部分都源于指针。指针提供了动态操控内存的机制，强化了对数据结构的支持，且实现了访问硬件的功能。不过，指针的这种能力和灵活性是有代价的，它很难掌握。

本书的不同之处

市面上已经出版了大量关于 C 的图书，这些书一般会讲到语言的方方面面，但是对于指针的讲解却翻来覆去总是那点东西，很少涉及指针基本知识以外的内容。这些书大部分只是粗略地提一下内存管理，其实内存管理技术很重要，会牵涉栈和堆。不讨论这些内容，读者就只能粗浅地理解指针。栈和堆是分别用来支持函数和动态内存分配的内存区域。

指针非常复杂，值得我们深入讨论。本书专门研究指针，旨在让读者对指针的理解更加深入。这种理解在某种程度上需要读者会使用程序栈和堆，且会在相关上下文背景下使用指针。人们对不同领域知识的理解程度不同，有大体粗略了解，也有深入透彻的理解。要想对 C 语言的理解达到更高的层次，就必须有坚实的指针和内存管理基础。

学习方法

编程其实就是操控数据，数据一般位于内存中。因此如果能更好地理解 C 如何管理内存，就能对程序的工作原理洞若观火，从而使编程能力更上一层楼。知道 `malloc` 是从堆上分配内存是一回事，理解内存分配意味着什么则是另一回事。如果我们分配一个逻辑长度是 45 字节的结构体，可能会惊讶地发现实际上分配的空间

往往大于 45 字节，而且分配的内存可能会碎片化。

调用函数过程中系统会创建一个栈帧并将其推到程序栈上。理解栈帧和程序栈就可以弄清楚传值和传址的概念。尽管栈帧和指针没有直接联系，但是理解栈帧有助于理解递归的工作原理。

为方便理解指针和内存管理技术，我们会讲到不同的内存模型，有简单的线性内存布局，也有更复杂的针对某个特定示例说明程序栈和堆状态的图例。显示在屏幕上或是印刷在书中的代码只是动态程序的一种静态表示，这种表示的抽象本质是理解程序行为最大的绊脚石。内存模型对清除这些绊脚石非常有用。

目标读者

C 语言是块结构的语言，其过程式编程方法在 C++ 和 Java 等很多现代语言中也被采用。这些语言都要用到程序栈和堆，也都会用到指针，只不过通常用引用来包装。阅读本书要求你对 C 有最基本的了解。如果你正在学 C，那么本书能为你提供比其他书更全面的指针和内存的知识，能扩大你的 C 知识面，也能让你发现自己在 C 上的薄弱环节。如果你是有经验的 C 或者 C++ 程序员，本书能帮你填补对 C 语言理解的空白，也能强化你对这两门语言工作原理的理解，从而让你成长为更优秀的程序员。如果你是 C# 或者 Java 开发者，本书能让你更好地理解 C，也能让你更加透彻地认识面向对象语言如何处理栈和堆。

本书结构

本书按照数组、结构体、函数等传统主题组织。不过，所有章节关注的焦点都在于指针的使用和内存管理。比如说，我们讲到了如何给函数传递指针和从函数返回指针，也讲到了指针在栈帧中的使用以及指针如何引用堆中的内存。

- 第1章，认识指针
  这一章为非专业人士和对指针比较陌生的读者介绍指针的基础知识，包括指针操作符和如何声明不同类型的指针（比如常量指针、函数指针），以及如何使用 NULL 及相关变体。这对内存的分配和使用方式有很大的影响。

- 第2章，C的动态内存管理
  主要介绍标准的内存分配函数和回收内存的相关技术。能否及时回收内存对大部分应用程序来说都至关重要，做不到这一点就会导致内存泄漏和迷途指针。我们也讲到了垃圾收集和异常处理函数等回收技术。

- 第3章，指针和函数
  函数是组成应用程序代码的基石，不过给函数传递数据和从函数返回数据可能让开发新手迷糊。这一章介绍传递数据的技术，以及用指针返回数据时的常见陷阱。另外还会详细讲解函数指针，这种指针为应用程序提供了另一层控制和灵活性。

- 第4章，指针和数组
  尽管指针表示法和数组表示法不能完全互换，但两者紧密相关。这一章介绍一维和多维数组及其与指针的配合使用，还特别针对不同的内存模型解释如何传递数组，以及用连续或是非连续的方式动态分配数组时的一些常见问题。

- 第5章，指针和字符串
  字符串是很多应用程序的重要组成部分，这一章介绍字符串的基本知识，以及如何用指针操控字符串。字面量池及其对指针的影响也是很多人忽视的一个C特性。这一章会用丰富的图例来解释和说明这个主题。

- 第6章，指针和结构体
  结构体提供了一种排序和操控数据的有用方式，指针则通过在结构体的创建方式上提供更多的灵活性进一步强化了结构体的用途。这一章介绍结构体的基本知识及其与内存分配和指针的关系，接着举例说明如何在几种数据结构中使用结构体。

- 第7章，安全问题和指针误用
  指针很强大，但也可能造成很多安全问题。这一章研究缓冲区溢出的基本问题以及相关的指针问题，也会讲到避免这类问题的技术。

- 第8章，其他重要内容
  最后一章介绍一些比较零散但重要的指针技术和问题。尽管C不是面向对象语言，但是很多面向对象的编程思想可以融入C程序，包括多态。还会讲到在多线程环境中使用指针的要点，以及 `restrict` 关键字的意义和用法。

