

04年第1期：荣耀——C程序设计之四书五经

<http://book.csdn.net> 2006-7-21 19:28:00

C程序设计之四书五经

荣耀/文

引言

自从Dennis M.Ritchie于1973年设计并实现C语言以来，计算机程序设计领域已经发生了巨大的变化。以C语言为根基的C++、Java和C#等面向对象语言相继诞生，并在各自领域大获成功。今天，C及其后裔几乎统治了计算机程序设计世界。可以这么说，C语言的诞生是现代程序语言革命的起点，它改变了程序设计语言发展的轨迹，是程序设计语言发展史中的一个里程碑。

然而，C并不仅仅是其他语言的起点和基础，否则它早已成为程序设计语言发展过程中又一个被遗弃者。和当初发明时一样，C语言今天依然至关重要。它的表达力、效率、简练、紧凑、对机器的最终控制以及跨平台的高度移植性，使其在系统编程、嵌入式编程等领域一直占据着统治地位，而C99标准的制订则再一次使C语言焕发出新的活力。下文介绍C程序设计领域中的几本好书，其中一些堪称经典。

1. Brian W.Kernighan, Dennis M.Ritchie, 《C程序设计语言》



清华影印版、习题集；机械译文版（新版、老版）：习题集

这是迄今为止在所有程序设计语言书籍中最广受尊敬的一部经典，是任何一名C程序员的必读之作。因为出自C语言的设计者 Dennis M. Ritchie 和著名的计算机科学家 Brian W. Kernighan之手，它被昵称为“K&R C”。是它首先引入了“Hello World!”程序，这个程序几乎成了后来任何一本入门性程序设计语言书籍中的第一个例子。

如同C语言本身简洁紧凑而极具威力一样，这本书轻薄短小而极富张力。通过简洁的描述和典型的示例，它全面、系统、准确地讲述了C语言的各个特性以及C程序设计的基本方法，内容涵盖基本概念、类型和表达式、控制流、函数与程序结构、指针与数组、结构、输入与输出、UNIX系统接口以及标准库等内容。

简洁清晰是这本书最大的特色。这本小书可以教给你许多比它厚几倍的“大部头”的知识。我认为那些动辄洋洋洒洒拼凑出好几百页乃至上千页的技术作者应该好好向K&R学一学。对于中、高级程序员而言，如果希望迅速获得C语言的严肃知识而又不愿意多花费哪怕一丁点时间，这本书就是首选。

顺便说一句，这本书的索引制作非常出色，极具实用价值，这可能首先要归功于正文部分的简明扼要。此外，尽管它看上去很像一本教程，但其实更是一本写给专业程序员的指南。如果你不具备任何其他语言程序设计背景或基本的C语言知识，这本书也许并不适合用作你的C语言启蒙读物。

2. Peter Van Der Linden, 《C专家编程》



C专家编程

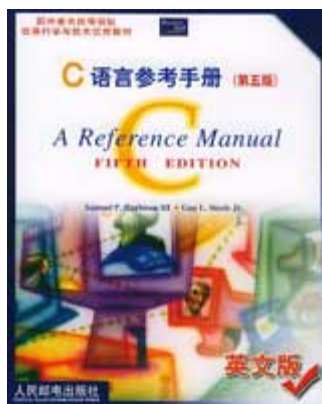
C语言是严肃的程序员语言，但这并不意味着C语言书籍必须板着脸说教。在这本被C程序员昵称为“鱼书”（封面上有一条“丑陋的”腔棘鱼）的著作中，作为SUN公司编译器和操作系统核心开发组成员之一，Peter淋漓尽致地展示了其深厚的技术沉淀、丰富的项目经验、高超的写作技巧以及招牌式的幽默。在这部作品中，作者以流畅的文字、诙谐的笔法将逸闻典故、智慧和幽默自然地融入技术描述中，读来宛若一本小说，极富趣味。

本书讲述了C语言的历史、语言特性、声明、数组、指针、连接、运行时以及内存使用等知识，揭示了C语言中许多隐晦之处，尤其深入解析了声明、数组和指针、内存使用等方面的细节。要想成为一名专家级C程序员，这些内容都是必须掌握的。和其他满是抽象例子的C语言书籍不同，这本书充满了大量的来自真实世界的C程序设计实例，它们对C程序员具有很高的参照价值。另外，每一章都以极富趣味的“轻松一下”收尾，而附录A“程序员工作面试的秘密”则是任何语言的程序员在应聘工作前增强自信的好材料。

我怀疑真正的C专家可能用不着看这本书——从内容到组织方式到行文风格都决定了这是一本轻松愉快的“从菜鸟到高手”的进阶读本，所以，它理应拥有更广泛的读者群。初级程序员往往更需要热情的鼓励，在阅读这本书的过程中，你定会深深地被作者对编程的激情所感染。

世间并无完美。我认为这本书的缺陷在于，和大多数平庸的C语言书籍一样，它画蛇添足地加入了一章关于C++的描述。在今天看来，这个描述既不全面也有失公允。不过，鉴于作者是在1994年从一名C程序员的角度去观察C++，这一点也就不足为奇了。

3. Samuel P. Harbison, Guy L. Steele, 《C语言参考手册（第五版）》



影印版（机械工业出版社）、中文版（机械工业出版社）

在C语言参考手册类书籍里，Samuel P. Harbison 和Guy L. Steele合著的C: A Reference Manual是非常出色的一本。这本手册的第五版新增了对C99标准的介绍，以便满足新时期C语言学习的需要。全书共分为两大部分，第一部分专注于C语言特性，第二部分则全面讨论了C标准库。本书涵盖C99、C89、传统的C、所有版本的C运行库以及编写与C++兼容的C代码等一切知识。

这本手册只是中等厚度，但它比“比它更厚”的其他参考手册更清晰地描述了C语言的现在和过去的方方面面。整本手册技术细节描述精确，组织条理清楚，内容完备详尽而又简明扼要。可以这么说，它在广度、深度和精度方面都是出类拔萃的。对于中、高级C程序员而言，这本手册值得常备案头，它几乎肯定要比K&R的著作使用频率更高。

4. David Hanson，《C语言接口与实现：创建可重用软件的技术》，机械工业出版社



C语言接口与实现：创建可重用软件的技术

C语言能够历经三十多年而不衰，一个重要的原因在于它的适应能力。在这“复用”、“面向对象”、“组件”、“异常处理”等先进机制漫天飞舞的年代，C语言仍然能够凭借它小而优雅的语言特性，在相当程度上满足现代软件体系架构提出的要求。只不过，想要达到这个程度，必须要在C的应用功力上达到最高层次。在嵌入式、系统软件以及对性能要求极高的系统开发中，开发人员必须达到这样的层次，熟练掌握C语言的高级特性，才能够同时满足效率和灵活性、复用性的要求。可惜，虽然C语言技术图书汗牛充栋，但是关注这个峰顶之域的作品却是屈指可数。David Hanson的C Interfaces and Implementations就是个中翘楚。

David Hanson是业内大名鼎鼎的自由编译器lcc的合作者。在这个项目中，他负责提供高度

可复用的基础架构。在不断的实践中，他完全使用ANSI C形成了一整套可复用组件库。这套组件库架构清晰，性能优异，而且提供了很多高级的特性，比如类Win32 SEH的异常处理机制、可移植的线程库、高性能的内存池、丰富的可复用数据结构组件。David Hanson把他在创作这些组件的过程中所积累的心得以及对其源码的精致剖析原原本本地写在了这本书里。这样的著作，当然堪称C语言领域里的铭心绝品。难怪已故著名技术作家Richard Stevens对此书赞不绝口，他说：“这本书中的技术，对于大部分C程序员来说，已经遗忘得太久了。”对于希望能在C语言应用上达到最高层次的核心程序员而言，这本书是难得的必读之作。

其他

除了以上四本书以外，我还乐意推荐Andrew Koenig的著作《C陷阱和缺陷》（人民邮电出版社）和Deitel父子合著的C How to Program两本书。



C陷阱和缺陷

Andrew Koenig是世界上屈指可数的C++专家，他的这本书可能是最薄的一本C语言经典。它简明扼要地讲述了C程序设计中的陷阱和缺陷，包括词法陷阱、语法陷阱、语义陷阱、连接、库函数、预处理器以及可移植性缺陷等，最后一章还给出了关于如何减少程序错误的建议以及前面各章问题的参考答案。尽管这个小册子成书于C89标准制定之前，然而，即使到了C99早已颁布的今天，书中提到的大多数陷阱和缺陷一如十五年前那样使我们警醒。



