

《高阶Perl》

图书基本信息

书名：《高阶Perl》

13位ISBN编号：9787111427735

10位ISBN编号：7111427734

出版时间：2013-7

出版社：机械工业出版社

作者：Mark Jason Dominus

页数：392

译者：滕家海

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.tushu111.com

前言

前言在编程圈子里有一句著名的俗语，一个优秀的Fortran程序员可以用任何语言写Fortran程序。然而，让人悲哀的是，不管他们是否愿意，Fortran程序员用任何语言写Fortran程序。类似地，作为Perl程序员，我们也在用Perl写C程序，不管我们是否愿意。这让人羞愧，因为Perl是一门比C更富有表现力的语言。我们本可以做得更好，以C程序员梦想不到的方式使用Perl，但是我们没有那样做。怎么会这样呢？Perl的设计初衷是一方面作为C的替代品，另一方面作为UNIX脚本语言（如Bourne Shell和awk）的替代品。Perl最初的主要拥护者是UNIX系统管理员，他们熟悉C和UNIX脚本语言，因此自然地倾向把Perl程序写成类似于C和awk的程序。Perl的发明人（Larry Wall）来自这个系统管理员社区，还有Randal Schwartz，与Perl合著了《Programming Perl》，本书是第一本也仍然是最重要的一本Perl参考著作。其他重要的早期贡献者还包括Tom Christiansen，也是古董级C与Unix专家。即使Perl程序员不是来自Unix系统管理员社区，他们也被其中的人或者其中的人训练的人训练成那样。大约在1993年我开始阅读关于Lisp的书，我有一个重要的发现：Perl更像Lisp而不是C。如果你拿起一本优秀的Lisp书，其中会有一段文字描述Lisp的优秀特点。例如，《Paradigms of Artificial Intelligence Programming》，作者Peter Norvig，有一节标题是“是什么让Lisp与众不同”，其中就描述了Lisp的七个特点。Perl分享了其中六个，C一个也没有分享。这些是巨大的、重要的特点，如头等函数、符号表的动态访问以及自动存储管理。Lisp程序员自1957年以来就在使用这些特点。他们知道如何高效使用这些语言特点的许多方法。如果Perl程序员能发现这些Lisp程序员已经知道的事情，那么他们将学到许多使Perl编程工作更轻松的方法。说比做容易。几乎没有人愿意听Lisp程序员的。Perl人对Lisp心存芥蒂，就像Larry Wall的著名评论所表明的，Lisp与混合了剪下的指甲碎屑的燕麦看起来一样。Lisp程序员经常制造像“cons”与“cooder”的滑稽噪声，他们还讨论像PC loser - ing问题的东西。他们相信Lisp比其他语言更好，他们也是这么说的，这让人不快。但是现在一切都好了，因为现在你不必听Lisp人的了。你可以听我的。我将制造一些轻缓的关于散列和存储以及glob的噪声，并讨论流行的和令人鼓舞的软引用与变量销毁问题。与其告诉你Lisp有多么好，不如告诉你Perl有多么好，到最后你将不会对Lisp有任何认识，但是对Perl知道得将会更多。然后你可以停止用Perl写C程序了。我认为你将发现这是一个很好的改变。Perl就是Perl的样子，比一个慢版本的C好得多。当你写Perl程序而不是C程序时，你将惊喜于所能够做到的。致谢每段致谢都以这样一句开头“要是没有我的编辑Tim Cox不知疲倦的支持与协助，这本书肯定还没写完”。除非你写一本书，不然你将不会意识到这是多么真实的。在这里我无法用语言表达，要是没有Tim不知疲倦的支持与协助，这本书就不会写完，感谢他的贡献、友善，还有他极大的耐心。这本书姗姗来迟，在我工作的时候Tim经历了三位助理。所有这些人都是乐于助人的和能干的，因此我谢谢Brenda Modliszewski、Stacie Pierce以及Richard Camp。“能干的”听起来可能无力，但是我认为这是最高的褒奖。多谢产品经理Troy Lilly与Simon Crump，不仅是能干的而且与他们合作愉快。临近成书之前，我开始为实例代码写测试。我厌恶地认识到几乎没有程序工作正常。有大量的小错误（有些也不小），代码与输出之间的不兼容、排版等。谢谢Robert Spier在最后关头的英勇，我认为他捕获了这些错误里的大部分。Robert不仅确实是能干的、乐于助人的、富有成效的，而且确实是活泼开朗的。如果本书中的任何代码程序运行如预期，那么要感谢Robert。（如果不是这样，你应该责备我，而不是Robert。）Robert也负责命名我用来准备手稿的MOD文件准备系统。我妻子Lorrie Kim的贡献太多太大以至于无法一一描述。本书题献给她。其他许多人对本书作出了贡献，但是他们中许多人在当时并不知道。我幸运地拥有许多卓越的老师，我有时不得不狠狠地考验他们的忍耐力。谢谢Mark Foster、Patrick X. Gallagher、Joan Livingston、Cal Lobel（第一位教我编程的老师）、Harry McLaughlin、David A. J. Meyer、Bruce Piper、Ronnie Rabassa、Michael Tempel以及Johan Tysk。正当我认为一切都错过了的时候，Mark Foster鬼使神差地为本书建议了书名。本书直接从两本更早的书获得灵感：《ML for the Working Programmer》，作者Lawrence Paulson，以及《Structure and Interpretation of Computer Programs》，作者Harold Abelson和Gerald Jay Sussman。其他有重要影响的是《Introduction to Functional Programming》，作者Richard Bird和Philip Wadler，以及《Paradigms of Artificial Intelligence Programming》，作者Peter Norvig。正式的技术阅校者的工作报酬比他们可能在别的项目中更低。这本书花了很长时间写就，尽管我想与阅校者就每件小事进行长时间交谈，但是我害怕如果那么做了，我将永远也不会完成。所以我很少与阅校者联系，也许他们认为我只是把他们的建议塞进了碎纸机。但我不是那样的，我极其认真地仔细研究他们所有的批评，并为这些批评中的大多数伤脑筋。我要谢谢

