

经久不衰的计算机科学经典著作

100001110011111000100100010110000111001111100010010001011000011100111110001001000101



CODE

隐匿在计算机软硬件
背后的语言

(英文版)

[美] Charles Petzold © 著

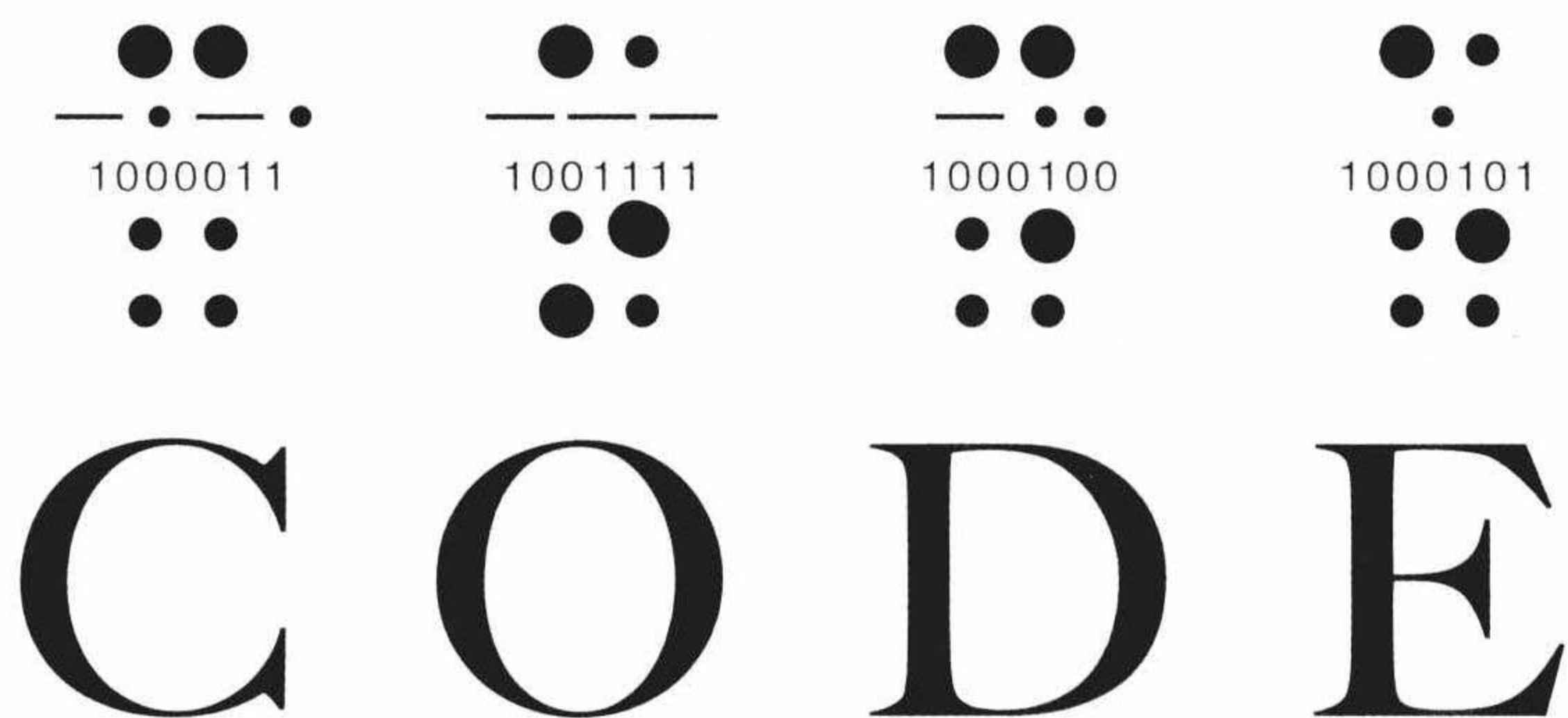
Code: The Hidden Language of
Computer Hardware and Software



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



隐匿在计算机软硬件 背后的语言

[美] Charles Petzold © 著

Code: The Hidden Language of
Computer Hardware and Software

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

这是一本讲述计算机工作原理的书。不过，你千万不要因为“工作原理”之类的字眼就武断地认为它是晦涩而难懂的。作者用丰富的想象和清晰的笔墨将看似烦杂的理论阐述得通俗易懂，你丝毫不会感到枯燥和生硬。更重要的是，你会因此更加深刻地理解计算机的工作原理。这种理解不是抽象层面上的，而是具有一定深度的，这种深度甚至不逊于“电气工程师”和“程序员”的理解。

不管你是计算机高手，还是对这个神奇的机器充满敬畏之心的“小白”，都不妨翻阅一下本书，读一读大师的经典作品，必然会有收获。

Authorized reprint from the English language edition, entitled Code: The Hidden Language of Computer Hardware and Software, ISBN: 9780735611313, by Charles Petzold, published by Microsoft Press, Copyright © 2000 Charles Petzold.

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.

ENGLISH language edition published by PUBLISHING HOUSE OF ELECTRONICS INDUSTRY, Copyright © 2019

本书英文影印版专有出版权由 Pearson Education 培生教育出版集团授予电子工业出版社。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书仅限中国大陆境内（不包括中国香港、澳门特别行政区和中国台湾地区）销售发行。

本书英文影印版贴有 Pearson Education 培生教育出版集团激光防伪标签，无标签者不得销售。

版权贸易合同登记号 图字：01-2019-0552

图书在版编目（CIP）数据

Code: 隐匿在计算机软硬件背后的语言 = Code: The Hidden Language of Computer Hardware and Software : 英文 / (美) 查尔斯·佩措尔德 (Charles Petzold) 著. —北京: 电子工业出版社, 2019.6

ISBN 978-7-121-35803-6

I. ①C… II. ①查… III. ①计算机科学—理论—英文 IV. ①TP3-0

中国版本图书馆 CIP 数据核字 (2018) 第 285074 号

责任编辑: 刘佳禾

印 刷: 天津千鹤文化传播有限公司

装 订: 天津千鹤文化传播有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 720×1000 1/16 印张: 25.75 字数: 498.1 千字