C程序设计教程

Deitel父子合著的C How to Program一直是非常好的C语言入门教程，我手头的中译本名为《C程序设计教程》（机械工业出版社出版，原书第二版）。除了对技术的正规描述（辅以许多简明扼要的例子）外，每一章后面都带有小结、术语、常见的程序设计错误、良好的程序设

计习惯、性能忠告、可移植性忠告、软件工程评述、自我测验练习及答案等。整书内容清晰，组织良好，易于阅读和理解。值得一提的是，有许多入门书读完一遍即可扔掉，而这一本是个例外。

结语

以上这几本书很大程度上局限于描述C语言及标准库本身，如果希望学习更专业的、领域相关的C程序设计技术，你可以在选择阅读这几本书的基础上继续查阅专门书籍。此外，在C语言书籍领域，水平不相上下的作品有很多，这几本只是根据我自身的阅读体验而做出的推荐。

我并不是一名C程序员，我的大多数时间都花在C的后裔语言尤其是C++身上了。然而，正如你知道的那样，C++并不全是类和模板，不全是面向对象和泛型编程，在较低的层面，C++的各个实现一如既往在很大程度上兼容C。C++和C的亲密关系决定了每一本C语言经典都应该是C++程序员的读物。

这世界变化得真是太快。各色新事物层出不穷，让人眼花缭乱、目不暇接。人们也很容易被那些五光十色的花哨玩意儿所吸引，常常会忘记构筑过去、现在和未来的坚实基础。C语言就是这样的基础之一。今天，世界上有许多我们看不见的重要软件是用C语言（以及C++等）编写而成的，正是它们在默默地支撑着这个信息世界的运转。

04年第2期：荣耀——C++程序设计之四书五经（上）

<http://book.csdn.net> 2006-7-21 19:32:00

C++程序设计之四书五经（上篇）

荣耀/文

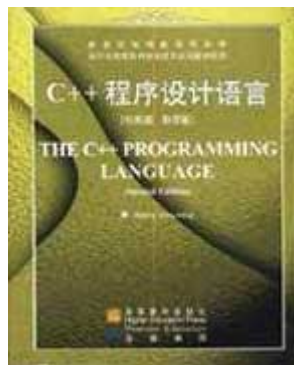
C++是一门广泛用于工业软件开发的大型语言。它自身的复杂性和解决现实问题的能力，使其极具学术研究价值和工业价值。和C语言一样，C++已经在许多重要的领域大获成功。

然而，一个不可否认的现实是，在低阶程序设计领域，C++挤压着C同时也在承受着C的强烈反弹，而在高阶程序设计领域，Java和C#正在不断蚕食着C++的地盘。也许C++与C合为一体永远都是一个梦想，也许Java和C#的狂潮终将迫使C++回归本位——回到它有着根本性优势的开发领域：低级系统程序设计、高级大规模高性能应用设计、嵌入式程序设计以及数值科学计算等。果真如此，我认为这未尝不是一件好事。

C++吸引如此之多的智力投入，以至于这个领域的优秀作品，包括重量级的软件产品、程序库以及书籍等，数不胜数。文题“C++程序设计之四书五经”一个不太严格的含义是：C++程序设计之四书×五经。是的，在本文（及其下篇）中，我将分门别类推荐20多本C++好书，你可以根据自己的需要选读。

TCPL和D&E

TCPL和D&E分别是《The C++ Programming Language》和《The Design and Evolution of C++》的简称，均出自Bjarne Stroustrup之手。我将它们单列出来，首先是因为Bjarne是C++语言的创建者，然后是因为比“首先”那个原因更重要的原因：这两本书是C++领域毋庸置疑的杰作。说它们是C++语言圣经，并不为过。

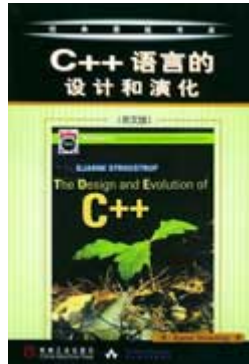


Bjarne Stroustrup, 《C++程序设计语言》影印版、中文版、题解

迄今为止，TCPL是除了C++标准文献之外最权威的C++参考手册。和大多数人的看法不大一样，我认为Bjarne的文字语言并不逊色于他所创建的程序语言，至少我喜欢这种学院气息浓厚的作品。本书对C++语言的描述轮廓鲜明、直截了当。它从C++语言创建者的角度来观察C++，这是任何别的作者和书籍做不到的——没有任何人比Bjarne自己更清楚该怎么来使用C++。

这是一本严肃的著作，以中、高级C++开发人员为目标读者。如果你是一名有经验的C++程序员，需要了解更加本质的C++知识，本书正是为你而写。它不是那种让你看了会不断窃喜的小书，需要用心体会，反复咀嚼。在阅读过程中，请特别留心Bjarne先生强调了什么，又对什么一语带过。我个人比较喜欢这本书的第四部分“使用C++做设计”，这样的内容在类似的程序

设计语言书籍中很难看到——我甚至认为Bjarne应该将这部分独立出来单独写一本书。



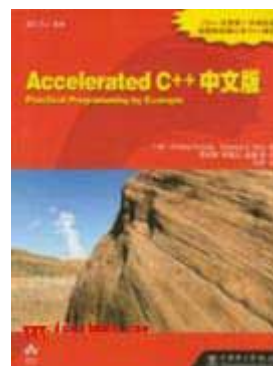
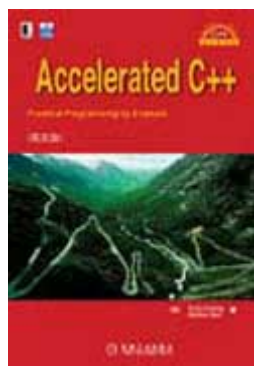
Bjarne Stroustrup, 《C++语言的设计和演化》影印版、中文版

D&E是一本关于C++语言设计原理、设计决策和设计哲学的专著。它清晰地回答了C++为什么会成为今天这个样子而没有变成另外一种语言。作为C++语言的创建者，Bjarne淋漓尽致地展示了他独到而深刻的见解。除了广受赞誉的语言特性外，Bjarne没有回避那些引起争议的甚至被拒绝的C++特性，他一一给出了逻辑严密、令人信服的解释。内容涵盖C++的史前时代、带类的C、C++的设计规则、标准化、库、内存管理、多重继承、模板等，对包括异常机制、运行时类型信息和名字空间在内的重要的新特性都分别进行了深入探讨。每一名C++程序员都应该可以从Bjarne的阐释中加深对手中这门语言的认识。

需要再次提醒的是，这两本书知识浓缩，信息量极大，请不要错过Bjarne每一句看似漫不经意的话。

入门教程

学习任何一门语言都需要一个从入门到精通、从新手到高手循序渐进的过程。不过，对于一个所谓的新手而言，究竟是一个完完全全的新手，还是一个熟悉某种别的语言的“新手”，甚至是在某种语言程序设计领域已经颇有建树的高手，很难一概而论？不同的C++新手需要不同的入门书籍。



Andrew Koenig, Barbara E. Moo, 《Accelerated C++》影印版、中文版

和市面上大多数C++教程不同，本书不是从“C++中的C”开始讲解，而是始于地道的C++特性。从一开始就使用标准库来写程序，随着讲述的逐渐深入，又一一解释这些标准库组件所依赖的基础概念。另外，和其他C++教材不同的是，这本书以实例拉动语言和标准库的讲解，对后两者的讲解是为了给实例程序提供支持，而不是像绝大多数C++教材那样，例子只是用作演示语言特性和标准库用法的辅助工具。

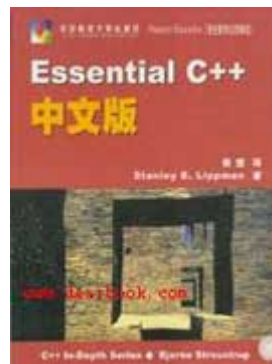
作者在C++领域的编程实践、教育培训以及技术写作方面都是世界一流水准。我喜欢这种大量使用标准库和C++语言原生特性的清新的写作风格。在这本教材面前，几乎迄今为止的所有C++教材都黯然失色或显得过时。尽管这本教材也许对于国内的高校教育来说有些前卫，不过我仍然极力向我的同行们推荐。顺带一提，在Bjarne和我最近的一封通信里，他这样评价本书：对于有经验的程序员学习C++而言，这本书可能是世界上最好的一本。



