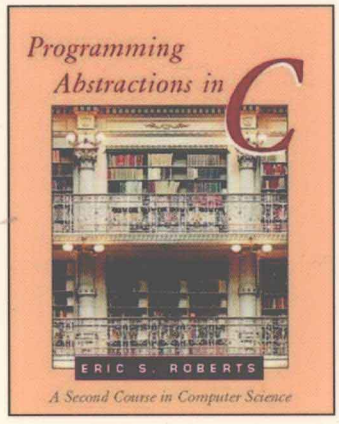
HZ BOOKS
华章科技

PEARSON

C语言经典译丛

C程序设计的抽象思维



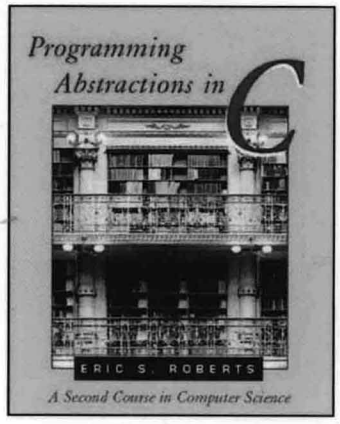
Programming Abstractions in C

(美) Eric S. Roberts 著
闪四清 译

机械工业出版社
China Machine Press

HZ BOOKS
华章科技
PEARSON
C语言经典译丛

C程序设计的抽象思维



Programming Abstractions in C

(美) Eric S. Roberts 著
闪四清 译

机械工业出版社
China Machine Press

本书是一本关于C语言的经典图书。本书共计17章，分为4部分，第一部分概述计算机导论课程中涉及的基本编程概念；第二部分讨论递归算法，其中结合大量示例，有助于读者轻松理解和掌握晦涩的概念；第三部分不仅介绍了用非递归算法实现的抽象数据类型，还提供了一些工具，有助于读者理解数据抽象的概念；第四部分重点介绍采用递归算法实现的抽象数据类型。本书重点突出，全面讲解了C语言的基本概念，深入剖析了具体的编程思路，揭示了独特的编程策略和技术细节。本书旨在通过介绍编程过程中遇到的难点和问题，来拓宽视野。本书结合具体的示例代码，由浅入深，介绍解决编程问题的策略和方法，有助于读者快速入门C语言编程。同时，每一章后面都有配套的复习题和编程练习，便于读者理论练习实践，通过编程实践查漏补缺，温故而知新。

本书适合希望学习C语言的初学者和中高级程序员阅读。

本书版权登记号：图字：01-2011-2882

图书在版编目（CIP）数据

C程序设计的抽象思维／（美）罗伯特（Roberts, E. S.）著；闪四清译. —北京：机械工业出版社，2012.4

（C语言经典译丛）

书名原文：Programming Abstractions in C

ISBN 978-7-111-38074-0

Ⅰ. C…　Ⅱ. ①罗…　②闪…　Ⅲ. C语言－程序设计　Ⅳ. TP312

中国版本图书馆CIP数据核字（2012）第071221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：谢晓芳

北京市荣盛彩色印刷有限公司印刷

2012年5月第1版第1次印刷

186mm×240mm · 39.25印张

标准书号：ISBN 978-7-111-38074-0

定价：99.00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

写给学生

每一年，计算机领域都会取得许多令人瞩目的发展。计算机硬件变得比以前更小、更快、更便宜。在计算机商店里的货架上，摆满了各种各样的、十年前根本无法想象的应用软件。像互联网和万维网这样的技术创新正在改变着人们获取信息、商业交易、与人沟通的方式。掌握计算机技术的人将拥有无穷的机遇。

计算机科学的学习有相似之处。你所学到的每一个新概念，都会使得编程变得越来越让人兴奋。你将具有更多的创造性，可以解决更难的问题，能够开发更加精密复杂的软件。本书适合那些学习了计算机科学导论，并对编程有一定了解的初学者。

大部分的入门课程都以编程技术为中心：学习某种语言的语法，并使用这种语言编写简单的程序。本书旨在通过介绍编程过程中遇到的难点和问题，来拓宽视野。编程不是记忆语法规则和写一些简单的过程代码，而是解决难题。在大多数情况下，这需要经过大量的思考和做大量的工作。