版 次: 2019 年 6 月第 1 版

印 次: 2019 年 6 月第 1 次印刷

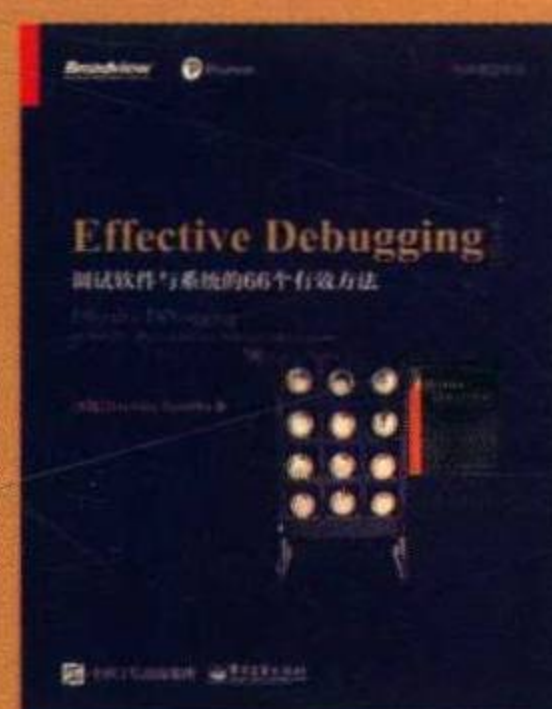
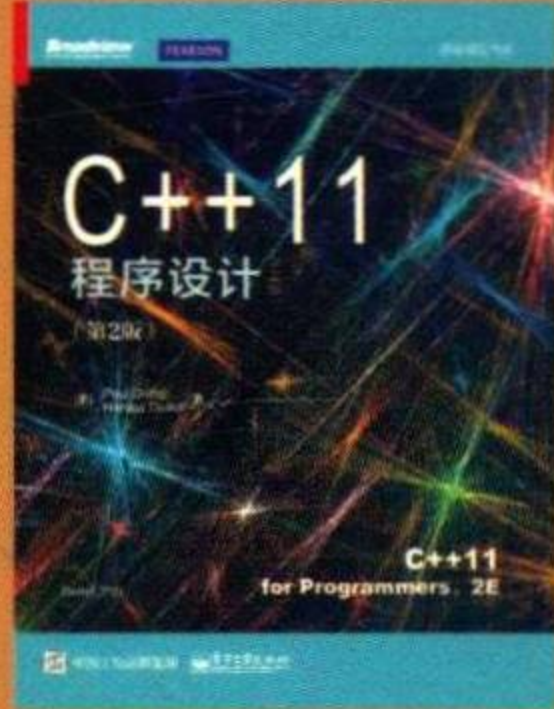
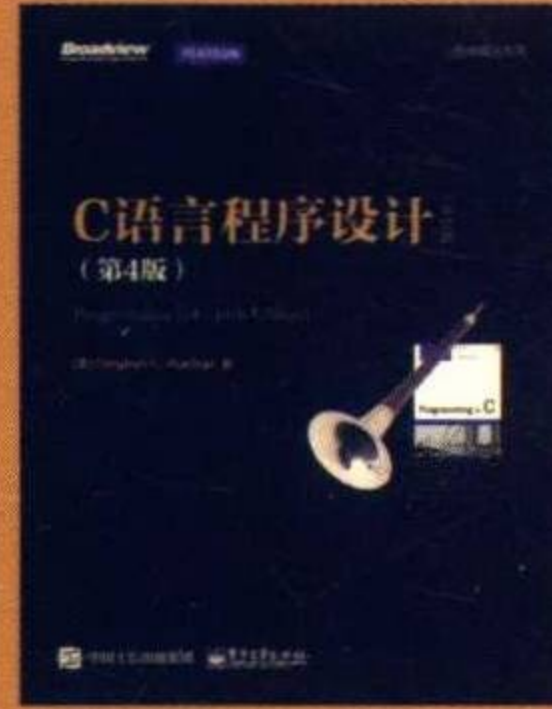
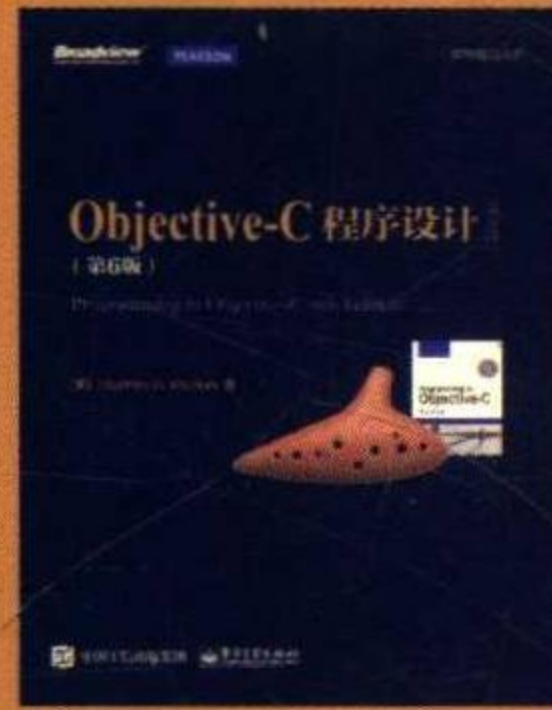
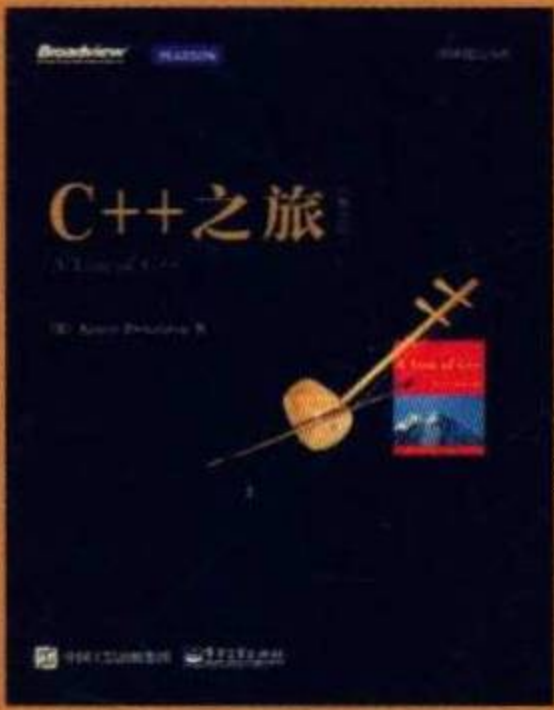
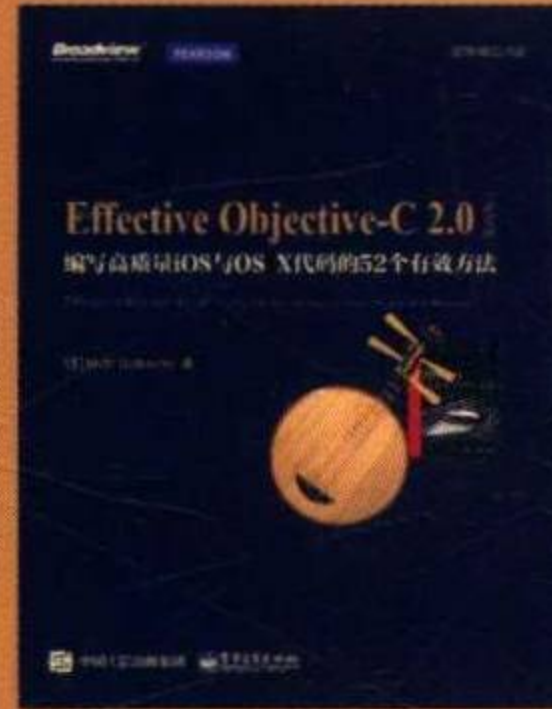
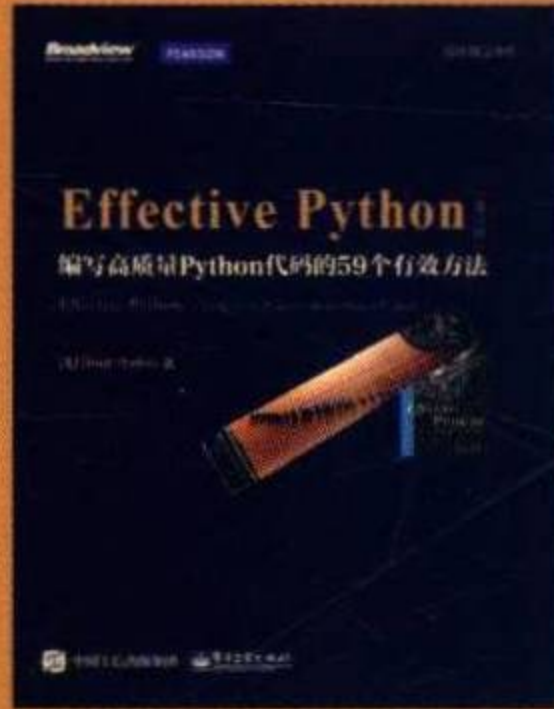
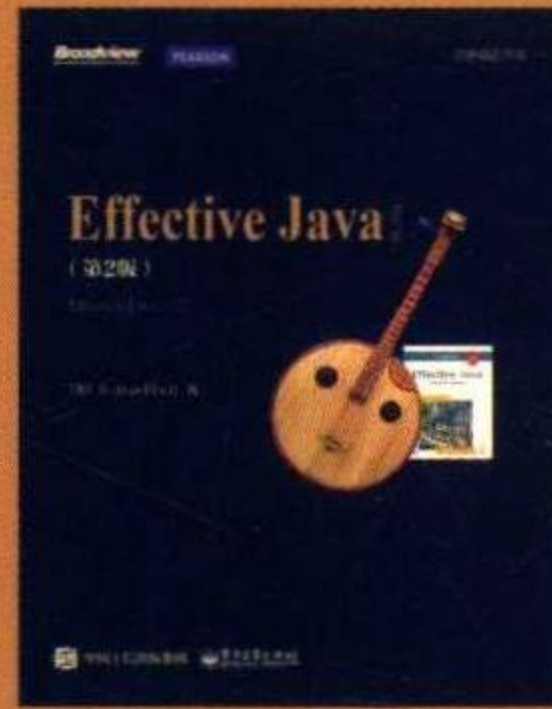
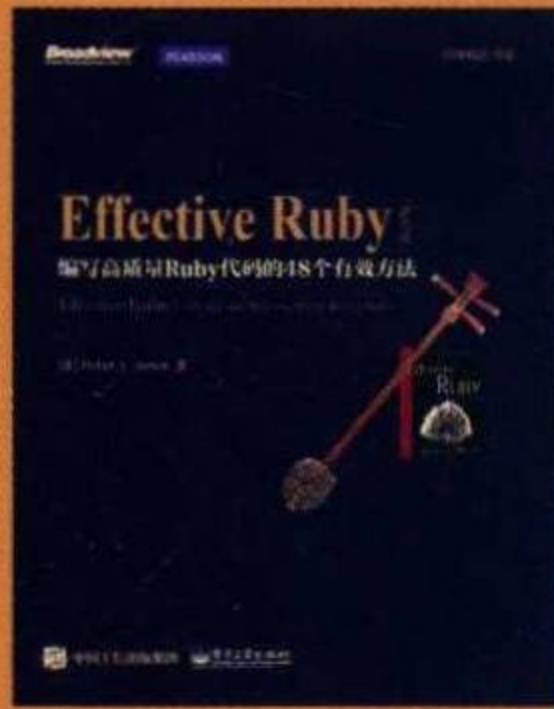
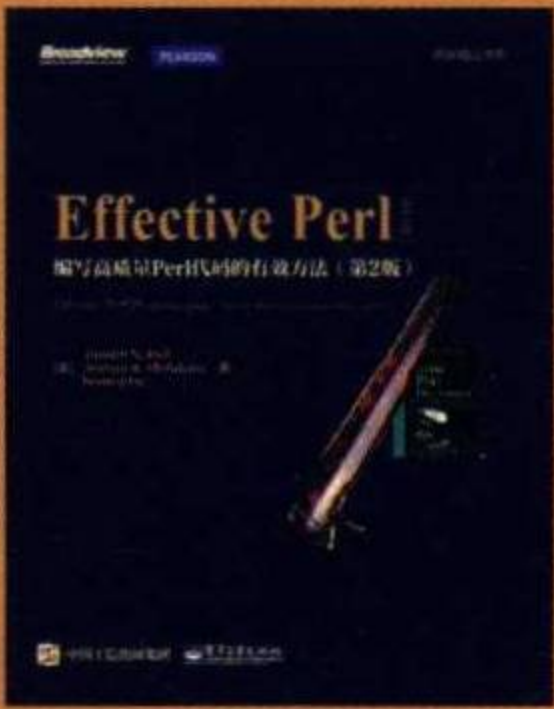
定 价: 118.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888, 88258888。