Stanley B. Lippman, Josee Lajoie, 《C++ Primer》影印第三版、中文第四版

这本书的名字多少有点让人误解。尽管作者声称这本书是为C++新手而写，但无论是它的厚度还是讲解的深度都暴露了似乎并非如此。也许说它是一本“从入门到精通”的C++教程会更合适一些。我个人认为它并不适合完全不懂C++的初学者——在阅读这本书之前，你至少应该先有那么一点C或C++的背景知识，或者至少要具有一些其他语言的编程经验。

尽管这本书省略了一些高级C++特性的讨论，但仍然可以称得上是迄今为止最全面的C++学习教程。事实上，如果一名C++初学者能够扎扎实实地读完本书并对照《C++ Primer Answer Book》完成全部习题的话，他的水平肯定可以进入职业C++程序员的行列。我个人认为，即使你已经拥有了TCPL，这本书依然有拥有的价值，因为在许多方面它比TCPL来得更详细、更易懂。



Stanley B. Lippman, 《Essential C++》影印版、候捷译文版

可以不太严格地认为这本书是《C++ Primer》的精简版。本书一一讲述了C++中最具代表性的主题，包括过程式编程、泛型编程、基于对象编程、面向对象编程、模板编程以及异常处理等。Stanley将门槛调低到“具有其他语言程序设计经验”的C++新手所能接受的最基本的层次，使他们能够迅速开始使用C++编程而又免于阅读《C++ Primer》那样的大部头。它以实例引导学习，力图使读者在最短的时间内把握C++的精髓。

也许换一个人来概述C++编程范型（paradigm）的方方面面需要好几百页才能说清楚，但

这本小书不可思议地做到了这一点。我个人非常喜欢这种满是技术、简明扼要并且“有话好好说”的书。这本书同样具有一个明显的风格：所有程序例子全部采用标准库组件，让人耳目一新。

以上三本书都不是为了完完全全的编程新手而写。完全的C++编程新手可以阅读Francis Glassborow的新书（尚未出版）：《A Beginners Introduction to Computer Programming: You Can Do It!》。这也是Bjarne的推荐。Francis Glassborow是ACCU主席，多年来他对几乎每一本C++经典名著评头论足，他自己的这一本自然会引起C++社群的极大兴趣。

高效、健壮编程

两年前我在负责一个省级电力调度系统项目时编写了一个网关程序，它从SCADA系统获取电力实时信息。通讯接口采用了不常用的数据库直连方式（这个网关程序一端连接SQL Server 6.5，另一端连接Oracle 8.1.6）。由于实时测点近万，每次将全部取样更新或插入一遍显然是低效的。我在网关程序里建了一个内存库，获取到的数据首先在其中进行比较，然后决定是否更新物理数据库（同时还做了别的更复杂的事情……），从而在效率和资源占用两方面达到了预期效果。

这个程序一直运行得很好，但在离开现场之后的某一天，系统管理员打来电话，说大概因为网络故障等原因，有时这个网关程序会崩溃掉——它自己崩掉也就罢了，问题是它还会把Windows 2000 Advanced Server搞成“蓝屏”！（坦白地说，我还从来没看过哪个非蓄意的程序有这个“能耐”）由于当时正忙于另外一个大项目，无法去现场调试，最后只有凭经验对内存库代码小心翼翼地封装以异常处理代码（同时也做了一些别的修改。这个程序本来不乏异常处理，可惜在开发调试期，很难模拟出真实的“异常”状况，以便验证那些异常处理代码真的可以工作）。这样，虽然没有彻底解决问题，但程序终究不再死得那么难看了。

在这儿讲这么一段花絮有什么意思呢（当初为那个可怕的bug朝思暮想时我可不认为这是一个“花絮”）？我想说的是，对于任何软件而言，离开强健，效率也就无从谈起。而对于C++程序员来说，也许编写一个高效的程序并不难，但要编写一个需要7×24小时持续运行的服务端软件就不是那么容易了（实际上，只要应用服务器不当机，即使发生网络故障，即使数据库服务器当掉，那个网关程序也应该有能力持续运行下去），需要考虑许多因素，有时这些因素甚至远远超出C++语言和开发工具的本身。作为一名开发实际项目软件的程序员，并非非得自己碰钉子才能积累经验，只要我们足够虚心，别人的经验往往都是我们很好的借鉴。鉴于此，我推荐以下几本书供你选读，它们可以让你从强健和效率两方面受益（当然了，它们涵盖的内容远不限于异常处理）。



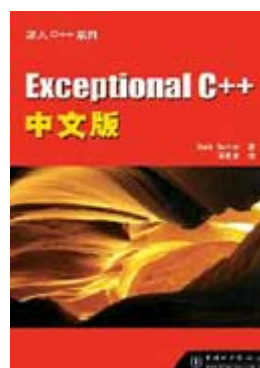
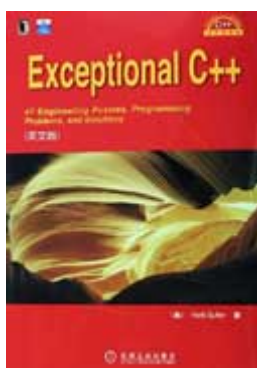
Scott Meyers, 《Effective C++》英文原版（二版、三版），候捷中文版（二版、三版）



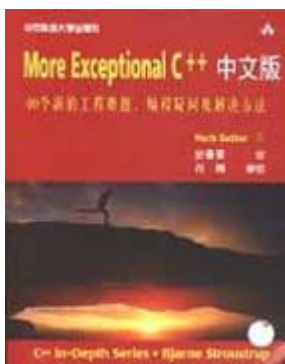
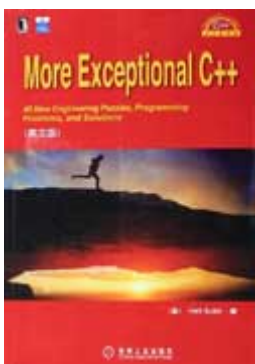
Scott Meyers, 《More Effective C++》英文原版、候捷中文版

如果说《Effective C++》主要讨论C++中一些相对基础的概念和技巧的话，那么《More Effective C++》则着重探讨了包括异常处理在内的一系列高级技术。与前者相比，后者具有两大主要区别:其一，它包含很多时新的标准C++的内容；第二，它讨论的主题倾向于“战略化”而非“战术化”，并且讨论得更深入、更彻底。尤其是对虚析构函数、智能指针、引用计数以及代理类（proxy classe）等技术和模式论述的深入程度，让人很难想象是出现于这样的一本小书之中。

游刃有余的技术，高超的写作技巧，Scott无疑是世界上最优秀的C++技术作家之一。在简洁、清晰、易读等方面，这两本书都卓尔不群。总之，Scott提供的这85个可以改善编程技术和设计思维的方法，都是中、高级C++程序员必备的技能。我强烈推荐这两本书（实际上还有一本，稍后就会看到）。



Herb Sutter, 《Exceptional C++》英文版、中文版



Herb Sutter, 《More Exceptional C++中文版》英文版、中文版

你自认为是一名C++语言专家吗？读一读ISO C++标准委员会秘书长的这两本书再回答。在这两本书中，Herb采用了“问答”的方式指导你学习C++语言特性。对于每一个专题，Herb首先合理地设想出你的疑问和困惑，接着又猜测出你十有八九是错误的解答，然后给你以指点并提出最佳解决方案，最后还归纳出解决类似问题的普适性原则。

这两本书是典型的深究C++语言细节的著作，很薄，但内容密集，远远超过Scott的那两本书，读起来很费脑筋——我个人认为它们要比Scott的书难懂得多。若要研习这两本书所包含的知识，至少需要花费数月的时间！（在Scott的荐序中，他坦陈不止一次陷入GotW问题的陷阱，你应该知道这意味着什么）对于语言细节的深究有什么好处呢？尽管在大多数情况下，我们不必关心C++代码幕后的动作，然而当我们不得不关心时，这两本书可以为我们提供很好的线索，因为它们揭示了C++语言中微妙而又至关重要的东西。