阅校者：Sean Burke、Damian Conway、Kevin Lenzo、Peter Norvig、Dan Schmidt、Kragen Sitaker、Michael Scott以及Adam Turoff。在写作的时候，我维护了对本书感兴趣的人的一个邮件列表，并把未定稿的章节发到邮件列表。这非常有帮助，我已经把这种做法推荐给其他人。我的邮件列表的650个令人惊奇的成员无法一一列在这里。他们都是提供了帮助与支持的，本书由于他们的投入而更加完美。一些突出的人贡献了大量的具体内容：Roland Young、Damien Warman、David “Novalis” Turner、Iain “Spoon” Truskett、Steve Tolkin、Ben Tilly、Rob Svirskas、Roses Longin Odounga、Luc St - Louis、Jeff Mitchell、Steffen Müller、Abhijit Menon - Sen、Walt Mankowski、Wolfgang Laun、Paul Kulchenko、Daniel Koo、Andy Lester、David Landgren、Robin Houston、Torsten Hofmann、Douglas Hunter、Francesc Guasch、Kenneth Graves、Jeff Goff、Michael Fischer、Simon Cozens、David Combs、Stas Bekman、Greg Bacon、Darius Bacon，以及Peter Allen。我向许多提供帮助的贡献者道歉，考虑到篇幅问题我没有放在上面的列表里，更要向我意外遗漏的几个特别有帮助的贡献者道歉。Wolfgang Laun和Per Westerlund特别勤勉地帮助我纠正第二次印刷的错误。在开始写作以前，我收到了关于挑选出版社的有价值的建议，它们来自Philip Greenspun、Brian Kernighan和Adam Turoff。Damian Conway和Abigail对我的提议给予了有帮助的建议与批评。Sean Burke录下了我的象牙塔谈话，刻成CD并发给我，还在最后一刻提供了有关RTF的咨询。他定期发邮件提醒我，书到什么流程了，还经常在我无法确定的时候到访。第4章里几个特殊的想法是由其他人建议的。Meng Wong建议了聪明与恰当的“里程表”比喻。Randal Schwartz帮我“增补”函数。Eric Roode建议了多列表迭代器。当我需要读Paul Graham绝版的书时，A . E . Sundstrom借给了我。当我需要《The Art of Computer Programming》第二卷的时候，Hildo Biersma与Morgan Stanley为我买来了。当我需要钱的时候，B . B . King就借给我一些。谢谢他们所有人。第9章的约束系统绘制程序是一个大项目，我很长时间盯着该项目。要是没有Wm Leler及时的帮助，我可能还盯着呢。Tom Christiansen、Jon Orwant以及Nat Torkington在使我融入Perl社团方面扮演了主要的与不可替代的角色。最后，“没有这个，这本书不可能写成”语句不能结束，要是不感谢Larry Wall写了Perl并创立了Perl社团，没有这个，这本书不可能写成。

内容概要

《高阶Perl》是Perl全球社区公认的经典著作，是资深Perl技术专家（核心贡献者之一）数十年工作经验的结晶，Perl6设计者之一Conway亲自作序推荐。《高阶Perl》处处皆珍宝，不仅对Perl领域的各种经典问题给出了独到且精辟的解释，而且深入探讨了Perl中各种最新的主题，如递归、迭代器、过滤器、记忆术、划分、数值方法、高阶函数、currying、切断排序、基于语法的解析、惰性求值和约束编程等内容，并将这些转换成现实编程工作中强有力的实用工具：文件系统互动、HTML处理、数据库访问、网页抓取、排版、邮件处理、家庭理财、文本描图和图表生成等。全书包含大量真知灼见和最佳实践。