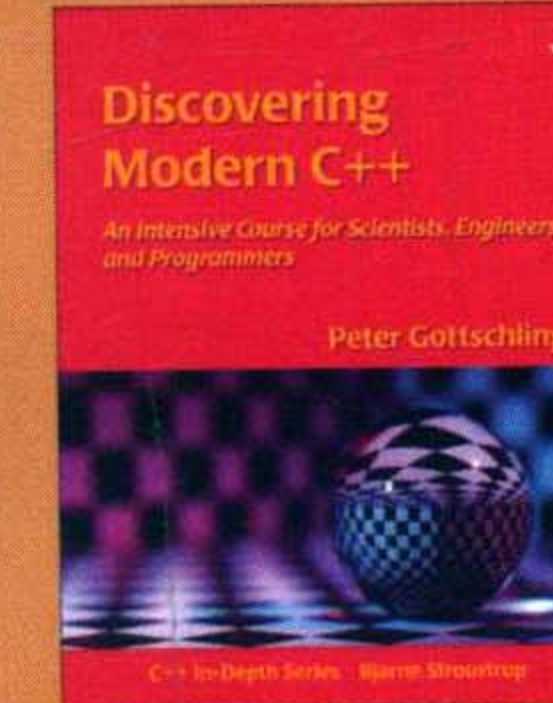
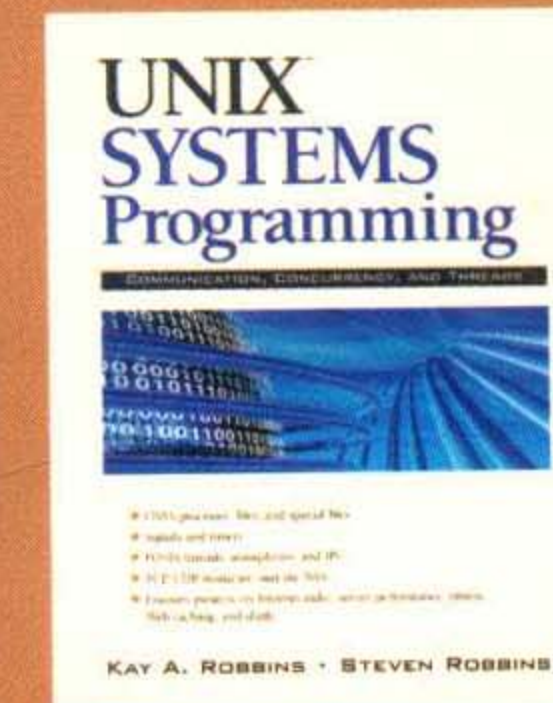
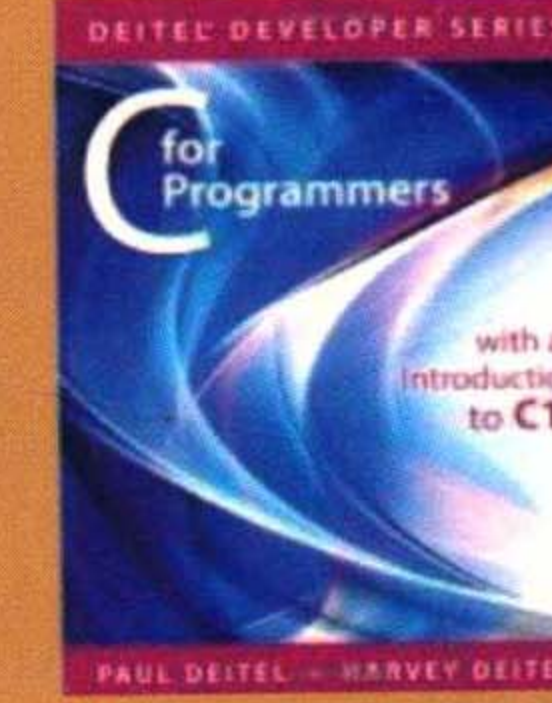
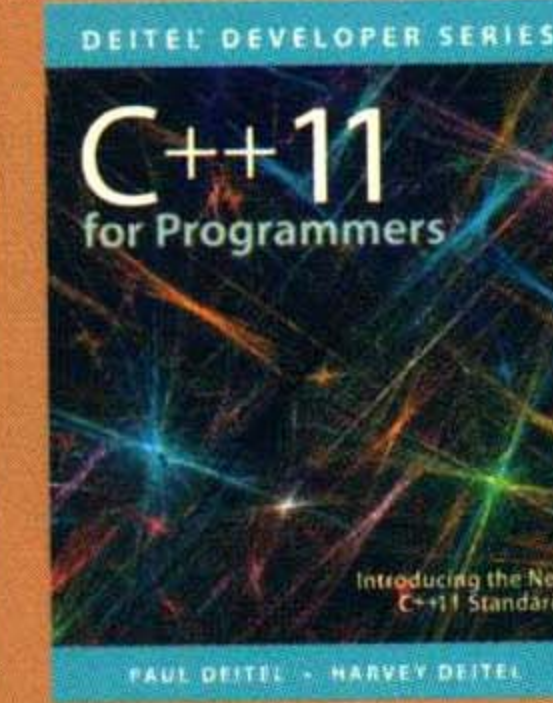
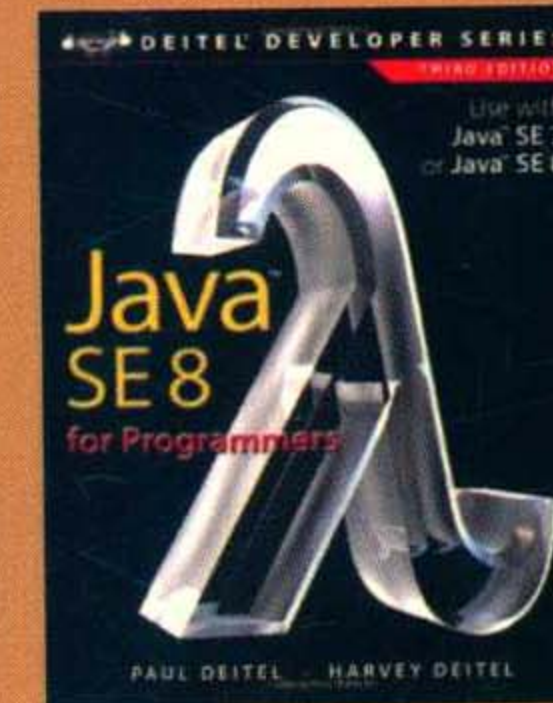
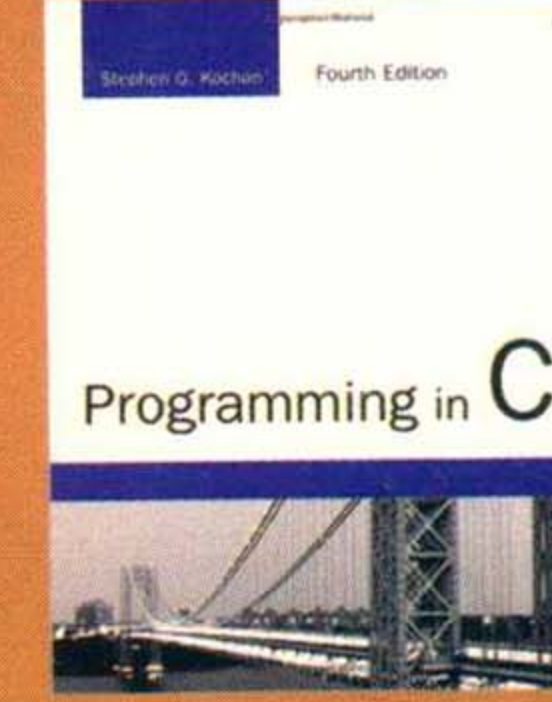