Stephen C. Dewhurst, 《C++程序设计陷阱》，中国青年出版社

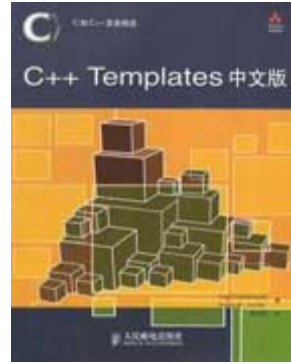
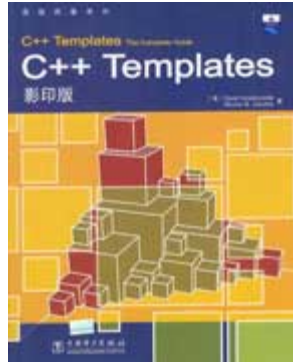
Stephen的理论素养和实践经验注定这是一本值得一读的好书。Stephen曾经是贝尔实验室中第一批C++使用者。他已经使用C++成功解决了包括编译器、证券交易、电子商务以及嵌入式系统等领域中的问题。本书汇集了作者来自开发一线的99条编程真知灼见，洞悉它们，你可以避免几乎所有常见的C++设计和编程问题。

我甚至认为，对于C++编程菜鸟而言，阅读这本书会比阅读Scott和Herb的书更能轻松而立竿见影地获得更大的提高。我个人很喜欢这本书的写作风格——Stephen的许多观点看似极端却无可辩驳。当然了，这种自信（以及冷幽默）来自于作者深厚的技术素养，而非自大的偏执。

除了上面推荐的书籍外，Dov Bulka和 David Mayhew合著的《Efficient C++: Performance Programming Techniques》（《提高C++性能的编程技术》，清华大学出版社）也值得一看。这本超薄小书聚焦于高性能C++应用程序开发。两位作者都是IBM软件专家，都工作于对性能要求极高的系统构建领域，本书是他们的经验之谈。也有人不喜欢这本书，因为它花了不少的篇幅讲述和C++无关的东西，我却恰恰因为这一点而对这本书产生好感，正是这些东西让我开阔了眼界。

模板和泛型编程

模板和基于模板的泛型编程无疑是当今发展最活跃的C++程序设计技术。模板的第一个革命性的应用是STL，它将模板技术在泛型容器和算法领域的运用展现得淋漓尽致，而Boost、Loki等现代程序库则将模板技术的潜能不断发挥到极致。在模板和泛型编程领域，我推荐以下两本重量级著作。



David Vandevoorde, Nicolai M. Josuttis, 《C++ Templates》影印版、中文版

有一种老套的赞美一本书的手法，大致是“没有看过这本书，你就怎么怎么地”，这里面往往夸张的成分居多。不过，倘若说“没有看过《C++ Templates: The Complete Guide》，你就不可能精通C++模板编程”，那么这个论断对于世界上绝大多数C++程序员来说是成立的。

这本书填补了C++模板书籍领域由来已久的空白。此前，上有《Modern C++ Design》这样的专注于模板高级编程技术和泛型模式的著作，下有《The C++ Standard Library》这样的针对特定模板框架和组件的使用指南。然而，假如对模板机制缺乏深入的理解，你就很难“上下”自如。鉴于此，我向每一位渴望透彻理解C++模板技术的朋友推荐这本书。

这本书在大陆、台湾各有一个译本，但出自不同的译者之手。当你看到这篇文章时，两个译本应该都已经上市，对于读者来说当然也就多了一种选择。侯捷先生个人网站上开放了繁体译本大部分章节，不妨先睹为快。



Andrei Alexandrescu, 《C++设计新思维：泛型编程与设计模式之应用》影印版、中文版

你自认为是C++模板编程高手吗？请看过这本书再回答。这是一本出自天才之手令人敬畏的杰作。泛型模式，无限延伸你的视野，足以挑战任何一名C++程序员的思维极限。

这本书共分为两大部分，第一部分讨论了Loki程序库采用的基础技术以及一些高级语言特性，包括基于策略的类设计、模板局部特化、编译期断言、Typelist以及小型对象分配技术等。第二部分则着重介绍了Loki中的重要组件和泛型模式技术，包括泛化仿函数（Generalization Functor）、单件（Singleton）、智能指针、对象工厂（Object Factory）、抽象工厂（Abstract Factory）、访问者（Visitor）以及多方法（Multimethods）等。每一种技术都让人大开眼界，叹为观止。

在C++的学习方面，过犹不及往往成了不求甚解的借口。然而，面向对象并非C++的全部，模板和泛型编程亦占半壁江山。对于“严肃”的C++程序员而言，及时跟进这项早经例证的成功技

术，不失为明智之举。

结语

这些著作是如此大名鼎鼎，也许根本不缺我一个推荐。然而，纵然C++程序员队伍的发展壮大速度不像其他更时髦的语言那样迅速，新人进总是多于旧人出。除了热忱地欢迎新人，我个人认为到了对C++书籍进行“盘点”的时候了，并且希望这样的“盘点”有益于感兴趣的读者。请保持耐心和宽厚。在下篇中，我将继续介绍标准库、网络编程以及其他方面的C++好书。有好书相伴，这个冬天不会冷。

04年第3期：荣耀——C++程序设计之四书五经（下）

<http://book.csdn.net> 2006-7-21 19:32:00

C++程序设计之四书五经（下篇）

荣耀/文

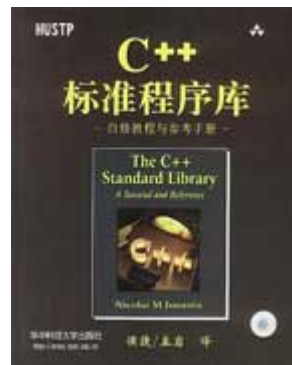
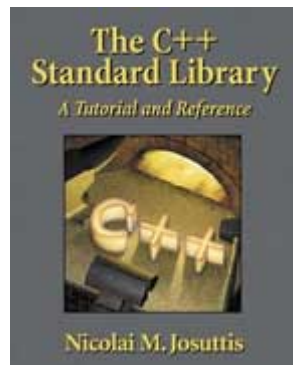
我在上篇中“盘点”了TCPL和D&E以及入门教程、高效和健壮编程、模板和泛型编程等方面共十几本C++好书。冬去春来，让我们继续C++书籍精彩之旅。

标准库

当我还在研究院工作时，与同院另外两家研究所合作开发过一个大型水利枢纽调度集成项目。我们三家软件系统之间都要相互通信。在调试通讯模块时，细心的客户（一名好学的系统管理员）发现对于同一通信规约的解释代码，我的不超过30行，而对方的则超过了150行且很难看懂。这位系统管理员很纳闷，我说大家编程风格和习惯不一样，我使用了标准库，而他使用了传统C编程风格以及他所习惯的另外一些技术。

别误会！我绝无贬低这位合作伙伴的意思。事实上，我对那些真正有着深厚的C编程功力的程序员常常怀有钦佩之心。毕竟，C++能有今天的成功在很大程度上缘于它深深地植根于C。作为一名C++程序员，倘若不熟悉C++中的C，我往往会认为他的基本功是不扎实的，他的技术底气是不足的。

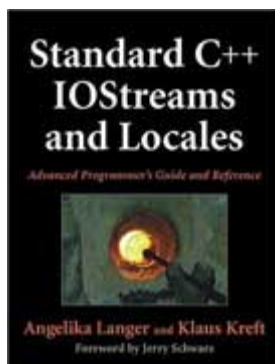
不过话又说回来，C++是一种多范型（paradigm）编程语言，具体采用哪种编程风格，专业程序员应该知道视具体情况而定。作为一名经常需要在现场做即兴开发的项目负责人，为了短平快地解决当务之急，我习惯尽量采用现有的库（和组件）。效率（以及强健性）久经验证的C++标准库已经摆在那儿了，何乐而不用呢？



Nicolai M. Josuttis, 《The C++ Standard Library: A Tutorial and Reference》原文版、中文版：
《C++标准程序库：自修教程与参考手册》

这是一本百科全书式的C++标准库著作，是一本需要一再查阅的参考大全。它在完备性、细致性以及精确性方面都是无与伦比的。本书详细介绍了每一标准库组件的规格和用法，内容涵盖包括流和本地化在内的整个标准库而不仅仅是STL。正如本书副标题所示，它首先适合作为教程阅读，尔后又可用作参考手册。

