

程序设计实践



[程序设计实践 下载链接1](#)

著者:[美] BRIAN W.KERNIG

出版者:机械工业出版社

出版时间:2007-1

装帧:简装本

isbn:9787111211273

从排错、测试、性能、可移植性、设计、界面、风格和记法等方面，讨论了程序设计中实际的、又是非常深刻和具有广泛意义的思想、技术和方法，它的翻译出版将填补国内目前这方面书籍的空白。《程序设计实践》（双语版）值得每个梦想并努力使自己成为优秀程序员的人参考，值得每个计算机专业的学生和计算机工作者阅读，也可作为程序设计高级课程的教材或参考书。

作者介绍:

Brian

W.Kernighan，贝尔实验室计算科学研究中心高级研究人员，著名的计算机科学家。他参加了UNIX系统、C语言、AWK语言和许多其他系统的开发，同时出版了许多在计算机领域具有影响的著作，包括《The Elements of Programming Style》、《The Practice of Programming》、《The UNIX Programming Environment》、《The AWK Language》、《Software Tools》等。

目录: 译者序

Preface / 前言

Chapter 1: Style / 风格

1.1 Names / 名字

- 1.2 Expressions and Statements / 表达式和语句
- 1.3 Consistency and Idioms / 一致性和习惯用法
- 1.4 Function Macros / 函数宏
- 1.5 Magic Numbers / 神秘的数
- 1.6 Comments / 注释
- 1.7 Why Bother? / 为何对此费心
- Chapter 2: Algorithms and Data Structures / 算法与数据结构
 - 2.1 Searching / 检索
 - 2.2 Sorting / 排序
 - 2.3 Libraries / 库
 - 2.4 A Java Quicksort / 一个Java快速排序
 - 2.5 O-Notation / 大O记法
 - 2.6 Growing Arrays / 可增长数组
 - 2.7 Lists / 表
 - 2.8 Trees / 树
 - 2.9 Hash Tables / 散列表
 - 2.10 Summary / 小结
- Chapter 3: Design and Implementation / 设计与实现
 - 3.1 The Markov Chain Algorithm / 马尔可夫链算法
 - 3.2 Data Structure Alternatives / 数据结构的选择
 - 3.3 Building the Data Structure in C / 在C中构造数据结构
 - 3.4 Generating Output / 生成输出
 - 3.5 Java / Java
 - 3.6 C++ / C++
 - 3.7 Awk and Perl / Awk and Perl
 - 3.8 Performance / 性能
 - 3.9 Lessons / 经验教训
- Chapter 4: Interfaces / 界面
 - 4.1 Comma-Separated Values / 逗号分隔的值
 - 4.2 A Prototype Library / 一个原型库
 - 4.3 A Library for Others / 为别人用的库
 - 4.4 A C++ Implementation / C++实现
 - 4.5 Interface Principles / 界面原则
 - 4.6 Resource Management / 资源管理
 - 4.7 Abort, Retry, Fail? / 终止、重试或失败
 - 4.8 User Interfaces / 用户界面
- Chapter 5: Debugging / 排错
 - 5.1 Debugger / 排错系统
 - 5.2 Good Clues, Easy Bugs / 好线索, 简单错误
 - 5.3 No Clues, Hard Bugs / 无线索, 难办的错误
 - 5.4 Last Resorts / 最后的手段
 - 5.5 Non-reproducible Bugs / 不可重现的错误
 - 5.6 Debugging Tools / 排错工具
 - 5.7 Other People's Bugs / 其他人的程序错误
 - 5.8 Summary / 小结
- Chapter 6: Testing / 测试
 - 6.1 Test as You Write the Code / 在编码过程中测试
 - 6.2 Systematic Testing / 系统化测试
 - 6.3 Test Automation / 测试自动化
 - 6.4 Test Scaffolds / 测试台
 - 6.5 Stress Tests / 应力测试
 - 6.6 Tips for Testing / 测试秘诀
 - 6.7 Who Does the Testing? / 谁来测试
 - 6.8 Testing the Markov Program / 测试马尔可夫程序

