

《程序设计实践》

图书基本信息

书名：《程序设计实践》

13位ISBN编号：9787111075738

10位ISBN编号：7111075730

出版时间：2000-8

出版社：机械工业出版社

作者：[美] Brian W. Kernighan, Rob Pike

页数：221

译者：裘宗燕

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《程序设计实践》

内容概要

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》

作者简介

Brian W.Kernighan和Rob Pike在朗讯科技贝尔实验室的计算机科学研究中心工作。Brian Kernighan是Addison-Wesley的“专业计算丛书”顾问编辑，也是《C程序设计语言》的合著者之一（与Dennis M.Ritchie合作）。Rob Pike是Plan 9和Inferno操作系统的主要结构设计与实现者，他的主要研究兴趣是如何帮助人们更容易地开发软件。

书籍目录

译者序

前言

第1章 风格

1.1 名字

1.2 表达式和语句

1.3 一致性和习惯用法

1.4 函数宏

1.5 神秘的数

1.6 注释

1.7 为何对此费心

第2章 算法与数据结构

2.1 检索

2.2 排序

2.3 库

2.4 一个Java快速排序

2.5 大O记法

2.6 可增长数组

2.7 表

2.8 树

2.9 散列表

2.10 小结

第3章 设计与实现

3.1 马尔可夫链算法

3.2 数据结构的选择

3.3 在C中构造数据结构

3.4 生成输出

3.5 Java

3.6 C++

3.7 Awk和Perl

3.8 性能

3.9 经验教训

第4章 界面

4.1 逗号分隔的值

4.2 一个原型库

4.3 为别人用的库

4.4 C++实现

4.5 界面原则

4.6 资源管理

4.7 终止、重试或失败

4.8 用户界面

第5章 排错

5.1 排错系统

5.2 好线索，简单错误

5.3 无线索，难办的错误

5.4 最后的手段

5.5 不可重现的错误

5.6 排错工具

5.7 其他人的程序错误

5.8 小结

第6章 测试

6.1 在编码过程中测试

6.2 系统化测试

6.3 测试自动化

6.4 测试台

6.5 应力测试

6.6 测试秘诀

6.7 谁来测试

6.8 测试马尔可夫程序

6.9 小结

第7章 性能

7.1 瓶颈

7.2 计时和轮廓

7.3 加速策略

7.4 代码调整

7.5 空间效率

7.6 估计

7.7 小结

第8章 可移植性

8.1 语言

8.2 头文件和库

8.3 程序组织

8.4 隔离

8.5 数据交换

8.6 字节序

8.7 可移植性和升级

8.8 国际化

8.9 小结

第9章 记法

9.1 数据格式

9.2 正则表达式

9.3 可编程工具

9.4 解释器、编译器和虚拟机

9.5 写程序的程序

9.6 用宏生成代码

9.7 运行中编译

后记

附录：规则汇编

索引

精彩短评

- 1、翻译的人感觉很熟，但是翻译的质量相当的挫，不过书的内容那是非常的好，很值得一读
- 2、看了第9章，和编译原理有点关系，实在非常精彩
- 3、没印象了，当你立志做一名程序员
- 4、C++语言的发明者的著作，语言语法设计的原理讲的相对其他书，更深入，理论性更强但不死板，可能看完了只能写出一段一段的代码，需要很长时间的联系消化，才能融会贯通，这时候也许算得上精通C++语法了吧，我还没到这地步
- 5、仔细读了一部分，后面大面积的c代码，看的比较粗略，以后再读一遍。
- 6、看这本书的时候还在学校，好怀念工大图书馆还有足球场..
- 7、短小精悍，信息量浓缩度之高，此类书籍无出其右。
- 8、其实还没有完全看完。是暑假的时候查了CMU的计算机课，顺便翻出来的这本书。里面的知识很浅，也很细致。在大一大二的时候就应该看的一本计算机编程基础书籍。
- 9、KISS神道之作，书虽薄，信息量却很大，一次性很难完全消化。
另外，可能是有些年代的原因，翻译版感觉不太顺畅。
- 10、个人认为柯尼汉是最好的计算机作家之一
- 11、程序开发对细节绝对不可忽视，好的程序可读性是非常重要的
- 12、多年编程之后才看到此书 已无惊艳之感
- 13、原书和译书质量都上乘
- 14、Princeton编程导论课程。因为编程主要不是编码费时间而是调试：程序首先给人读，偶尔的供计算机执行。本书主要介绍的是书写的规范，关键说了：数据结构和算法都是固定的而且很少，大多数是针对具体问题的组合和变形。
- 15、听说不错，就买过来看看，结果呢，超级棒，太精彩了，真知灼见，顶礼膜拜
- 16、新手码农最大的苦恼往往是：书看的太少而想到太多。就像书中说的：“好的设计是思考和经验的结合；纯粹靠玩命干不可能造就出好系统。”当然，对于老码农，也是一本不可多得的好书，阅读的过程中，你（好吧，其实是我）会一次次的跪谢大师，帮你把心中模糊的想法做出如此完美的系统性的总结。特别是第九章 记法（Notation）值得每个程序员认真读几遍。
- 17、读了英文原版，顺便把它重新排个版（LaTeX），为一些难懂的单词作了中文注释，如果大家有兴趣，可以在 github 上搜索 tpop，欢迎提出批评与建议。
- 18、厚积薄发
- 19、K.I.I.S贯穿全书
- 20、略读，代码未敲。
- 21、这本书能够对程序员有很好的指导作用。里面指出一些往往是容易被人忽略的重要的东西，例如命名规范，书写规范等。
- 22、反正越王后看越看不下去，没什么新鲜的内容了。
- 23、薄薄小书，绝对大收获。
- 24、真是编程实践方面的绝顶好书，无数的实际问题可以从这里找到答案或启发。工作中遇到不少问题，这本书中的经验总结完全适用。
- 25、前面还好，最后一章作者放大招了，叹为观止
- 26、都是干货，就是有点老
- 27、不错的非技术书。
- 28、值得一看
- 29、不要看
- 30、* 我初初撸了一遍，有启发，但是后来没有翻了，我要标记一下以后再去翻一遍_(:3
- 31、获益良多
- 32、首先这封面就不对，我读的版本是棕色封面。内容还不错，一些译法已经与今天不同，但别有古韵。
- 33、从这里我知道变量名的长度为什么是那么长。
- 34、书有点老，但有些启发。减一星是因为翻译:-(