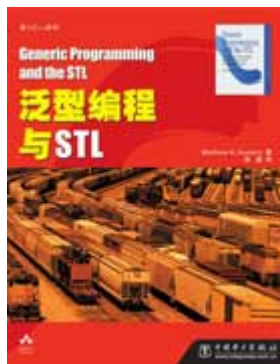
浅显易懂的写作风格使得这本书非常易读。如果你希望学习标准库的用法并尽可能地发挥其潜能，那你必须拥有这本书。正如网络上所言，这本书不仅仅应该摆在你的书橱中，更应该放到你的电脑桌上。我向每一位职业C++程序员强烈推荐。



Angelika Langer, Klaus Kreft,, 《Standard C++ IOStreams and Locales: Advanced Programmer's Guide and Reference》原文版、中文版《标准C++输入输出流与本地化》

C++标准库由STL、流和本地化三部分构成。关于STL的书市面上已经有不少，但罕见流和本地化方面的专著。本书是这两个领域中最优秀的一本，迄今为止没有任何一本书比这一本更全面详尽地讨论了流和本地化。如果你不满足于停留在“会用”流库的层面，千万不要错过它。

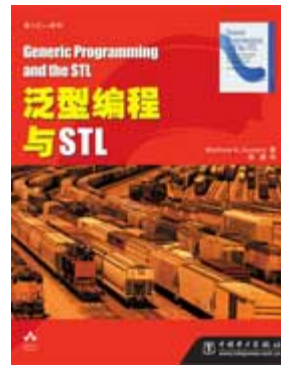
2001年夏天，我草草翻阅过这本书的中文版，从内容到包装都给我留下了比较深刻的印象——不过负面的居多一些。2003年秋天，无意中得知某网络书店正以超低价格甩卖这本书的中译本，情不自禁，一阵唏嘘。



Scott Meyers, 《Effective STL》影印版、中文版

读完Scott 的《Effective C++》和《More Effective C++》的中译本之后，我一直期待这本书的中文版。我从潘爱民先生的个人主页上了解到，他和他的合作伙伴似乎早已完成了这本书的翻译工作，可惜至今市面上仍不得见。幸运的是，我们可以看到它的原版。

本书是使用STL的程序员必读之作。在这本书中，Scott向我们讲述STL容器和算法的工作机制以及如何以最佳方式使用它们。和Scott的其他作品一样，这本书的写作风格清晰、精确，具有极佳的可读性。看过这本书以后，我想你也许会和我以及其他C++程序员一样产生这样的想法：Scott什么时候会写出一本“More Effective STL”？



Matthew H. Austern, 《Generic Programming and the STL: Using and Extending the C++ Standard Template Library》影印版、中文版《泛型编程与STL》

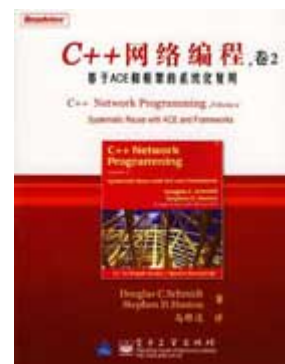
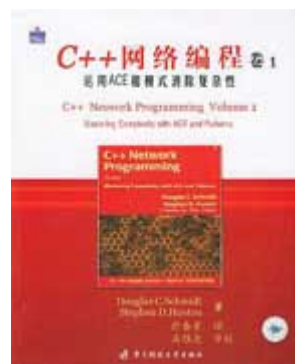
关于STL，我还提醒你留心Matthew H. Austern的《Generic Programming and the STL: Using and Extending the C++ Standard Template Library》（《泛型编程与STL》，中国电力出版社）。这本书散发着浓厚的学院气息。Andrew Koenig和Barbara Moo在《Accelerated C++: Practical Programming by Example》一书末尾郑重推荐另外两本进阶好书（除了他们自己的《Ruminations on C++》外），其中一本是TCPL，另外一本就是本书！

网络编程

在网络编程时代，C++应该扮演着怎样的角色，让ACE（Adaptive Communications Environment）来告诉你。



Douglas C. Schmidt, Stephen D. Huston, 《C++ Network Programming》Volume 1: Mastering Complexity with ACE and Patterns、Volume 2: Systematic Reuse with ACE and Frameworks



中文版：，《C++网络编程》卷1：运用ACE和模式消除复杂性、卷2：基于 ACE 和框架的系统

化复用

采用C++进行企业级网络编程，目前ACE（以及这两本书）是一个值得考虑的选择。ACE是一个面向对象、跨平台、开放源码的网络编程框架，目标在于构建高性能网络应用和中间件。Douglas是ACE的创始人，Stephen则已为ACE提供了数年的技术支持和顾问服务，两位都是ACE社群（是的，ACE的影响和实际应用的程度已经形成了一个社群）的专家。

ACE并不单单被大学和研究所追捧，它已经被成功地应用于世界上成千上万个商业应用中。在电信、宇航、医药和财经领域的网络系统中，ACE已经并继续发挥着重要的作用。如果你准备开发高性能通讯系统，你应该考虑考虑这一汇集世界顶尖专家智慧的成果。

除了使用C++面向对象设计技术和模板等高级语言特性外，ACE还运用了大量的模式。

《C++网络编程》卷1和卷2并不仅仅教你关于ACE的方方面面，它还会教给你模式和通用框架设计等高级技术等。所以，作为一名中、高级C++程序员，即使你很少进行正儿八经的C++网络程序设计，阅读这两本书同样可以从中受益。

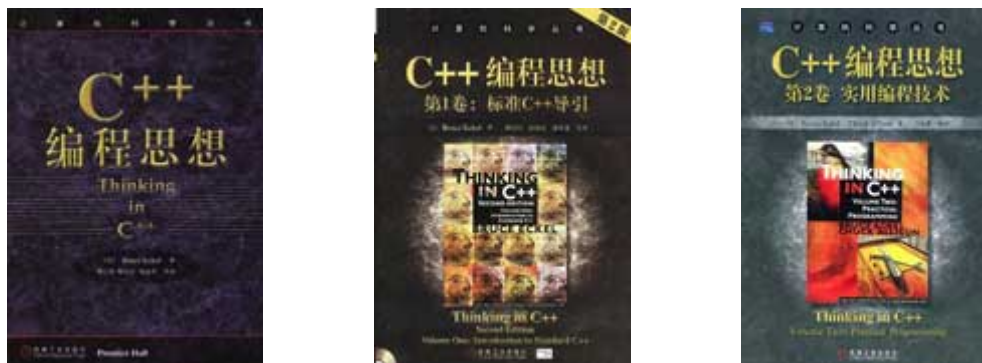
是的，并非所有网络应用都要使用Web服务器（以及其他应用服务器）和重量级组件模型，换个思路，它们或许也可以从轻量级的ACE组件中获益。

杂项

以下几本书所以被列入“杂项”单元，是因为我没有考虑到合适的归类方法，它们和上面的书籍一样，值得一读。



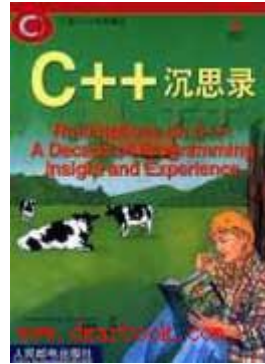
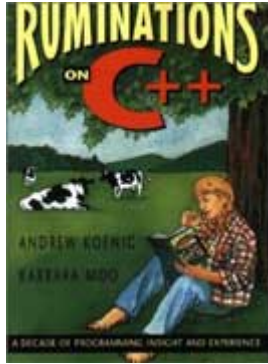
Bruce Eckel,《Thinking in C++》影印版二版、三版（又名卷二）



中文《C++编程思想》二版、卷一：标准C++导引 卷二：实用编程技术

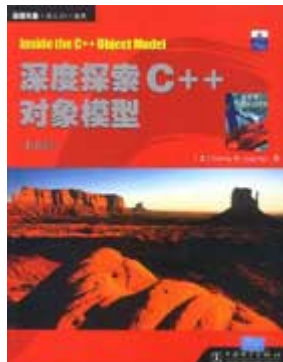
《Thinking in C++》的第1版于1996年荣获“软件研发”杂志评选的图书震撼大奖。最新推出的第2版对内容进行了大幅改写和调整，以反映C++标准化带来的影响以及近几年面向对象领域最新研究和实践成果。“输入输入流”、“多重继承”、“异常处理”和“运行时类型识别”等高级主题