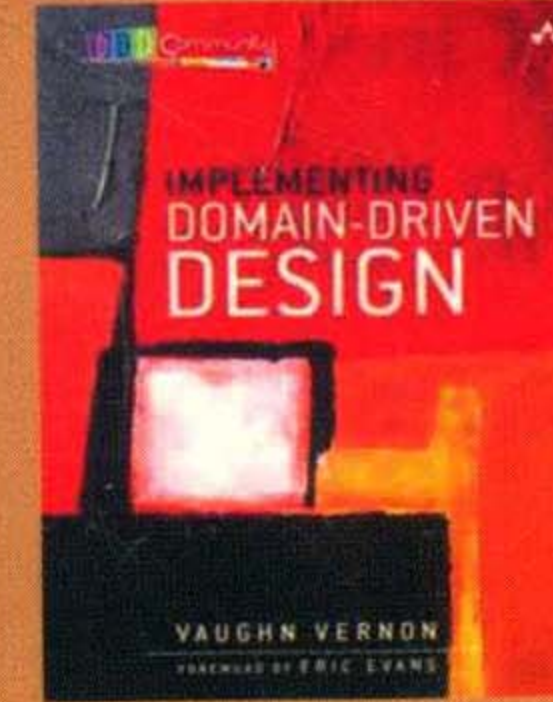
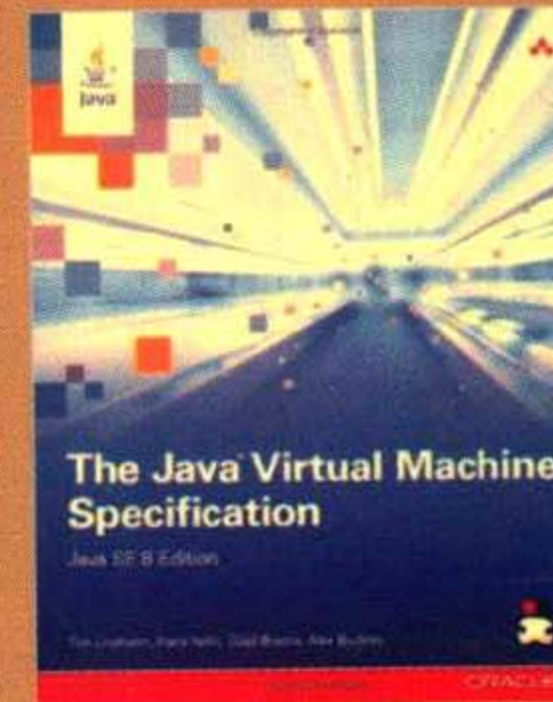
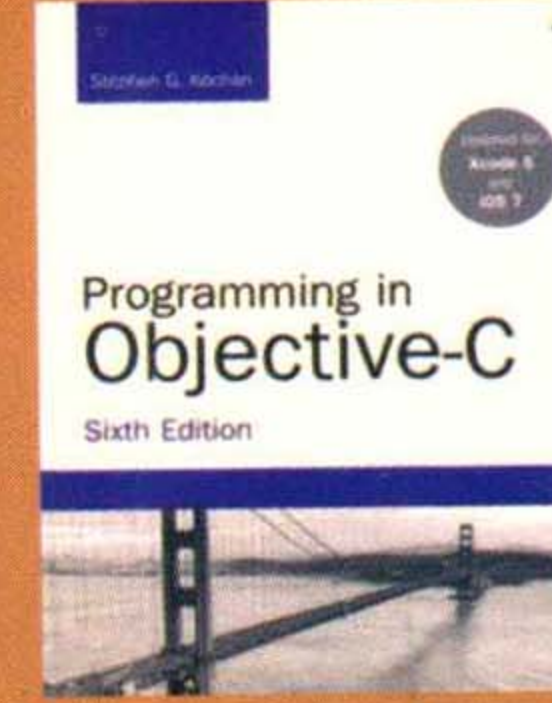
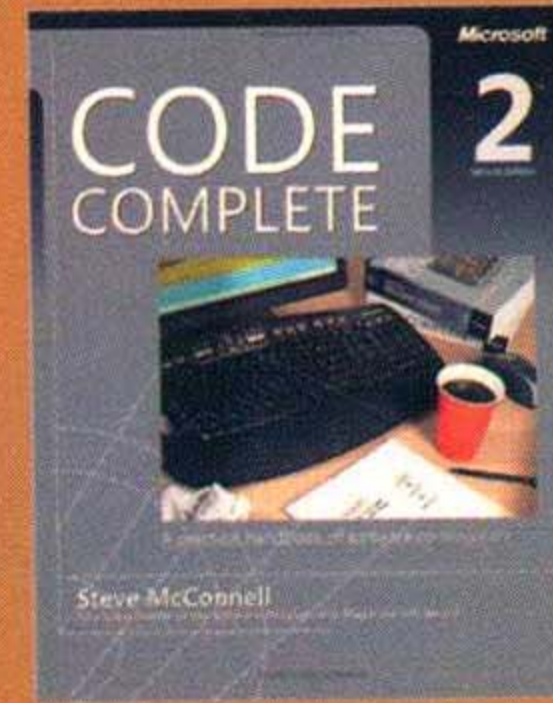
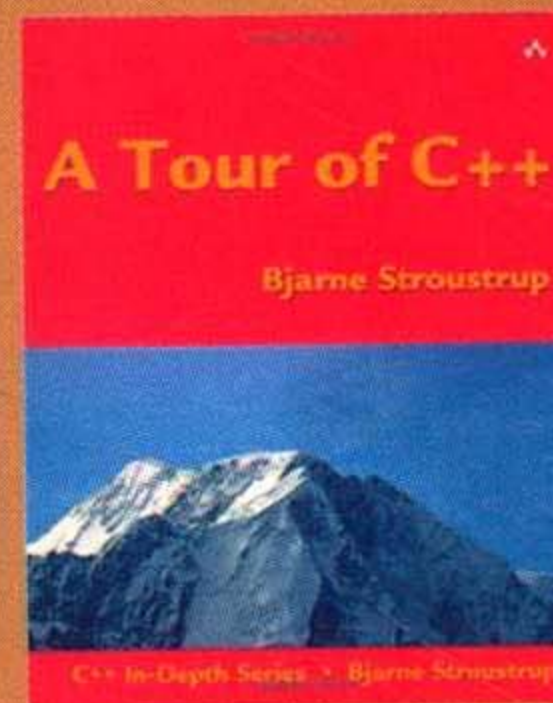
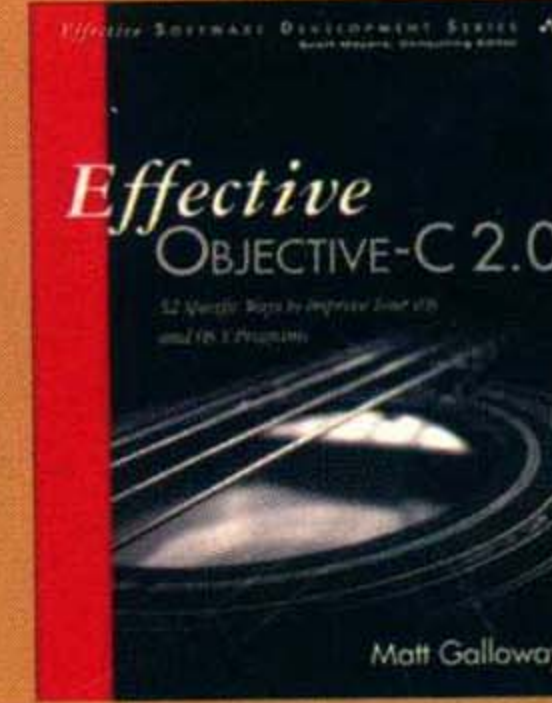
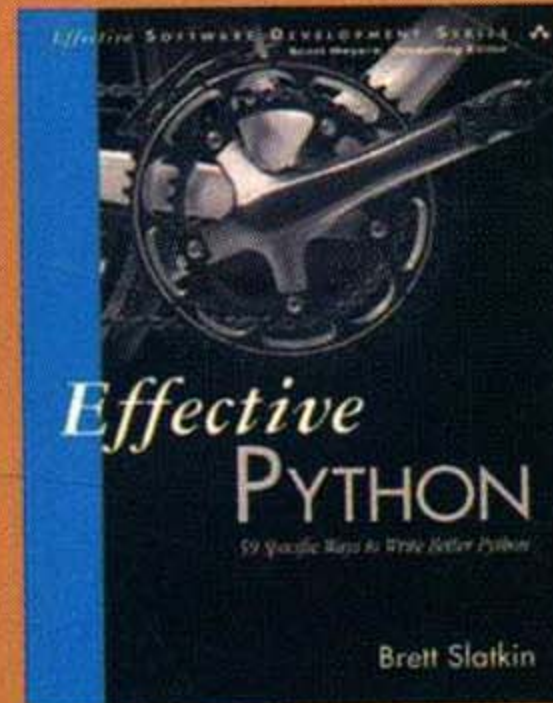