但是，通过本书介绍的策略和方法，读者可以简化这个学习过程。在学习本书中介绍的概念时，将从策略的角度（例如递归算法）和技术细节的角度（例如散列[⊖]技术）进行介绍，这样将使得读者能够解决现在看来似乎不能解决的问题。当然，学习这些概念是一项挑战性的工作。有时，对于某些概念，读者可能迷惑不解。但是，如果克服困难，真正地理解了这些概念，那么读者将会更加深刻地理解计算机的强大功能。

祝学业顺利。

⊖ 也称为“哈希”。

写给教师

本教科书适用于大学第二门编程课程，覆盖了 AMC *Curriculum* '78 报告中所定义的计算机科学 2 标准课程的材料，并且包括 *Computing Curriculum* 1991 学科领域中“算法与数据结构”课的大部分知识。

本书将教会学生现代软件工程方法论。本书的内容建立在我于 1995 年写的《C 语言的艺术与科学》[⊖]教科书的基础上，并将抽象和接口设计作为核心主题。这两本书都用到了一些通用库集，从而大大简化了代码的编写工作，使得初学者更加容易学习和掌握。这些通用库集已经被斯坦福大学和其他学校的学生们证明是非常成功的。然而，之所以有这些成功，是由于这些学校的教师和其他工作人员向学生提供了这些通用库集。这些库的代码可以从 Addison-Wesley 上用 FTP 下载，其网址如下：

```
ftp://aw.com/aw.computer.science/Roberts.CS1.C
```

虽然我写作这两本书是有先后顺序的，但是读者完全可以单独使用本书。本书第一部分包括《C 语言的艺术与科学》课本中学生应该掌握的所有背景知识。这些背景知识对于理解本课程其他部分中的例子和方法已经绰绰有余了。由于第一部分的介绍是比较快的，因此学生必须熟悉计算机导论课程中涉及的基本编程概念。但是，读者不需要对 C 语言有所了解，因为本书的前几章将介绍 C 语言的基础。学习过《C 语言的艺术与科学》课程的学生完全可以跳过第一部分的内容。

在学习完了第一部分的预备知识之后，学生可以继续该课程的学习。第二部分将讨论递归算法。在第二部分的 4 章内容中，穿插了大量的实例。根据我个人的经验，介绍递归算法的最佳时刻是在第二门编程课程开始学习的时候。很多学生都会觉得递归是一个难以理解的概念，必须花很多时间才能较好地掌握它。如果在新学期的一开始就面临递归这个难点，那么学生将有更多的时间来掌握它。在本书中，尽可能早地介绍递归概念，其目的是让读者在作业和考试中运用这种知识。期中考试可以检查学生对递归概念的掌握情况，对于那些不完全理解递归概念的学生，可以进行警示，以便他们及时采取相应的补救措施。

如果想压缩学习递归的时间，那么可以跳过第二部分的 6.1 节，这对整个课程的讲述没有什么影响。也许最小最大算法对于部分学生来说有点儿太复杂了，但是它很好地说明递归算法可以使用很少的代码来解决非常困难的问题。类似地，第 7 章中大 O 的理论基础也不是该课程的重点内容。

第三部分有双重的目的。一方面，介绍了数据结构课程中涉及的非递归算法的概念，包括

⊖ 由机械工业出版社引进并出版，英文书名为《The Art and Science of C》。——编辑注

栈、队列以及符号表；另一方面，该部分为学生提供了一些工具，从而帮助学生理解其他部分中涉及的基于接口编程的数据抽象概念。与这个概念相一致的是抽象数据类型（Abstract Data Type，ADT），它是由行为而不是由表现形式定义。

本书的一个重要特点是，不完全使用 ANSI C 的工具来定义 ADT，其中 ADT 的内部表示对于客户端来说是不可访问的。由于这样的编程风格强调了抽象的难度，因此将培养学生编写结构良好的程序和模块的习惯。我认为在本书中学习的接口是个实用的工具。在许多情况下，学生可以在他们自己的代码中包含和实现这些接口。

第三部分的最后一章（第 11 章）将介绍几个重要的概念，例如函数指针、映射函数以及迭代器。相对来说，迭代器在斯坦福的课程中是新近加入的，但是教学效果相当好。根据我们的经验，减少客户代码的复杂性所带来的收益远远超过建立迭代器抽象所花的工作。