6.9 Summary／小结
Chapter 7: Performance／性能
7.1 A Bottleneck／瓶颈
7.2 Timing and Profiling／计时和轮廓
7.3 Strategies for Speed／加速策略
7.4 Tuning the Code／代码调整
7.5 Space Efficiency／空间效率
7.6 Estimation／估计
7.7 Summary／小结
Chapter 8: Portability／可移植性
8.1 Language／语言
8.2 Header and Libraries／头文件和库
8.3 Program Organization／程序组织
8.4 Isolation／隔离
8.5 Data Exchange／数据交换
8.6 Byte Order／字节序
8.7 Portability and Upgrade／可移植性和升级
8.8 Internationalization／国际化
8.9 Summary／小结
Chapter 9: Notation／记法
9.1 Formatting Data／数据格式
9.2 Regular Expression／正则表达式
9.3 Programmable Tools／可编程工具
9.4 Interpreter, Compiler and Virtual Machines
／解释器、编译器和虚拟机
9.5 Programs that Write Programs／写程序的程序
9.6 Using Macros to Generate Code／用宏生成代码
9.7 Compiling on the Fly／运行中编译
Epilogue／后记
Appendix: Collected Rules／附录：规则汇编
．．．．．(收起)

[程序设计实践_下载链接1](#)

标签

程序设计

编程

计算机

Programming

计算机科学

编程方法论

软件开发

实践

评论

总而言之，只有熟悉了这个领域的工具和技术才能对特殊的问题提供正确解答，只有丰富的经验才能提供坚实的专业性结果：自动化，简单，清晰，通用才是计算机程序的本质。c++,Java都是C的直系衍生语言。更复杂的算法和数据结构都是几个基本的结构衍生的。学习低级语言的意义在于理解细节的机会增加了，而高级语言的正确使用是建立在更多细节或者更多底层语言的熟悉上的。隐藏在设计模式后面的基本思想是：大部分程序所采用的不过是很少几种不同的设计结构，与此类似，实际上也只有不多的几种基本数据结构。说的远一点，这与我们在第1章讨论过的编码习惯用法也是很相像的

比编程珠玑的实践性更强

要是本科的时候遇到这本书就好了，不过即使是现在读来还是有收获的。

我为什么不去看代码大全呢？

通过考察各种不同语言中的一些程序设计例子，从风格、算法和数据结构，到设计、调试、排错和性能改进，本书展现了许多具有普遍意义的工程概念。P.S.
很多方法的罗列没有结构化，搞不清它是怎么排布的。

TP311.1/4943

适合高级新手及以上

可能是自己写码太少，总感觉后几章有一些过时了，第三章值得多读几遍

程序设计方方面面，时隔多年，绝大多数东西没有过时，特别适合入行一两年经验的人（当然对于多年的也有收获）。不足：
1,翻译方面，一些名词由于译者没有采用约定俗成的译法导致理解不太顺畅，特别是“界面”一词。2,时效性，特别涉及语言细节的内容。

纯工程派作品，每章都拿出了大量代码作为实例，程序编写的所有方面都有所涉及

前1/4详读，后面一扫而过，等有时间再捡起C，再看。20160426

找机会还要再看一遍。

如果看这本书时，脑子里想的是“这不是废话么？当然得这样”，那也就达到本书作者的期望了。

书中涉及到编程的方方面面，能从整体上了解编程，扩展阅读又可针对某方面深入学习。不过这本书历史较久远，书中有些例子显得比较陈旧。

星星的书，去年过年就拿来了，马上今年又要过年了还没看过。。汗自己下

书上的内容很合理，而且对于写程序很有帮助，但是现实中会有各种条件的限制使得我

们无法完全按照书中的要求做。有些名词的翻译和现在的说法不同，如“interface”被翻译成了“界面”。总的来说，和编程珠玑差不多。

从初级到中级时必读的一本书!

所有程序员必读书

学技术

纯工程派作品，程序编写的所有方面都有涉及，不过书中所举例子显得比较陈旧

[程序设计实践 下载链接1](#)

书评

我看过很多类似#程序员必读XX本书#的书单，似乎在几乎所有的推荐书单里面，《程序设计实践》（英文名The Practice of Programming）都没有出现作为一本被推荐的书。我也是在今年暑假突然想开始读一些英文版的技术书籍，然后豆瓣上大致浏览了一下，机缘巧合的买下了这本书。在...

这是一本牢牢占据经典书架最起眼位置的小册子，也是最符合KISS之道的神作之一。抛开已有的业界的best practices，这本书还修正了我记忆中的很多错误认识，Brian Kernighan、Rob Pike的大名自然是本书最好的招牌；但作者的求实态度更值得每一个认真的programmer学习。整体上...

英语不好，第一次买英文书，没信心，正好发现这个评注版，欣然买之。然后发现所谓的评注不是针对难点的解释，大都是没用的，该解释的作者都已解释清楚了，而这本书又是写思想的不是技术，没有什么难点，评注看的反而碍眼，影响书原汁原味的感觉，

这本书40多块329页，而原版的才...

书非常好。但翻译却不能令人满意，就好像一个外行在翻译。
裘宗燕是计算机书籍翻译的前辈，也翻译了大量经典，但可能是由于翻译时年代久远，很多词汇都与现今流行的翻译方式不同。
另外这本书已经绝版了，所以只能找电子版来看。期待本书再版。
这里只举几个例子： 本书的翻译...