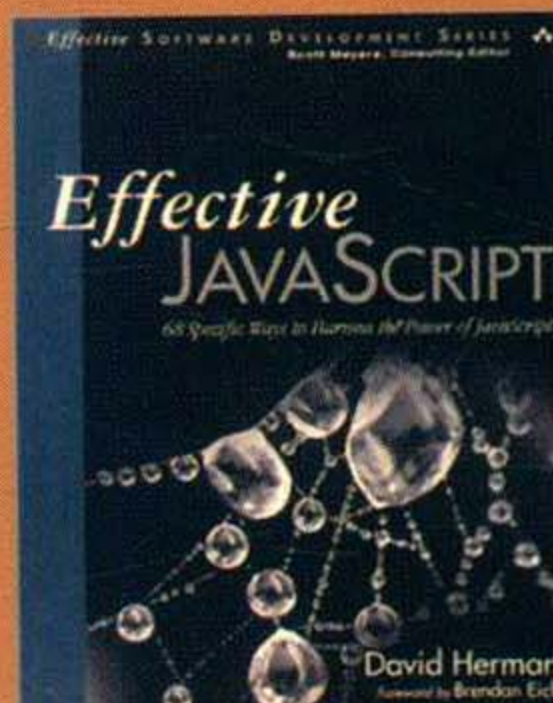
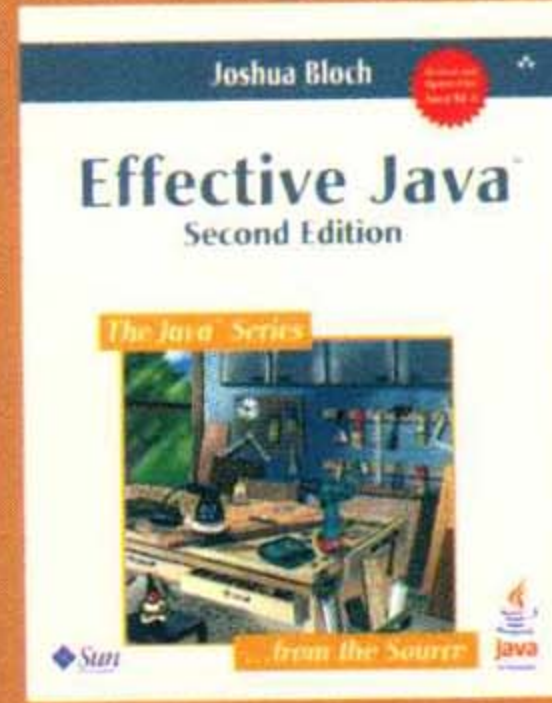
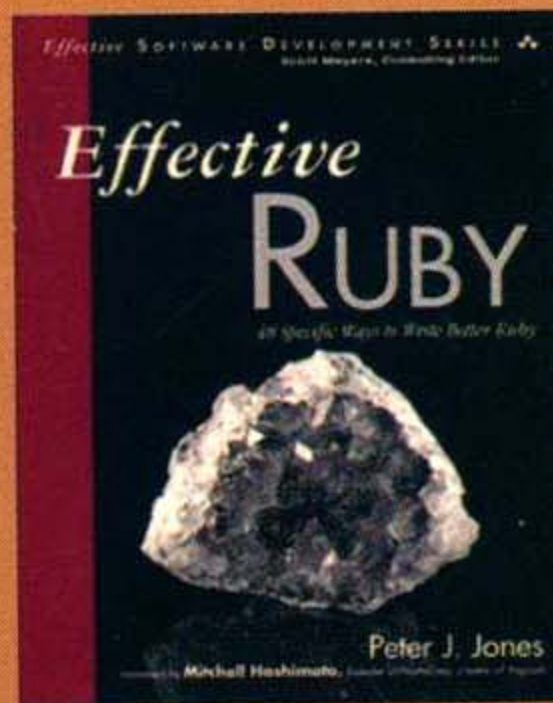
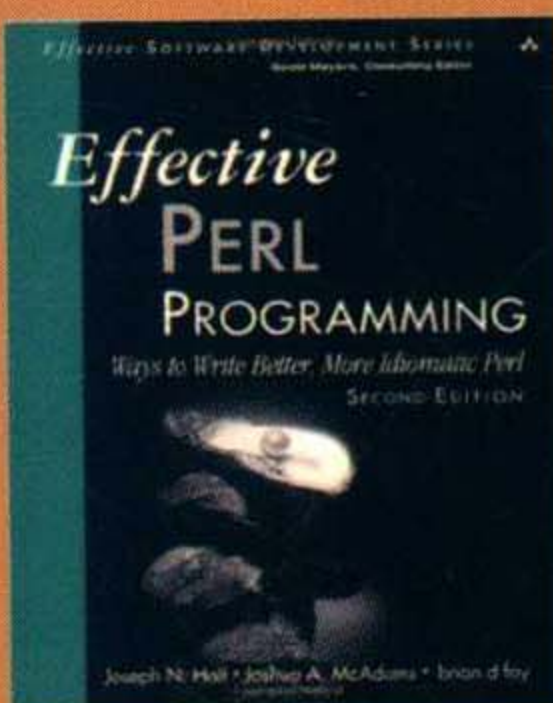
质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819, faq@phei.com.cn。