第三和第四部分的重点是抽象数据类型。在某种程度上，这是人为划分的结果。这两部分的不同之处在于，第四部分中的 ADT 是用递归实现的，而第三部分则不是。这样安排的好处是第四部分在本书中起到综合的作用，将前两部分的递归和 ADT 进行综合。

尽管跳过第 14 章中关于表达式树的内容并不影响连贯性，但是我发现尽早地在课程中包括这些内容是很有价值的，因为这样可以减少对 C 语言编译器操作的神秘感，可以帮助学生更好地理解和控制程序。

第 17 章不是本课程的主要内容，而是为学生继续接下来的学习做准备。该章主要介绍面向对象编程，并且使用 Java 语言讲述主要的概念。尽管有些学校把 Java 作为导引课程，但是我们认为，由于下列一些原因，先介绍结构化编程方法再介绍面向对象编程方法是很有意义的：

1）Java 环境的变化太快了，无法为教学提供稳固的基础；

2）学生有必要理解结构化编程方法；

3）如果在入门课程中强调数据抽象和接口，那么学生学习面向对象编程将更加容易。

在斯坦福的经验给我们的启示是，这种策略很有效，能够使学生相对容易地接受面向对象的概念。

致　谢

仅仅一个人是难以成就一本书的。在完成本书的过程中，我很幸运地得到了许多人的积极帮助。我想感谢斯坦福的以下各位同事，他们以各自不同的方式给予了我帮助：

- 过去几年一直以该书的草稿进行教学的讲师，包括 Jerry Cain、Maggie Johnson、Bob Plummer、Mehran Sahami 以及 Julie Zelenski。
- 我的助教，Stacey Doerr 和 Brain O'Connor，他们为本书精选了作业题，很多作业作为练习题出现在本书中。
- 所有本科部的教师，他们帮助我向学生解释了我在较早版本的手稿中没有包含的概念。
- Steve Freund 和 Thetis 开发团队的成员，他们提供了一个优秀的计算机环境供学生使用。
- 教育部的工作人员，特别是 Claire Stager 和 Eddie Wallace，保证了一切都进展顺利。
- 我的关于计算机科学入门教学研讨会的各位参与者。
- 斯坦福学习该课程的学生，他们完成了很多实验。

非常感谢 Addison-Welsey 的各位书评员提出的关于改进本书结构和质量的建议：

- Phillip Barry，明尼苏达大学
- Martin Cohn，布兰代斯大学
- Dan Ellard，哈佛大学
- Gopal Gupta，新墨西哥大学
- Phillip W. Hutto，埃默里大学
- Randall Pruim，波士顿大学
- Zhong Shao，耶鲁大学

我同时也收到了里德学院的 Joe Buhler、普雷斯瓦勒公司的 Pavel Curtis、巴尔的摩马里兰大学的 Jim Mayfield 很好的建议。本书的大部分工作是我在里德休假时完成的，非常感谢数学系 Joe 和他的同事，他们为我提供了十分适合工作的地方。

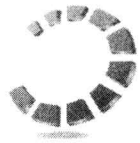
我想感谢我的编辑 Susan Hartman 和 Addison-Welsey 的所有工作人员——Cynthia Benn、Lynne Doran Cote、Jackie Davies、Julie Dunn、Peter Gordon、Amy Willcutt、Bob Woodbury 和 Tom Ziolkowski，感谢他们对本书和以前我出版的图书的支持。

最后，我要感谢我的搭档 Lauren Rusk，她又一次发挥了她作为我的开发编辑的魅力。Lauren 的专业知识使本书条理更为清晰，使本书增色不少。没有她，就不可能有这么好的一本书诞生。

目　录

第四部分 递归数据

第一部分
预 备 知 识

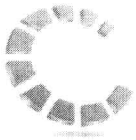


概述

第一部分介绍 ANSI C 的编程基础和基于接口的设计的概念。如果你是在学过其他语言之后首次接触 C 语言，那么这些章节中的内容将为你提供充足的背景知识，以理解本书其他部分中的程序。如果你已经使用过 C 语言编程，那么这些内容只是对所学知识的一个复习。即便如此，这些章节的内容，特别是第 3 章，对了解如何使用库和接口也是非常重要的。