<http://hop.perl.plover.com/hopcn/>

作者简介

Mark Jason Dominus (陶敏修)，资深Perl技术实践者和研究者，对Perl技术有非常深刻的认识和理解，拥有丰富的实践经验。他是Tie::File、Text::Template和Memoize模块以及perlreftut手册页的作者，还是Perl核心贡献者之一。2001年赢得Larry Wall实用性大奖。热衷于分享，撰写了大量关于Perl的技术文章并分享在自己的博客上，由于文章技术含量高，所以非常受欢迎，这使得他在Perl技术圈内颇具知名度和影响力。他喜欢演讲，经常在大型公司和团体的Perl研讨会和培训课程上发表演讲或提供培训服务。

书籍目录

本书赞誉

译者序

前言

第1章 递归与回调1

1.1 十进制到二进制的转换1

1.2 阶乘2

1.2.1 为什么私有变量是重要的3

1.3 汉诺塔4

1.4 层次化数据8

遍历的应用和变化11

1.6 函数式编程与面向对象式编程17

1.7 HTML17

1.7.1 更灵活的选择21

1.8 当递归膨胀时22

1.8.1 Fibonacci数22

1.8.2 划分24

第2章 分配表27

2.1 配置文件处理27

2.1.1 表驱动配置28

2.1.2 分配表的优势29

2.1.3 分配表策略32

2.1.4 默认行为34

2.2 计算器35

2.2.1 再访HTML处理38

第3章 缓存与记忆术41

3.1 缓存修正递归42

3.2 内联缓存43

3.2.1 静态变量44

3.3 好主意44

3.4 记忆术45

3.5 MEMOIZE模块45

3.5.1 作用域和有效期47

3.5.2 词法闭包49

3.5.3 再谈记忆术52

3.6 CAVEATS52

3.6.1 返回值不依赖参数的函数52

3.6.2 有边界效应的函数53

3.6.3 返回引用的函数53

3.6.4 带记忆的时钟54

3.6.5 非常快的函数54

3.7 键的生成55

3.7.1 用户提供的键生成器的更多应用58

3.7.2 内联的参数归一化的缓存管理59

3.7.3 带有引用参数的函数61

3.7.4 划分61

3.7.5 为非纯函数定制的键生成62

3.8 对象方法里的缓存62

- 3.8.1 对象方法的记忆术64
- 3.9 持续的缓存65
- 3.10 可供选择的记忆术66
- 3.11 传播福音71
- 3.12 速度的好处71
 - 3.12.1 剖析和性能分析72
 - 3.12.2 自动剖析73
 - 3.12.3 钩子74
- 第4章 迭代器75
 - 4.1 简介75
 - 4.1.1 文件句柄是迭代器75
 - 4.1.2 迭代器是对象76
 - 4.1.3 迭代器的其他普通实例77
 - 4.2 自制迭代器78
 - 4.2.1 一个平凡的迭代器：upto()78
 - 4.2.2 dir_walk()80
 - 4.2.3 聪明的灵感81
 - 4.3 实例82
 - 4.3.1 排列83
 - 4.3.2 基因组序列生成器88
 - 4.3.3 文件句柄迭代器91
 - 4.3.4 一个纯文本文件数据库91
 - 4.3.5 反向搜索数据库97
 - 4.3.6 随机数生成100
 - 4.4 过滤和变换103
 - 4.4.1 imap()103
 - 4.4.2 igrep()105
 - 4.4.3 list_iterator()106
 - 4.4.4 append()106
 - 4.5 半谓词问题107
 - 4.5.1 避免问题108
 - 4.5.2 可选的undef109
 - 4.5.3 重写功能函数111
 - 4.5.4 返回多个值的迭代器112
 - 4.5.5 明确的耗尽函数112
 - 4.5.6 四操作数迭代器114
 - 4.5.7 迭代器方法116
 - 4.6 可选的迭代器界面116
 - 4.6.1 用foreach循环多个数组117
 - 4.6.2 带有类each界面的迭代器120
 - 4.6.3 系住变量界面121
 - 4.7 一个扩展的例子：网页抓取124
 - 4.7.1 仅追逐有趣的链接126
 - 4.7.2 参考URL127
 - 4.7.3 robots.txt130
 - 4.7.4 总结132
- 第5章 从递归到迭代器134
 - 5.1 再访划分问题134
 - 5.1.1 搜寻所有可能的划分136