原味精品书系



原书封面





推荐序

《道德经》有云：“大方无隅，大器晚成。大音希声，大象无形。”这也就是老子所说“道”的至高境界。世界上最恢宏、壮丽的气派和境界，往往并不拘泥于一定的事物和格局，而是表现出“气象万千”的面貌和场景，科学知识的论述也是如此。科学不一定等同于晦涩和深奥，最复杂的理论也可用最简单的方式描述。这就是普遍存在于现实世界的辩证的、朴素的唯物主义精神。

我想 Charles Petzold 就是这样一位智者。作为 Windows 编程界开创者之一和世界顶级的技术专家，他在 Windows 编程方面著述丰富，在基础计算机理论书籍方面也取得了卓尔不凡的成就。他善用丰富的想象和清晰的笔墨将看似烦杂的理论描述得生动、简单，使读者不会感到枯燥和生硬。尤其这本畅销多年、经久不衰的 *Code: The Hidden Language of Computer Hardware and Software* 更是化腐朽为神奇、化晦涩为通俗之杰作。即使读者不是计算机专业的人士，在 Petzold 笔墨的引导之下，也能够轻松阅读本书。书中使用像电线、电灯泡、触发器等这些非常常见的零件，拼拼接接，最后出人意料地建造了一台电子计算机。读者在阅读过程中，可以循序渐进地学习有关计算机系统基础和构造的知识，作者的奇思妙想和妙趣横生的讲述方式不得不令人赞叹。

透过现象进而探索本质，可以发现新事物；将复杂的事物简单化，可以发现解决问题的新方法。作者将那些看似复杂的理论转换成我们熟悉的诸多景象，使读者不由自主地跟随他的思路打造这台只由电线、电灯泡和触发器组构成的庞大的计算机系统。

最后，我还是想将《道德经》上的两句话来送给各位读者——“为学日益，为道日损，损之又损，以至于无为，无为而无不为。”衷心希望各位读者有朝一日都能够在学问上到达无为而治的境界。

丁秋林
2009.11.28.

丁秋林教授：博士生导师，南京航空航天大学信息科学与技术学院和机电工程学院资深教授。英国皇家航空学会高级会员、英国剑桥世界名人录传记人物（IBC Cambridge the 28th Edition）、美国世界名人录传记人物（Who's Who in the World the 16th Edition）、国际高级制造技术杂志编辑部成员（Springer）。

推荐序

在真正开始撰写这本书之前，纷繁的思绪在我的脑海中已经萦绕了近十年之久。但从着手写作，一直到出版，我不断反复地问着自己一个问题：这本书是讲什么的？

关于这个问题我总是找不到合适的答案。我想说：“这本书将带领大家通过一段信息技术革命的旅程重新感受现代计算机技术。”语毕，我内心纠结，这个回答真的将这本书的内容完整诠释出来了吗？

最后我不得不承认：“这本书其实是讲述计算机是如何工作的。”

我之所以有些底气不足，是因为我已经猜到了某些读者的反应：“嗨，这种书我早就读过了。”但我会立刻告诉他：“不，这种书你并没有读过。”对于这点，我坚信不疑。因为这本书并不是讲“计算机—如何—工作”。书中并没有用一张又一张描述磁盘的插图，没有用各种箭头解释数据是如何被输送到计算机的，书里面也没有装载着一节又一节“0和1数据”的“火车”图片。明喻与暗喻是文学描述中精妙的辅助手段，但它们常常掩盖了科学技术的真正光芒。