主要内容：

- 第 1 章　ANSI C 概述
- 第 2 章　C 的数据类型
- 第 3 章　库和接口



第1章 ANSI C概述

本章学习目标：

- 了解典型C程序的各个组成部分。
- 理解C语言中预定义的数据类型以及它们如何在程序中存储信息。
- 熟悉简化形式的工具和简单的标准输入输出（I/O）库，并用它们读取输入的数据和显示结果。
- 理解C语言中表达式的结构，并掌握如何使用常见的运算符来表达计算过程。
- 了解`if`、`switch`、`while`和`for`语句的格式，并使用它们编写简单的程序。
- 认识到将程序分解成独立的函数的重要性，并理解那些函数的工作机制。
- 能够综合使用本章中的控制机制编写简单的程序。

在刘易斯·卡洛尔的《爱丽丝漫游奇境》中，国王建议白兔："从起点开始，直到终点，然后停止。"如果你是刚开始学习编程，这是一个很好的建议。本书是为计算机科学的第二门课程所设计的，因此，我们假设你已经接触过编程。同时，因为你们学习的第一门课程包含的内容也有很大的不同，所以依赖于任何特定语言是很困难的。例如，部分学生可能已经有了用C语言编程的经验，然而大部分学生则是在第一门课程中学习了Pascal或其他的语言。

正是由于这个原因，最好的方法是采取国王的建议，从头开始。因此，本书前三章将以较快的速度介绍一下我认为对后续章节比较重要的内容。如果你学习过我写的《C语言的艺术和科学》一书，那么可以跳过这些章节，从第4章开始学习。如果你在其他课程中有了C语言编程经验，那么可以略过第1~2章，大概了解一下整体的思路，然后继续第3章关于接口的讨论。如果对编程有所了解，但是对C语言编程不熟悉，那么可以从本章开始学习，学完前三章的内容之后，您很快就会具备C语言编程的基础。

1.1 什么是C

早期，程序是用**机器语言**编写的，它包含很多基本指令，这些指令可以由机器直接执行。由于机器语言的结构反映的是机器的结构，而不是程序员的需求，因此机器语言编写的程序十分难懂。20世纪50年代中期，IBM中由约翰·贝克斯带领的一个编程小组产生了一个想法，这个想法对计算机的本质产生了深远的影响。他们在想，能不能使用数学公式来编写他们想要进行的计算，然后让计算机将这些公式解释为机器语言呢？1955年，该团队完成了Fortran（它是`formula`

translation 的缩写）的最初版本，这是第一种**高级编程语言**的示例。

从那以后，各种新的编程语言相继出现。其中，最成功的语言之一就是 C 语言，它是在 1972 年由贝尔实验室的丹尼斯·里奇设计的，后来进行了修订。1989 年，美国国家标准组织（American National Standards Institute，ANSI）对它进行了标准化。自从发明 C 语言的 25 年时间以来，C 语言已经成为在全世界使用最广泛的语言之一。今天很多新兴的语言（如 C++ 和 Java）都是基于它的，这也说明了 C 语言的成功。

在编写 C 程序的时候，首先应该创建一个**源文件**，它包含程序的文本。在运行程序之前，还必须将源文件编译为一种可执行的形式。其中的第一步利用**编译器**将源文件编译成**目标文件**。目标文件包含的是相应的机器语言的指令。该目标文件再和其他的目标文件组装生成能够在系统上运行的**可执行文件**。其他的目标文件主要是预定义的目标文件，在 C 语言中叫**库**（library），它包含程序所需要的各种操作的机器语言指令。组装各个独立的目标文件生成可执行文件的过程叫做**链接**。整个过程如图 1-1 所示。