- 5.1.2 优化138
- 5.1.3 变化139
- 5.2 如何把一个递归函数转换成一个迭代器142
- 5.3 一个通用的搜索迭代器149
- 5.4 其他通用的移除递归的技术152
 - 5.4.1 尾调用移除152
 - 5.4.2 产生尾调用159
 - 5.4.3 明确的栈161
- 第6章 无限流170
 - 6.1 链表170
 - 6.2 惰性链表171
 - 6.2.1 一个平凡的流：upto()172
 - 6.2.2 流的功能函数173
 - 6.3 递归流175
 - 6.3.1 使流带记忆176
 - 6.4 HAMMING 问题179
 - 6.5 正则字符串生成181
 - 6.5.1 按次序生成字符串189
 - 6.5.2 正则匹配191
 - 6.5.3 切断排序192
 - 6.6 NEWTON-RAPHSON方法200
 - 6.6.1 近似流203
 - 6.6.2 导数203
 - 6.6.3 乌龟和兔子205
 - 6.6.4 金融207
 - 6.7 幂级数209
 - 6.7.1 导数213
 - 6.7.2 其他函数213
 - 6.7.3 符号计算213
- 第7章 高阶函数与currying217
 - 7.1 currying217
 - 7.2 普通的高阶函数222
 - 7.2.1 自动的currying223
 - 7.2.2 原型225
 - 7.2.3 更多的currying227
 - 7.2.4 还是更多的currying228
 - 7.3 reduce()和combine()229
 - 7.3.1 布尔操作符232
 - 7.4 数据库234
 - 7.4.1 操作符重载238
- 第8章 解析239
 - 8.1 词法分析器239
 - 8.1.1 评估《》操作符240
 - 8.1.2 更通用的词法分析器243
 - 8.1.3 链式词法分析器245
 - 8.1.4 偷看249
 - 8.2 一般的解析250
 - 8.2.1 语法250
 - 8.2.2 解析语法254

- 8.3 递归下降解析器256
 - 8.3.1 非常简单的解析器257
 - 8.3.2 解析器操作符258
 - 8.3.3 合成的操作符259
- 8.4 算术表达式261
 - 8.4.1 一个计算器267
 - 8.4.2 左递归267
 - 8.4.3 star()的一个变体272
 - 8.4.4 通用操作符解析器275
 - 8.4.5 除错277
 - 8.4.6 完成的计算器283
 - 8.4.7 错误诊断和矫正285
 - 8.4.8 大数290
- 8.5 解析正则290
- 8.6 大纲294
- 8.7 数据库查询解析299
 - 8.7.1 词法分析器299
 - 8.7.2 解析器301
- 8.8 回溯解析器304
 - 8.8.1 续篇305
 - 8.8.2 解析流308
- 8.9 重载310
- 第9章 声明式编程315
 - 9.1 约束系统315
 - 9.2 本地传播网络315
 - 9.2.1 实现一个本地传播网络317
 - 9.2.2 本地传播的问题325
 - 9.3 线性方程326
 - 9.4 linogram : 一个绘图系统327
 - 9.4.1 方程334
 - 9.4.2 值343
 - 9.4.3 特征类型354
 - 9.4.4 解析器360
 - 9.4.5 缺失的特征373
 - 9.5 总结375

编辑推荐

《高阶Perl》编辑推荐：Perl全球社区公认经典著作，资深Perl专家（核心贡献者之一）数十年工作经验结晶，Perl6设计者之一Conway亲自作序推荐！深入探讨Perl领域的各种经典问题和新主题，包含大量最佳实践和真知灼见，Perl程序员进阶必读。

精彩短评

- 1、翻译实在是有些烂
- 2、内容不错，翻译的准确度还不错，整体读来较涩。可参阅英文原文。
- 3、虽然中文版的质量并不理想，但这依然是一本好书。部分章节可以看作是Perl版的SICP。
- 4、很帅，不过中文翻译真是烂到家
- 5、不得不说，读起来比英文版轻松点。代码比一般的Perl的书难。
- 6、perl的函数式编程？
- 7、刚到手，看了前几页，忍不住来这里说一下。第一章，第一页: `sub binary { binary my ($n) = @_ #`这里多了一个binary吧？第三页 为什么私有变量是重要的虽然去掉 \$n 是全局变量，但是每次调用函数的时候会正确的初始化，所以我并不认为和书中说的一样。我认为可以得到正确的结果。只是多了一个全局变量\$n.
- 8、不想把Perl当C用的人应该读读。
- 9、本以为它会讲解递归到迭代的自动转换呢，结果一看，原来只不过是手工转换的！有几处技术上的难点，但总的来说，技术含量不高，讲解的是2005年的Perl 5，有些过时了相对来说，书的价钱就有些不值了！
- 10、这本开头就讲算法，，第一章看了好几天才明白，，，

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.tushu111.com