我还听过另一个说法：“人们不愿意了解计算机是如何工作的。”对于这句话我毫无保留地赞同，因为就我个人来说，也是在不经意之间才体会到理解事物如何运作这一过程是多么有趣。但是这并不意味着我对世上一切事物的运作机制都感兴趣，都要有所了解，因此我绝对不会在任何场合向大家解释我房间里冰箱的工作原理。

然而与此同时，经常有人向我问起一些涉及计算机内部运行机制的问题。一个常见的

例子就是：“存储器（storage）与内存（memory）有何区别？”

对于计算机用户而言，这个问题要是搞不清楚的确“寸步难行”，可以看到，市场上考察个人计算机的存储性能，最主要考察的就是这两个概念。即便对于最初级的计算机用户来说，他们也一定需要了解到底多少“兆字节”或多少“吉字节”的存储器才能应对运行在其上的程序。如果进一步思考，这些初级用户或许更加想了解计算机中的“文件”是什么概念，甚至连带这些文件如何从存储器加载进内存，又如何从内存存储到存储器，他们也非常期待学习这些知识。

像这类“存储器—内存”问题的解答在本书中大都使用类比法——内存就好比你的办公桌，而存储器就好比你的文件柜。就问题本身而言，它的确给出了满意的回答。但我对这个答案并不满意，主要原因在于这个答案将计算机的体系结构与办公室的结构等同起来，但是内存与存储器的区别其实是在逻辑层面上的，它体现着计算机体系结构的实际需求与存储器客观性能之间的矛盾。简单地说，就是我们找不到一种同时具备这两种存储器所有优点的存储媒介，这些优点就包括存储速度快、存储容量大、非易失性，等等。今天的计算机都采用“冯·诺依曼体系结构”——五十年来它一直是计算机体系结构的主导，而内存与存储器的区分也正是由于这种体系结构的不足所导致的。

还有一些计算机用户问我这样一个问题：“为什么 Macintosh 环境下的程序不能在 Windows 下运行？”我想立刻回答这个问题，但在我刚开口时就立刻意识到，这个问题涉及太多的技术细节，要想彻底搞清楚，那么提问的这位朋友一定要和我来一次“茶话会”不可。

我希望这本书能够成为大家理解这些问题的“助手”，这种理解我希望不是抽象层面上的，而是具有一定深度的，甚至不逊于“电气工程师”和“程序员”的理解深度。我同时也非常希望大家能够理解：计算机是 20 世纪技术领域的“登峰造极之作”，它是一种值得欣赏、具有“美”学文化底蕴的人类伟大成果，这种“美”不需要明喻与暗喻的额外修饰。

计算机拥有与生俱来的层次化体系结构，这种结构的底层是晶体管，其顶层则是计算机显示器上所呈现的信息。从下向上分析该结构的每一层——这也是本书的编写结构——并没有人们想象中的那么难。当然，现代计算机的内部结构不断推陈出新，但其本质上仍然是一些常见且简捷的操作组合。

尽管今天的计算机与 25 年前，以及 50 年前的计算机相比都复杂许多，但它们在本质上是完全一致的。学习技术发展史的重要意义正在于此：追溯的历史越久远，技术的脉络

就变得越清晰。因此，我们需要做的就是确定某些关键的历史阶段，在这些阶段，技术最天然、最本质的一面将清晰可见。

在这本书中，我回溯了自己能找得到资料的计算机发展史。令我自己也感到惊讶的是，竟然一直追溯到了 19 世纪，甚至使用了早期的电报设备来演示计算机是如何构建的。至少从理论角度来看，本书的前 17 章中提到的所有设备，都可以利用已经存在了一个世纪的简单电子器件来构造。

这些古董级技术的使用令本书蒙上了一层怀旧的面纱。我要强调的是，*Code* 是这样一本书，它永远不会被命名为诸如 *The Faster New Faster Thing* 或 *Business @ the Speed of a Digital Nervous System*。在这本书中，“bit”定义在第 68 页，“byte”的定义则出现于第 180 页。晶体管（transistor）直到第 142 页才被介绍，而且只是顺便提及而已。

在对计算机工作原理介绍方面，本书将会一直深入到本质（例如，只有少数书会去介绍计算机处理器的实际工作机制），但整本书的节奏是相对轻松的。在保证内容深度的基础上，我尽量使读者在学习的旅程中保持轻松愉悦的心情。

我最后还是要说，本书中没有那种装载着一节又一节“0 和 1 数据”的“火车”图片。

Charles Petzold

2000 年 8 月 16 日



目录

1 Best Friends / 3

编码是什么？在本书中，这个词是指一种用来在机器和人之间传递信息的方式。换句话说，编码就是交流。对任何能听见我们的声音并理解我们所说的语言的人来说，我们发出的声音所形成的词语就是一种编码。用手电筒能代替声音来与朋友交谈吗？当然值得一试。

2 Codes and Combinations / 9

莫尔斯码也被称做二进制码，因为这种编码的组成元素只有两个——“点”和“划”。不过，点、划的组合却可以表示你想要的任意数目的码字。这其中的规律是什么？

3 Braille and Binary Codes / 15

布莱叶盲文是为了便于盲人阅读而发明的一种编码。在这一章中我们将解析布莱叶盲文，来看看它是如何工作的。我们并不是要真的学习布莱叶盲文，而且也无须刻意记住关于它的什么内容。我们仅仅希望从中归纳出编码的一些性质。

4 Anatomy of a Flashlight / 22

为了理解电在计算机中的工作原理，我们先得仔细钻研一番电学，不过不要担心，只需要一部分基础知识就够了。在本章，我们将以手电筒为教学道具，引导你走入神秘的电学世界。

5 Seeing Around Corners / 32

在第 1 章，我们曾经讲过用手电筒与朋友进行交谈的方式，但是这样的方式是有局限性的，你的交流对象必须住在街对面，而且你们卧室的窗口正好相对。但是，现实不会总是如此。当手电筒的光无法到达朋友的卧室时，怎样与他进行无声的交流呢？电路或许可以助你一臂之力。

6 Telegraphs and Relays / 40

全球即时通信对于我们来说已经司空见惯，但你要是生活在 19 世纪早期，可没这么方便。你当然可以进行即时通信或者远距离通信，但是不可能同时做到这两点。即时通信会受声音传播距离或者视野的限制。使用信件倒是可以进行更远距离的通信，但是寄信耗费的时间太多，并且需要借助交通工具。为了解决这个问题，电报应运而生，而伴随着电报诞生的继电器更是具有重要意义的伟大发明。

7 Our Ten Digits / 47

人们很容易理解语言只不过是一种编码。比如英文中的“cat”（猫）在其他语言中可以写做 gato、chat、Katze、КОИИК 或 *kátta*。然而，数字似乎并不会那么容易随文化的不同而改变。不论我们说什么语言，或使用什么样的发音，在这个星球上的所有人都用以下方式来书写数字：0, 1, 2, 3, 4, 5, 6, 7, 8, 9。你了解这十个数字吗？

8 Alternatives to Ten / 54

对人类而言，10 是一个非常重要的数字。它是我们大多数人拥有的手指或脚趾的数目。我们人类已经适应了以 10 为基数的数字系统。但是只能使用十进制来计数吗？如果人类像卡通人物那样每只手只有 4 根手指会怎样？