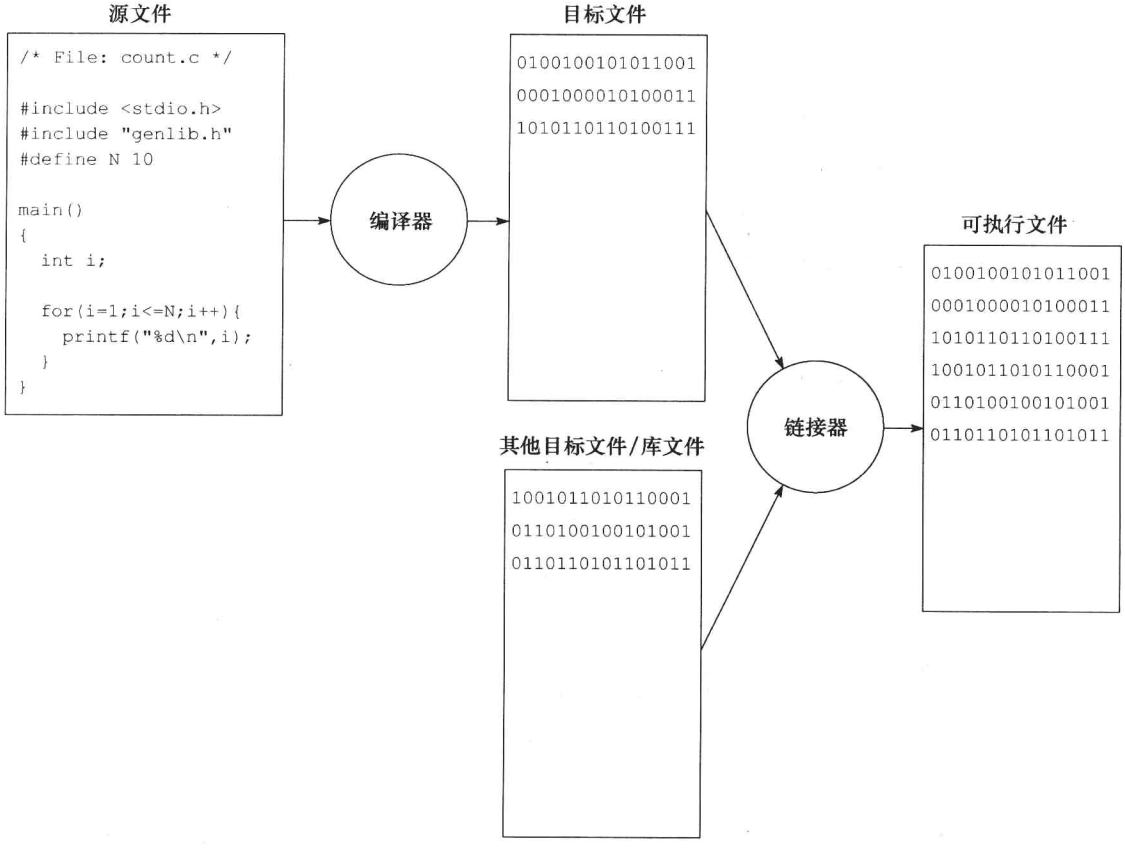


图 1-1　编译过程

遗憾的是，这个复杂过程的具体细节由于机器的不同而存在很大的区别，因而无法用一本通用的书来告诉你在你的系统上如何用确切的命令来运行一个程序。因为这些命令随着机器的不同

而不同，所以你必须参考C语言编译器所带的文档。然而，值得欣慰的是，C程序本身是没有差别的。用像C这样的高级语言编程的主要好处之一就是，你可以忽略硬件上的区别而编写可以在不同机器上运行的程序。

1.2 C程序的结构

对C语言进行感性认识的最好方法是看一个简单的实例程序，如图1-2所示。该程序产生一个表，比较 N^2 和 2^N 的值——第7章会证明这种比较是非常重要的，该程序的输出如下图所示：

N	N^2	2^N
0	0	1
1	1	2
2	4	4
3	9	8
4	16	16
5	25	32
6	36	64
7	49	128
8	64	256
9	81	512
10	100	1024
11	121	2048
12	144	4096

```
/*
 * File: powertab.c                                          } 程序注释
 * ----------------
 * This program generates a table comparing values
 * of the functions n^2 and 2^n.
 */

#include <stdio.h>                                           } 包含的库文件
#include "genlib.h"

/*
 * Constants                                                 } 程序段注释
 * ---------
 * LowerLimit -- Starting value for the table
 * UpperLimit -- Final value for the table
 */

#define LowerLimit  0                                        } 常量定义
#define UpperLimit 12

/* Private function prototypes */

static int RaiseIntToPower(int n, int k);                    } 函数原型
```

图1-2 示例程序 powertab.c