连同C++标准化以后增加的一些内容则被放入第二卷中。Bruce是一名经验丰富的C++讲师和顾问，其培训和写作经验都是世界一流水准，他的作品比那些“玩票”的技术人员写的东西更能吸引读者。事实上，在同类图书中，对于大多数读者而言，这本书的可读性要超过TCPL和《C++ Primer》。顺带一提，访问作者的站点，你可以先睹第二卷的风采。



Andrew Koenig, Barbara E. Moo,, 《Ruminations on C++: A Decade of Programming Insight and Experience》原版、中文版《C++沉思录》

Andrew是世界上屈指可数的C++专家。这是一本关于C++编程思想和程序设计技术而非语言细节的著作。如果你已经具有一定的基础，这本书将教你在进行C++编程时应该怎样思考，应该如何表达解决方案。整本书技术表达透彻，文字通俗易懂。Bjarne这样评价这本书：本书遍布“C++是什么、C++能够做什么”的真知灼见。



Stanley B. Lippman, 《Inside The C++ Object Model》影印版、中文版《深度探索C++对象模型》

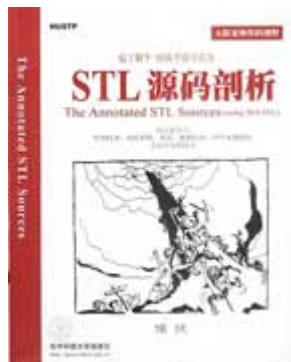
从编译器的角度观察C++可以使你知其然并知其所以然。本书探讨了大量的C++面向对象程序设计的底层运作机制，包括构造函数、函数、临时对象、继承、虚拟、模板的实例化、异常处理、运行期类型识别等，另外还介绍了一些在实现C++对象模型过程中做出的权衡折衷。喜欢刨根问底的C++程序员不要错过这本书。

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented software



Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, 《Design Patterns: Elements of Reusable Object-Oriented software》影印版、中文版《设计模式：可复用面向对象软件的基础》

设计可复用的面向对象的软件，你需要掌握设计模式。本书并非专为C++程序员而写，但它采用了C++（以及Smalltalk）作为主要示例语言，C++程序员尤其易于从中受益。四位作者都是国际公认的面向对象软件领域专家，他们将面向对象软件的设计经验作为设计模式详细记录下来。这本书影响是如此深远，以至于四位作者以及本书都被昵称为GoF（Gang of Four）。本书学院气息浓厚，行文风格严谨简洁，虽然它不如某些讲解模式的书籍易读，但真正要精准地理解设计模式，本书是终极权威。学习设计模式，这本书需要一而再、再而三的咀嚼。顺带一句：请将设计模式化作开拓思维的钥匙，切莫成为封闭思维的枷锁。



John Lakos, 《Large-Scale C++ Software Design》中文版《大规模C++程序设计》、侯捷：《STL 源码剖析》

还有一些C++好书值得一读，恕此处无法一一列出。例如John Lakos的著作《Large-Scale C++ Software Design》（《大规模C++程序设计》，中国电力出版社）和侯捷先生的《STL 源码剖析》（华中科技大学出版社）等。

《STL 源码剖析》是一本很有特色的书，但我认为它还可以更好。我个人期待侯捷先生自第一版发行以来经过对模板技术的沉淀和再思考之后，再写一本剖析得更深入、更透彻并且更全面的“第二版”。遗憾的是，侯捷先生在完成《C++ Templates: The Complete Guide》一书的翻译后似乎决定暂时告别模板、泛型编程和STL领域。

使用C++成功开发大规模软件系统，不仅需要很好地理解大多数C++书籍中讲述的逻辑设计问题，更需要掌握《大规模C++程序设计》中讲述的物理设计技术。当然，这本书的确有点过时了，不过，如果你的精力和金钱都比较宽绰，买一本看看并无坏处。

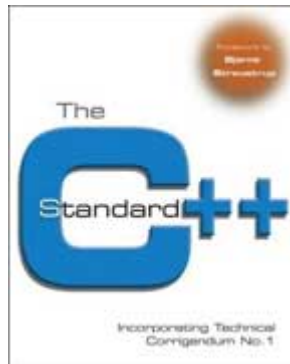
至此，我想有必要声明一下，有一些（好）书没有得到推荐，主要原因如下：

- 以上这些书已经足够多、足够好了。
- 我不会推荐通过正常渠道很难购买到的书籍 ——不管是中文版还是英文版。
- 作（译）者名气大小不影响我的推荐。我们是在看书，不是看人。
- 我不会推荐我从来没有看过的书。我至少要看过其中的某个版本（包括电子档）。这个“看”，一般指“认真阅读”，不过有一些也只能算是“浏览”。

结语

作为一名普通技术写译者，我深知技术创作和翻译的艰辛（和快乐），并多多少少了解一些有关技术书籍创作、翻译、制作、出版以及市场推介背后的细节。今天，我不会再对一本看上去差强人意的图书信口开河。罗列同一本书的各种版本的用意只在于为你多提供一些信息，让你多一种选择。

在本文成文的后期，我给Bjarne写了一封信，请教如果他来写这篇文章会怎么写。他给了我简明扼要的建议。在肯定以上列出的绝大部分图书都是世界顶尖水平的C++著作的同时，Bjarne提醒我别忘了向专家级程序员推荐《The C++ Standard: Incorporating Technical Corrigendum No. 1》



《The C++ Standard : Incorporating Technical Corrigendum No. 1》

Bjarne还友好地提醒我，在我的推荐列表中没有哪一本有助于C++程序员进行Windows编程——这正是我的本意。在这篇文章中，我只推荐、点评平台中立的C++著作（网络编程除外）——和操作系统无关，和集成开发环境无关，我甚至幻想它们和编译器也无关。你可以根据业务开发需要，选读自己喜爱的领域相关的C++书籍。

说到“系统无关、平台中立”，我不由得想起了“抽象层”的概念。开发实际应用的C++程序员通常工作于特定操作系统、特定开发环境和特定业务领域之中，而对标准C++和C++标准库扎实而深刻的把握，无疑是你得以在不同的操作系统、不同的开发环境以及不同的业务领域之间纵横驰骋的“抽象”本钱。

03第11期：孟岩——Win32系统编程之四书五经

<http://book.csdn.net> 2006-7-20 11:18:00

Win32系统编程之四书五经

孟岩/文

所谓的Win32开发，就是在C语言的层面上，直接使用Win32 API开发Windows应用程序或者系统程序。虽说现在直接用Win32 API开发应用程序的人已经不多，但是深入理解Windows系统程序设计原理，仍然是成为Windows开发高手的必经之路。所谓的Win32，其实是一个API规范，与UNIX系统编程接口标准POSIX是相对应的。从设计上讲，Win32绝对算不上是简洁优雅，因此学习Win32系统编程，相对来说是比较困难的。Win32来源于Win16，而Win16最初设计于1980年代中早期。设计Win16的时候，Microsoft还是一家小公司，应该说能力还有限。同时，Win16的主要目标是DOS之上GUI系统，设计上偏重GUI部分，在系统编程领域考虑不多。等到设计Win32时，微软能力增强了，有心要把Win32打造成第一流的操作系统API规范，但是已经背上了Win16的历史包袱。所以相对而言，Win32的设计实在赶不上POSIX那么简介优雅。微软在随后又先后进入了COM时代和.NET时代，每一次进步都需要继续背负以前的历史包袱，使得微软的技术体系越来越复杂。微软技术的复杂性，是微软与Java竞争中的最大弱势。而对我们程序员来说，无论你现在是在学习和使用最新的.NET技术，还是继续坚持COM开发模型，要克服微软技术的复杂性，必须上溯到Win32。那么Win32领域的“四书五经”是哪些？让我们一道来。

1. Windows程序设计



Charles Petzold, Windows程序设计（第五版） 北京大学出版社

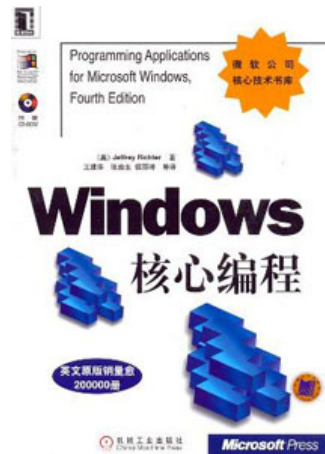
本书是尽人皆知的Win32 API编程经典，也称为“Petzold Book”，学习Win32 API编程的人几乎都从这本书入手。此书的优点是落墨细致，对基本概念和基本技巧的解释非常详尽，不厌其烦。同时，这本书篇幅不小，但是包容的东西并不多，基本上把重点放在GUI编程上和常规编程任务上，所以能够把很多问题讲得非常透彻。通常第一次阅读此书，会觉得作者行文有些罗嗦。但在日后的开发中，却发现，这些当时觉得罗嗦的论述，都是实际工作中值得反复查阅和咀嚼的宝贵资料。这就是所谓的Petzold风格，既可做tutorial，又可做reference。