9 Bit by Bit by Bit / 69

二进制是最简单的数字系统，其中只包含两个数字：0 和 1。二进制中的 1 位（bit）称为 1 比特，我们可以用它来表达简单的信息：是或不是，亮或灭，打开或关闭，等等。而事实上只要信息能转换成两种或多种可能性的选择，就都可以用比特来表示。这种例子在日常生活中随处可见，比如照相机胶卷的胶片速度，各种商品包装上的条形码等。

10 Logic and Switches / 86

对于古希腊人而言，逻辑是在追求真理的过程中所使用的一种分析方法，是一种哲学形式。而英国的数学家乔治·布尔却认为可以找到一种数学形式来描述逻辑，因此他发明了布尔代数。更重要的是，布尔代数运算可以用开关、导线和灯泡组成的电路来实现，布尔代数中的 AND 和 OR，与线路中开关的串联和并联，有着奇妙的对应关系。

11 Gates (Not Bill) / 102

继电器像开关一样，可以串联或并联在电路中执行简单的逻辑任务。这种继电器的组合叫做逻辑门 (Logic Gate)，也简称门。这里提到的逻辑门执行“简单的”逻辑任务是指逻辑门只完成最基本的功能。本章就介绍那些用以完成最基本逻辑任务的门。

12 A Binary Adding Machine / 131

加法是算术计算中最基础的运算，如果想搭建一台计算机，首先就要搭建出计算两数之和的器件。本章我们将利用前面章节中用过的开关、灯泡、导线、电池、逻辑门等这些简单的元件，搭建一个二进制加法器。

13 But What About Subtraction? / 143

当你确信继电器连接在一起真的可以实现二进制数加法时，你可能会问：“如何实现减法呢？”问得好！这表明你是相当有觉察力的，加法和减法在某些方面互相补充，但是在机制上二者却存在本质区别。不过没关系，我们可以想一些办法，把减法运算变成加法。

14 Feedback and Flip-Flops / 155

想象一下，如果你没有了记忆力，该如何去数数？我们不记得刚刚数过的数，当然也就无法确定下一个数是什么。同理，一个能计数的电路必定需要触发器。本章要介绍的就是各种触发器。

15 Bytes and Hex / 180

在前面的章节中，加法器、锁存器以及数据选择器的输入和输出形式都是 8 位的数据流，也即数据路径的位宽为 8，为什么要定义为 8 位呢？为什么不是 6 位、7 位、9 位或 10 位？本章就要解释其中的缘由。

16 An Assemblage of Memory / 190

每天清晨，我们将自己从沉睡中唤醒，这时大脑的空白会很快被记忆填满。我们立刻会意识到自己身在何方，最近做了些什么事情，有什么计划打算。有的事情我们很快就能想起来，但有时并非如此。我们可以借助许多工具来记录信息，比如笔和纸、磁带，当然现在还可以使用存储器。

17 Automation / 206

人类的本性中带有一些懒惰的特质。我们总是抵触繁重的工作，对枯燥的、重复性的工作深恶痛绝。所以，当你必须用前面搭建的加法器计算 100 个数，甚至更多个数的加法时，有一种念头就会不可遏制地从脑子里冒出来：怎样让加法器自动完成数据输入和计算呢？办法肯定是有的，那就是编写程序。

18 From Abaci to Chips / 238

算盘、滑尺、纳皮尔骨架、差分机、解析机、继电器、电子管、晶体管、芯片、计算机；甘特、帕斯卡、莱布尼兹、杰奎德、巴贝芝、图灵、冯·诺依曼、香农；IBM、贝尔实验室……你觉得应接不暇了吗？把这些你或熟悉或生疏的名词和名字串起来，就是人类的计算工具发展史。让时光倒流，去看看那些精巧的工具，感受天才们的巧思吧！

19 Two Classic Microprocessors / 260

将中央处理器的所有构成组件封装到一块硅芯片上，就得到了微处理器。第一片微处理器芯片诞生于 1971 年，即 Intel 4004 系列，其中集成了 2300 个晶体管，你或许觉得可笑——如今家用计算机的微处理器上所安置的晶体管数量已经以亿为计量单位了。但是，从本质上来说，微处理器实际所做的工作并没有变。在本章，我们就来看看两种有着辉煌历史的典型微处理器：Intel 8080 和 Motorola 6800。

20 ASCII and a Cast of Characters / 286

计算机中的存储器唯一可以存储的形式是比特，因此如果想在计算机上处理信息，就必须把它们转换为比特的形式来存储。我们已经掌握了如何用比特来表示数字和机器码。如何用它来存储文本呢？毕竟，人类所积累的大部分信息，都是以各种文本形式保存的。下面就轮到 ASCII 码出场了！

21 Get on the Bus / 301

一台计算机包括很多部件：中央处理器、存储器、输入/输出设备等。通常这些部件按照功能被分别安装在两个或更多的电路板上。这些电路板之间通过总线（Bus）通信。如果对总线做一个简单的概括，可以认为总线就是数字信号的集合，而这些信号被提供给计算机上的每块电路板。

22 The Operating System / 320

你或许梦想过自己组装一台近乎完整的计算机，像老木偶匠盖比特雕刻木偶匹诺曹一样，全部亲自动手用小零件完成。不过在你的机器能完成你想要的操作之前，还差一个重要的东西——操作系统！

23 Fixed Point, Floating Point / 335

整数、分数以及百分数等各种类型的数字与我们形影不离，它几乎出现在我们生活的所有角落。例如你加班 2.75 小时，而公司按正常工作时间的 1.5 倍支付你工资，你用这些钱买了半盒鸡蛋并交了 8.25% 的销售税。在计算机的内存里，所有的数都表示为二进制形式。通过前面的学习，我们知道 2 用二进制可以表示为 10_2 ，可是 2.75 用二进制怎样表示呢？这就是本章的主题。

24 Languages High and Low / 349

第 22 章介绍了如何编写一段简单的程序，让我们可以利用键盘将十六进制机器码输入计算机，以及通过视频显示设备来检查这些代码。但使用机器码编写程序就如同用牙签吃东西，伸出手臂费半天劲刺向食物，但每次都只取到小小的一块，用这种低级语言编写程序既费力又费时，有悖于我们发明计算机的初衷。不过，人们想出了一种效率更高的编程方法——使用高级语言。

25 The Graphical Revolution / 364

回顾历史，从第一台继电器计算器到现在为止，六七十年过去了，计算机的处理速度飞速增长。不过要充分利用计算机日益增长的运算和处理能力，就必须不断改进计算机系统中的用户接口（User Interface），因为它是人机交互的轴心。图形化革命来了！

Acknowledgments / 383

Index / 385

由于黑白印刷，书中的红色只能显示为淡灰色。

——编者注

编码 (biān mǎ)

3. a. 一种在信息传输过程中用来表述字母或数字的信号系统。
 - b. 由被赋予了一定主观意义的符号、字母及单词所组成的系统,该系统可用于传输需要保密或简短的信息。
4. 一种由若干符号和规则组成的系统,用来向计算机表述指令。

——摘自《美国传统英语词典》

code (kōd) ...

- 3.a. A system of signals used to represent letters or numbers in transmitting messages.
 - b. A system of symbols, letters, or words given certain arbitrary meanings, used for transmitting messages requiring secrecy or brevity.
4. A system of symbols and rules used to represent instructions to a computer...

— *The American Heritage Dictionary of the English Language*