小结

本书旨在为读者提供比其他相关图书更深入的关于指针用法的知识，展示从指针的核心用法到罕见用法的例子，同时也指出常见的指针问题。

排版约定

本书使用的排版约定如下。

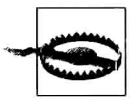
- 楷体

  表示新术语。

- 等宽字体

  表示程序代码，也用于正文中表示程序中使用的变量、函数名、命令行代码、数据类型、环境变量、语句和关键字等代码文本。

这个图标表示提示、建议或是一般注意事项。



这个图标表示警告。

使用代码

本书就是要帮读者解决实际问题的。也许你需要在自己的程序或文档中用到本书中的代码。除非大段大段地使用，否则不必与我们联系取得授权。因此，用本书中的几段代码写成一个程序不用向我们申请许可。但是销售或者分发 O'Reilly 图书随附的代码光盘则必须事先获得授权。引用书中的代码来回答问题也无需我们授权。将大段的示例代码整合到你自己的产品文档中则必须经过许可。

使用我们的代码时，希望你能标明它的出处。出处一般要包含书名、作者、出版商和书号，例如：*Understanding and Using C Pointers* (O'Reilly). Copyright 2013 Richard Reese, Ph.D. 978-1-449-34418-4。

如果还有其他使用代码的情形需要与我们沟通，可以随时与我们联系：permissions@oreilly.com。

Safari® Books Online

Safari Books Online（www.safaribooksonline.com）是应需而变的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品。

Safari Books Online 是技术专家、软件开发人员、Web 设计师、商务人士和创意人士开展调研、解决问题、学习和认证培训的第一手资料。

对于组织团体、政府机构和个人，Safari Books Online 提供各种产品组合和灵活的定价策略。用户可通过一个功能完备的数据库检索系统访问 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 以及其他几十家出版社的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息，我们网上见。

联系我们

请把对本书的评价和发现的问题发给出版社。

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）
奥莱利技术咨询（北京）有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息。本书的网站地址是：

http://oreil.ly/Understand_Use_CPointers

对于本书的评论和技术性问题，请发送电子邮件到：

bookquestions@oreilly.com

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

http://www.oreilly.com

我们在 Facebook 的地址如下：

http://facebook.com/oreilly

请关注我们的 Twitter 动态：

http://twitter.com/oreillymedia

我们的 YouTube 视频地址如下：

http://www.youtube.com/oreillymedia

目录

第1章

认识指针

C 程序员新手和老手的一大差别就在于是否对指针有深刻理解，能否高效利用指针。指针在 C 语言中随处可见，也提供了极大的灵活性。指针为动态内存分配提供了重要支持，与数组表示法紧密相关，指向函数的指针也为程序中的流控制提供了更多的选择。

一直以来，指针都是学习 C 语言的最大障碍。指针的基本概念很简单，就是一个存放内存地址的变量。然而，当我们开始应用指针操作符并试图看懂那些令人眼花缭乱的符号时，指针就开始变得复杂了。但情况并非总是如此，如果我们从简单的知识入手，打好扎实的基础，那么掌握指针的高级应用并不难。

理解指针的关键在于理解 C 程序如何管理内存。归根结底，指针包含的就是内存地址。不理解组织和管理内存的方式，就很难理解指针的工作方式。为此，只要对解释指针的原理有帮助，我们就会说明内存的组织方式。牢牢掌握了内存及其组织方式，理解指针就会容易很多。

本章简要介绍指针、指针操作符以及指针如何与内存相互作用。1.1 节研究如何声明指针、基本的指针操作符和 `null` 的概念。C 支持好几种不同类型的 `null`，所以仔细研究 `null` 会对我们有所启发。

1.2 节将细致地介绍几种不同的内存模型。毫无疑问，我们在使用 C 的过程中肯定会遇到各种内存模型。特定编译器和操作系统下的内存模型会影响指针的使用方式。我们也将仔细研究跟指针和内存模型有关的几种预定义类型。

1.3 节会深入探讨指针操作符，包括指针的算术运算和比较。1.4 节探究常量和指针。众多的声明组合提供了有趣通常也很有用的方法。

无论你是 C 程序员新手还是老手，本书都能帮助你深入理解指针，填补你知识结构中的空白。老手可以挑选感兴趣的主题，新手还是按部就班为好。

1.1 指针和内存

C 程序在编译后，会以三种形式使用内存。

- *静态/全局内存*
  静态声明的变量分配在这里，全局变量也使用这部分内存。这些变量在程序开始运行时分配，直到程序终止才消失。所有函数都能访问全局变量，静态变量的作用域则局限在定义它们的函数内部。

- *自动内存*
  这些变量在函数内部声明，并且在函数被调用时才创建。它们的作用域局限于函数内部，而且生命周期限制在函数的执行时间内。

- *动态内存*
  内存分配在堆上，可以根据需要释放，而且直到释放才消失。指针引用分配的内存，作用域局限于引用内存的指针，这是第 2 章的重点。

表 1-1 总结了这些内存区域中用到的变量的作用域和生命周期。

表1-1：不同内存中变量的作用域和生命周期

	作 用 域	生命周期
全局内存	整个文件	应用程序的生命周期
静态内存	声明它的函数内部	应用程序的生命周期
自动内存（局部内存）	声明它的函数内部	限制在函数执行时间内
动态内存	由引用该内存的指针决定	直到内存释放

理解这些内存类型可以更好地理解指针。大部分指针用来操作内存中的数据，因此理解内存的分区和组织方式有助于我们弄清楚指针如何操作内存。

指针变量包含内存中别的变量、对象或函数的地址。对象就是内存分配函数（比如 `malloc`）分配的内存。指针通常根据所指的数据类型来声明。对象可以是任何 C 数据类型，如整数、字符、字符串或结构体。然而，指针本身并没有包含所引用数据的类型信息，指针只包含地址。