几年前，人邮出过一本Charles Calvert的《21天学通Windows 95编程》，那本书的定位与

“Petzold Book”相似，而且写得也非常出色，在控件编程方面的阐述甚至超过了“Petzold Book”。可惜这本书完全被“Petzold Book”的光芒所覆盖。不仅国内没有再版过，而且原出版者也彻底打消了与“Petzold Book”对抗的念头。

现在仍然用Win32 API直接开发GUI的人，恐怕是少之又少，所以“Petzold Book”的意义不及当年。更公允的说，现在该书最大的意义是作为Win32的入门教材。不经过它的洗礼，很难升堂入室。

2、Windows核心编程



Jeffrey Richter, 《Windows核心编程》第四版 机械工业出版社

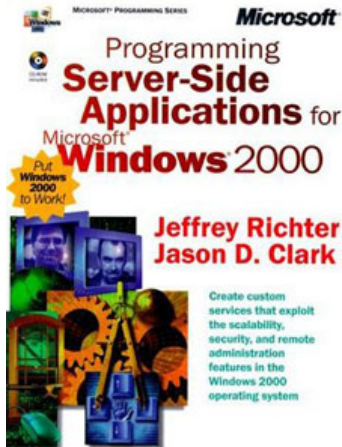
这本书的前三版名为Advanced Windows，第四版改名为Programming Applications for Microsoft Windows。作者在前言里还特别讲了改名的原因。但在我看来，这本书还是叫做Advanced Windows更贴切。因为这本书虽然同样是讲Win32 API，却几乎和Petzold Book没有任何重复。初级的内容完全略过不提，GUI部分也是一带而过，全书倾力于系统编程，一上来就讲Win32核心对象，从进程到线程，从DLL到SEH，从钩子到内存管理，可以说是不避艰险，迎难而上，将最复杂最微妙的细节完整展现出来，同时配备大量精心准备的例子，代码质量非常高超。因此，这本书从问世之日起，就成为Win32系统编程方面的圣经，作者Jeff Richter也成了Win32编程方面的绝对权威。想要学习Win32核心编程，不读这本书是无法想象的。遗憾的是，这本书的中文版翻译配不上原书的高水平，可以说是大错不多，小错不少。如果有读者要认真研读此书，可以找北京大学出版社出版的影印版。

这本书是Richter的代表作，也奠定了Richter世界顶级技术作家的地位。我本人非常喜欢读Richter的书，因为他的文字是写给专业人士看的，决不假装弱智，同时有相当体贴细致，非常符合我的习惯和胃口。所以通常在某个领域有了Richter的书，我就不会去看别人的书了。

对于某些读者来说，Marshall Brain的Win32: System Service 3/e（中文名《深入学习：Win32系统服务开发与实例》）和Johnson Hart的Win32 System Programming 2/e（中文名《Win32系统编程》）也是不错的选择。两书的论述都不如Richter深入，但是例子精当，讲解恰到好处。Brain的书覆盖面很广，例子非常有启发性，而Hart的书最适合从UNIX转到Win32的朋友。我发现自己在有空闲的时候喜欢啃Richter的书，但是在实践中需要快速学习解决方案时，反而Brain和Hart的书对我更有帮助。所以这两本书虽然不像Richter那么必需，但是也是非常实用的补充。

Petzold Book和Richter的这两本书，构成一个连贯一致的系列，一般Win32学习者精读并且基本掌握这两本书的主要思想和技术，就可以称得上Win32系统编程的专家了。如果说这两本书的弱点，主要是两书写作时以Win9X的开发作为一个基本出发点，因此对于Win2000平台的新特性介绍不足。特别是对于Windows 2000服务端高性能软件的开发，基本上没有涉及，很多Win32特有的机制也没有覆盖到。因此，如果想在Win32方面再上一层楼，可以考虑Jeffrey Richter的下面这本书。

3. Programming Server-side Application for Microsoft Windows 2000



Jeffrey Richter, Programming Server-side Application for Microsoft Windows 2000

知道I/O Completion Port吗？知道如何在多CPU服务器上把机器的能力发挥到极至吗？知道如何利用Windows的logging service吗？知道如何利用WMI开发服务端程序吗？即使你是一个Win32开发高手，也可能会对上述问题感到困难。随着一般应用程序的开发逐渐转移到Java、.NET或者Delphi这样的平台上，Win32能够完全发挥威力的最后一个角落，大概就是服务端高性能程序设计了。而如果你想要在这个领域有所建树，Jeffrey Richter的这本书就不可不读。这本书几乎是这个领域里唯一的一本书，而Jeffrey Richter的实力又保证了这本书的水平达到了相当的高度。读这本书给人以震撼的快感，你会感叹，Jeffrey Richter对于Windows系统的理解竟能够达到如此深度！

这本书跟上面提到的Petzold Book和《Windows核心编程》几乎又是零重叠，所涉及的内容都是高级技术，特别是自Windows 2000以来出现的服务器软件开发新技术。与前两本配合起来，形成了一条从基础到专家的完整链条。在当今业界，仍然使用纯Win32开发程序的开发者，有不少是在金字塔顶端的“高手”级人物。如果说Petzold Book在今天的主要意义在于教育，《Windows核心编程》的主要意义在于深化你对系统的理解，那么这本书中教授的知识是实实在在吃饭的家伙。因此，对于高手和有志最终成为高手的程序员来说，这本书是一本难得的经典。

03第12期：孟岩——TCP/IP网络编程之四书五经

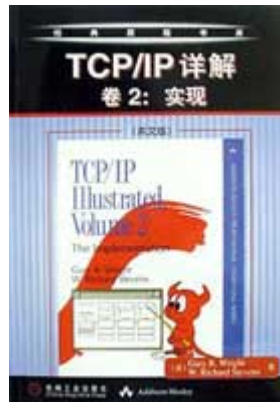
<http://book.csdn.net> 2006-7-20 11:18:00

TCP/IP网络编程之四书五经

孟岩/文

TCP/IP协议是目前广域网和局域网通用的网络协议，因此，基于TCP/IP的编程就格外重要。从应用上来说，现在直接利用C层次Socket API进行TCP/IP编程的人确实越来越少了，各种现成的框架（如ACE、Java、.NET FCL）和控件（如IP*Works）大大简化了TCP/IP编程的难度。但是，如果想要在这个领域达到融会贯通的地步，不打下坚实的基础是不可想象的。正如Richard Stevens在TCP/IP Illustrated中所说，在网络编程领域，开发者所遇到的实际问题中，大约有90%都与开发者对于TCP/IP的理解相关。高层的框架和控件总结了TCP/IP的主要使用模式，并且进行了抽象和封装，这固然非常好，但是要想真正掌握TCP/IP网络编程的真谛，恐怕需要认真学习下面的几本著作。

1. TCP/IP Illustrated, Vol.1 / Vol.2 / Vol.3

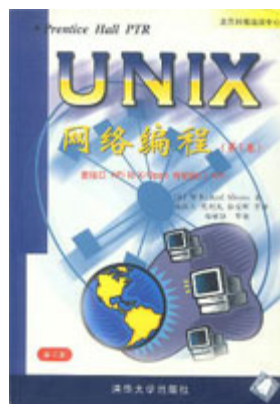
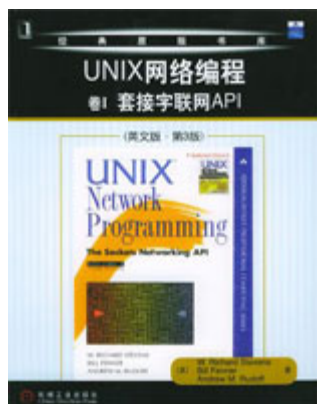