《程序设计实践》

- 35、读得很快，写得很好，但是明显一遍不能吸收这本书的全部精髓。编程在于实践，这本书常看常新，在实践中运用相关规则
- 36、简练的程序设计指南，同类读物包括编程珠玑，程序员修炼之道。
- 37、语文没学好，看裘老的书都是不理解汉语想表达什么意思。
待工作以后，再在有体会的时候看看英文版的吧。
推荐阅读英文版，我个人认为读起来逻辑上要比中文的好理解，好组织书中的内容。中文的，需要你先懂汉语想表达什么，然后再去想背后想表达的知识是什么。很痛苦，每句汉语你都要认真读才知道说的什么意思！！
- 38、正如后记所言，软件行业日新月异，新工具层出不穷，而本书着力讲述的是那些持久的概念，书中讲的原则在当下仍未过时（大师的总结还真是一针见血啊）
p.s. 每章开头的引言也很妙，十分契合内容
- 39、这本应该在大学语言课程结束的时候读的，现在读到太晚了哎
- 40、适合刚工作时读，有利于培养好习惯。现在读来没有太多收获了（怪我自己太老了）
- 41、简短、精炼而有效。感谢现代硬件和现代编程语言，一般不用担心内存分配回收和代码可移植性的问题，但其余部分还是历久弥坚的，强力推荐！#应该成为全部大学生的第二本程序教材#
- 42、10块钱从二手书店买的。。好书
- 43、有人说:书上的知识和动画里的妹子,终究不是你的。
但是这些东西都可以花时间去把玩，挺好的。在校的时候觉得搬砖简单，想想认识得真肤浅。
- 44、《The practice of programming》的中文版，很古老的书，翻译的太。。。interface、inline等都翻译成啥了。。。对于内容，虽然现在才看，但还有启发
- 45、一般般吧，感觉可实践性不强。
- 46、第一遍
- 47、经验是智慧的结晶，要重视。
本书有着太多太多需要你学习的东西。
- 48、打通任督六脉之作--BWK老师篇
- 49、没看完，还行吧。
- 50、学完C语言就应该读这本书。
- 51、翻译扣星！
- 52、翻译真的很烂。接口interface竟然翻译成界面。。。
- 53、2000年读的老书了，Rob Pike的经典，2000年版封皮其实不是这样的。

精彩书评

- 1、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 流行的翻译方式（个人意见）界面 接口在线函数 内联函数建构函数 构造函数废料收集 垃圾收集两千年问题 千年虫问题神秘的数 魔数黑箱测试 黑盒测试书籍名称也是如此：《完整编程》(Code Complete) 代码大全《熟练的C程序设计》(Expert C Programming) C专家编程《神秘的人月》(The Mythical Man Month) 人月神话《程序设计精萃》(Programming Pearls) 编程珠玑
- 2、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，立马去翻编程珠玑，果然在记忆中有所讨论，自己思考了一下中文切词用该算法的可行性，也算是温故而知新，先前看编程珠玑还未有此想法。排错部分的各种方法很实际，以前一直觉得没什么书好好介绍这方面技巧，这回总算发现了知音。自己重度依赖调试器的使用，所以关于这是最后一个手段的论点不赞同，其它部分都是毫不犹豫的赞。测试部分也许当时还未有现在如此丰富吧。性能部分算是老调重弹，不过也算给了实例，最近听闻了两个真实案例，如果那两个项目pm能读过类似本书的内容也不会有让人惊讶的举措了。移植性，涉及好多常识，如字节高低位顺序等，好多摩登程序员未必明白吧。也许这就是高级语言的优势。最后关于语言是自己的兴趣所在，不过毕竟这不是编译的书，全都一笔带过了。书中示例代码充足易懂，认真研读，绝对可以增长，巩固一些知识。补充一下标题，感慨现在很多程序员的质量，希望你不是其中一员。
- 3、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。
- 4、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”。
- 5、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。
- 6、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。
- 7、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来看。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。
- 8、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，作者首先通过一个简单的实现，分析了这个实现的各个缺点，然后根据前面的分析，完成了一个可用的库函数。新手最大的问题就是不知道自己的问题在哪里，而作者的这种表述方式对新手的帮助无疑是莫大的。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com