首先这个书评和原书内容几乎没有关系，针对这本评注版。
当初选择买评注版的原因在于，它是英文版的，就这么简单。但是看完后发现这版有一两个说小不小，但是确实恼人的缺陷。
其一是校对问题。书中出现很多单词排版错误，即两个单词连到一块去了，而且从第三章开始几乎每章...

你可曾…… 花费了大量时间编写错误的算法？ 使用了过于复杂的数据结构？
测试程序却忽略了明显的错误？ 需要让程序使用更少的内存，但运行快三倍？
纠结地在工作站和PC间移植程序？ 试图对他人的程序做适当的修改？
因无法理解程序而重写？ 这有趣吧？ 以上种种，程序...

足足看了一整天才翻到了最后的后记。
昨天告诉朋友在读此书，朋友看了目录说这个大而全，感觉不到有东西在里面。
说说自己的感受吧。 风格部分最简单，类似于代码大全中的一些。
算法，数据结构很简略，把树拿出来大概能吓跑一堆没经过学习的程序员。
马可夫链很好，立马去翻编...

推荐章节：第一，二，六，七章。
尤其是第一章，值得熟记并在以后的编程过程中加以运用。
其他章节可以以后再看。
注意有些部分的技巧可能过时了，不过大部分地方还是值得参考。
书的最后还有各章节的要点汇总，方便查询。

同样是出自大师之手，同样都是珠玑，这本书的名气实在太小。这就像做生意一样，生意好的店会越来越好，生意差的店只会越来越冷清。
但是，没有看过这本书，是自己的损失。
这本书有三好：1，实践性强；2，实践性强；3，实践性强。例如：第四章，将库函数的界面，作者首先通过...

这本书都是一些经验介绍，很好，如C的尽量少用函数宏等，很值得一看，有经验的程序员可以发现你以前的一些习惯可能会有问题，自己体会吧。

被大多数人推崇的书是不会让人失望的,虽然书中的有的思想,在看之前就已经有在应用了,但是看了之后仍觉得收获很大。书本的内容并不算多,但是仍然能感觉到一个集大成者在程序设计中的严谨分析,作者引据论典,以经典的实例层层剖析,让我们在实践中强化理论,印证理论。毕业以...

个人认为比《设计模式》好，不论是否使用OO，都会有所收获。最近又在重读，当思路遇到死胡同时，看一看往往会有新的想法。文笔够精炼，译笔又够流畅。总体来说相得益彰，不过个人不太喜欢把interface翻译成“界面”。

本书涉及到编程的各个环节, 结合这些书可以制定出一个软件研发的各环节checklist. 从团队的代码规范(命名, 结构与组织)到设计阶段的要求, 再到后面开发/调试/优化阶段的技术方法. 软件开发人员可以按这个图谱去加强各方面的技艺. 第一章命名部分, 值得学习与借鉴并写出规范. 并...

作者不是个美国人吗？怎么成中国科学院高级工程师了？译者和作者分不清吗？
 译者不是个美国人吗？怎么成中国科学院高级工程师了？译者和作者分不清吗？
 译者不是个美国人吗？怎么成中国科学院高级工程师了？译者和作者分不清吗？
 译者不是个美国人吗？怎么成中国科学院高级工程师了？译者和作者分不清吗？
 ...

现在看完了前面的第一章，对于第一章中的style非常有感触。程序毕竟是给人看的，如果风格不好，就没与人愿意去读。当然风格好不好是仁者见仁，智者见智的事情。自己由于也看过一些教你如何写出一些优美代码的书籍，书看多了，结果为了优美而优美，出现风格不一致的情况。这应...

推荐所有有一年编程以上的人看 真是句句血泪呀 不论语言不论平台 总有适合的部分 而且要做为备查的工具书 时不时看 一次不够还要多次看 嗯 考虑是不是买一本了 图书馆的几天就被人recall回去了

简短，精炼却又全面的程序员必读，涉及编码的各个方面，建议每个有一年左右编程经验的人都读一下，可以拓展视野，纠正问题。其中的例子十分经典，易于理解又不乏实用性。

就是最近流行的敏捷编程，很多概念这本书也体现得很清晰了。文档寓于代码之中。测试先行，渐进式开发和测试。甚至结对编程都有一个例子。

这本书从排错、测试、性能、可移植性、设计、界面、风格和记法等方面，讨论了程序设计中实际的、又是非常深刻和具有广泛意义的思想、技术和方法。

[程序设计实践 下载链接1](#)