Richard Stevens, TCP/IP Illustrated英文版（卷一、卷二、卷三）、中文版（卷一、卷二、卷三） 机械工业出版社

这是一本众望所归的书。自从1994年出版以来，TCP/IP Illustrated就一直是TCP/IP领域具有传奇色彩的圣经典著作。这一套著作共分为三卷，第一卷通过实际动手实验的方式深入探讨TCP/IP协议；第二卷逐行解释了4.4BSD系统TCP/IP协议栈的15,000行实现代码；第三卷探讨了T/TCP和HTTP等高级协议。整套书最大的特点就是贴近实践，完全符合程序员的胃口，不空谈，不说教，用实验和代码来说明问题。写作这样的书，对于作者要求非常之高，即使在今天看来，敢于这么写作的作者也是凤毛麟角。Richard Stevens以他的写作风格和作品，在TCP/IP编程领域竖起一座丰碑，令其他作者高山仰止，心向往之。在下文将要提到的Effective TCP/IP Programming一书中，甚至把“读Stevens的书”专门列为一个条款。这套著作的地位，可见一斑。

学习这套书，与其说需要读，不如说需要做。Stevens最强调程序员的两项能力：一是读代码，二是动手实验。他在自己所有的书里都敦促读者将范例输入电脑，亲自编译、运行和调试。这当然需要花费大量的时间和精力，但是如果有读者能够扎扎实实地这么做，那么他所付出的每一分钟都将会有双倍的回报。

2. UNIX Network Programming, 2/e, Vol.1 / Vol.2



Richard Stevens, UNIX Network Programming 英文版（卷一、卷二）、中文版（卷一、卷二） 清华大学出版社

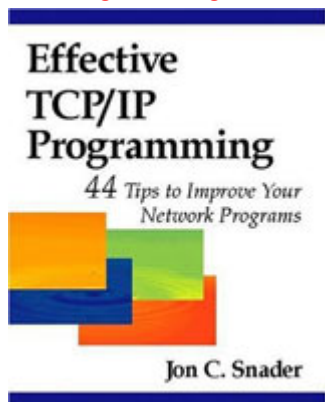
又是Richard Stevens，又是经典名著。这本书的第一版问世于1990年，是Stevens的开山之作，一举奠定了Stevens在网络编程领域的重要地位。如果说TCP/IP Illustrated解答了“什么是TCP/IP”和“TCP/IP是怎么实现的”两个问题，那么这本书则是在回答“怎样运用TCP/IP开发网络应用程序”的问题。显而易见，这本书的内容更加贴近普通程序员，因此，大多数人是先学习这本书，然后再深入学习TCP/IP Illustrated。

这也是一本让人伤感的书，Richard Stevens从1997年开始撰写此书第二版，制定了一个宏大的计划：卷1阐述Socket API和TLI API；卷2阐述除Socket之外的其他IPC机制；卷3阐述网络编程应用。应该说，卷1和卷2都是为了给卷3打基础。然而天妒英才，就在世人翘首企盼第三卷的时候，1999年9月1日，Richard Stevens英年早逝。留下这套未完成的巨著，让人感慨不已。不过最近从Prentice Hall传来好消息，出版社已经找到合适的作者，重新修订这套经典巨著，其中第一卷即将问世，并且已经在Amazon上热卖。相信会有当世高人沿着Stevens的足迹给我们续写第三卷的，让我们拭目以待。

可选图书：Douglas E. Comer, Internetworking with TCP/IP, Vol.1 / Vol.2 / Vol.3。

这套书也是三卷，前两卷内容与TCP/IP Illustrated相似，最后一卷则偏重应用程序的开发。总体上来看，这三卷书的内容与上述Stevens的两套著作相当。Comer教授的这套书定位是教材，因此写出来中规中矩，寓精彩于平和之中。读者如果喜欢阅读教材风格的书，这套应该是很好的选择。

3. Effective TCP/IP Programming



Jon C. Snader, Effective TCP/IP Programming、中文版：高级TCP/IP编程

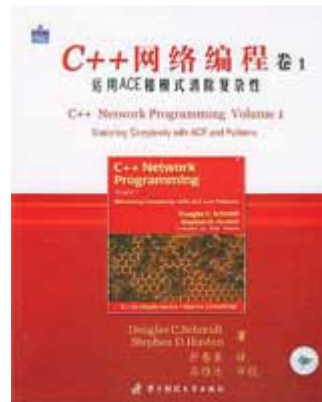
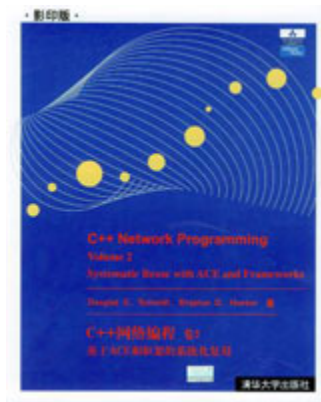
自彻读Effective C++之后，我总结出一条经验。无论在哪个领域中，一旦掌握基础知识，则应该迅速去寻找该领域中一本名为“Effective XXX”的著作并阅读之，一定可取得事半功倍的效果。Effective C++如此，Effective STL如此，Effective Java如此，这本Effective TCP/IP同样如此，也是一剂功效强大的催化剂。

本书作者是经验丰富的TCP/IP程序员，他在这本书中总结了44个难得的TCP/IP开发经验，并且将每条经验背后的道理都详细地讲解。毫无疑问，这种“专家经验”型著作对于提高读者水平，能够起到最高效、最直接的作用。

初学TCP/IP的朋友可能觉得，自己应该先把基本知识熟练掌握，再看这本书，效果会更好。我认为不必如此，您可以在基本观念建立起来之后直接品读此书。这本书篇幅不大，起点不高，却对很多关键问题进行了全面而细致地介绍，在具体问题的阐述上颇有Richard Stevens之风，也是纯而又纯的程序员风格——拿代码说话。

系统学习和重点难点的突破相结合，历来是快速掌握一门技术的良方。由Scott Meyers开创的“Effective”之风，已有愈演愈烈之势，这是程序员们的巨大福音。我强烈建议大家，不要放过任何一本以“Effective”为名的技术书籍。

4. C++ Network Programming, Vol.1 / Vol.2



Douglas C. Schmidt, Stephen D. Huston, C++ Network Programming英文版（卷一、卷二）、中文版（卷一、卷二）

今天的网络编程环境是建立在C的基础之上。如果你满足于C层次的TCP/IP编程，那么Stevens的书可以说达到了一个最高境界，你大可以登东山而小鲁。但是如果再上一个层次，考虑如何运用C++强大的抽象机制使网络编程变得更加富于艺术性，更加高效可靠，那么恐怕必须要高度重视这套书。

这套书共两卷，篇幅都不大，主要内容是介绍ACE。在C++领域，由于这套书被Bjarne Stroustrup编入C++ In Depth系列，因此具有崇高的地位。而在网络编程领域，也有人称赞此书为Stevens之后最重要的网络编程著作。作者Douglas Schmidt是C++、网络程序设计和模式领域国际公认的权威专家，潜心研究C++网络编程已经十多年。他发起的ACE开源项目是目前当之无愧的最优秀和最成功的C++网络编程环境，不仅具有非常出色的可移植性，而且性能卓越。美国国防部的ARPA已经支持这个项目好几年了，其最终目的是在ACE的基础上建立可用于现代战争的实时分布式系统。毫无疑问，ACE无论在理论上还是在实践中，都是最值得程序

员关注的网络编程典范项目，而这一套两卷对ACE的介绍性书籍，也毫无疑问是最值得网络程序员品读分析的经典。学习这两本书，你可以将经典的基于C API的网络编程知识与现代面向对象、模式和框架等最新技术思想结合起来，从而使你对网络编程的认识上升到一个新的高度。

TCP/IP是计算机网络协议中的一族，我们学习TCP/IP，是为了更好的进行网络编程。而为了更好地学习网络编程，还应该打下扎实的计算机网络理论基础。虽说“计算机网络”书籍应该是另外一个主题，但是如果考虑到与TCP/IP的学习相配合，我倒想在这里额外推荐一本相关的书籍：



Larry Patterson & Bruce Davie, Computer Networks: Systems Approach Second Edition

这本书的作者是普林斯顿大学教授，对于网络系统有极为深刻的理解。作品的最大特点是以非常系统的方式介绍了网络系统的“why”和“how”，整本书思维非常清晰，读来连贯一致，欲罢不能，而对于提高你的TCP/IP编程能力，有着非常切实的作用。