

Perl Testing

程序高手秘笈

Ian Langworth &
chromatic 著

盛春 蒋永清 译

- Testing strategies
- Testing whole applications
- Testing Perl & non-Perl code
- Helpful testing libraries

O'REILLY®

東南大學出版社

Perl Testing

by
David
M.
Dries

Test::More
Test::Exception
Test::Fatal

Test::More
Test::Exception
Test::Fatal

OSCAR
PUBLISHING

Perl Testing

程序高手秘笈

Ian Langworth & chromatic 著

盛春 蒋永清 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目 (CIP) 数据

Perl Testing 程序高手秘笈 / (美) 朗沃茨 (Langworth, I.),
(美) 克劳麦迪克 (chromatic) 著; 盛春, 蒋永清译.
南京: 东南大学出版社, 2008.7

书名原文: Perl Testing: A Developer's Notebook

ISBN 978-7-5641-1008-6

I . P… II . ①朗… ②克… ③盛… ④蒋… III . PERL 语
言—程序设计 IV . TP312

中国版本图书馆 CIP 数据核字 (2007) 第 193754 号

江苏省版权局著作权合同登记

图字: 10-2007-099 号

©2005 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2008. Authorized translation of the English edition, 2005 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2005。

简体中文版由东南大学出版社出版 2008。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

Perl Testing 程序高手秘笈

出版发行: 东南大学出版社

地 址: 南京四牌楼 2 号 邮编: 210096

出 版 人: 江 汉

网 址: <http://press.seu.edu.cn>

电子邮件: press@seu.edu.cn

印 刷: 扬中市印刷有限公司

开 本: 787 毫米 × 980 毫米 16 开本

印 张: 12.75 印张

字 数: 214 千字

版 次: 2008 年 7 月第 1 版

印 次: 2008 年 7 月第 1 次印刷

书 号: ISBN 978-7-5641-1008-6/TP · 165

印 数: 1~4000 册

定 价: 36.00 元 (册)

本社图书若有印装质量问题, 请直接与读者服务部联系。电话 (传真): 025-83792328

O'Reilly Media, Inc. 介绍

为了满足读者对网络和软件技术知识的迫切需求,世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权东南大学出版社, 翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司, 同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》(被纽约公共图书馆评为二十世纪最重要的 50 本书之一) 到 GNN (最早的 Internet 门户和商业网站), 再到 WebSite (第一个桌面 PC 的 Web 服务器软件), O'Reilly Media, Inc. 一直处于 Internet 发展的最前沿。

许多书店的反馈表明, O'Reilly Media, Inc. 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比, O'Reilly Media, Inc. 具有深厚的计算机专业背景, 这使得 O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员, 或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群——他们本身是相关领域的技术专家、咨询专家, 而现在编写著作, O'Reilly Media, Inc. 依靠他们及时地推出图书。因为 O'Reilly Media, Inc. 紧密地与计算机业界联系着, 所以 O'Reilly Media, Inc. 知道市场上真正需要什么图书。

译者的话

从前有个美好的童话国度 ...

在软件开发技术领域有数不清的语言存在,如果给世界上每个国家分配一种语言,只怕还有很多著名语言没有着落。Perl 就是其中一种默默奉献的语言,在计算机使用的各个领域都能发现它的踪影。作为证据可以打开 CPAN 页面搜索感兴趣的问题,几乎所有的领域都有人在尝试用它来解决问题。

在互联网普及的同时,这个语言承载着 CGI 技术而逐渐被人所知,开始进入中国。但是它的能力还远不止制作动态页面,其中更是蕴含着计算机文化的根基。它是一种在设计的时候就充分鼓励人性当中美好品性的语言,因此非常特殊地吸引着一群普通用户深入学习。比起其他的语言来说,Perl 的社区也是非常成熟的。特别是其中志愿者互相帮助的风尚如此盛行,实在让人惊叹。

作为一种如此特殊的语言,它的学习渠道也是充满特色的。很多人用它来做系统管理,写脚本帮助提高操作自动化程度。很多人用它的胶水功能来连接自己喜欢的小工具,实现更加强大的功能。也有很多人用它来做快速开发,充满创意地使用各种模块实现各种客户定制的应用。这些人都大声感谢 Larry Wall 的发明,并声明自己的选择是幸运的。

但是 ...

有一天,管理员的脚本突然罢工了,原因是某个人(也许就是你自己)不经意

间改变了系统的配置。开发人员的程序在演示的时候崩溃了，并且怎么样也很难解释原因，只能归因于外星生物电磁干扰。而那些胶水程序的某个环节也脱胶了，吸引人的小把戏演砸了。IT 领域的东西就是这么充满动态，虽然面对的只是静静闪动的光标。

也许这些问题背后的共同点就在于问题的简化过了头，从某个时间开始简洁变成了整洁、精密变成了秘密、快乐变成了傻乐。而我们都习惯于这个转变的发生，好像一个干净的台布总是会变得肮脏一样。

难道没有办法能阻止这个不可抗拒的进程？为什么每个项目开头都是那么美好，最后都变成行业典型的失败？会不会有个白马王子出现，振臂一呼改变格局？如果这样，那么我们又可以重新埋头写自己的小脚本、小模块、小程序了，也许还能贡献点什么到 CPAN 上面去呢？

就让我们开始测试吧，告别那些臃肿的名词：UML、SOA、网格计算，回到问题的根本。为什么我要定义那个变量，它真的被合理使用了么？让我们的测试自动化，因为人人都是有可能犯错误的，包括测试部门的经理。在我们觉得不可能测试的地方，真的没有可以努力的余地了么？

从此王子和公主 ...

在这本书的翻译之后，我个人开始明白了模块的编写，不再迷惑为什么 CPAN 上面中国人的贡献是如此之少了。我开始爱上了 Plain Old Document，并用它来书写各种文档、简历。我开始如此高效的工作，用 Perl 来教导数据库课程。这是边翻译边学习的回报，而我也希望读者朋友们能够分享我的快乐，因为 Perl 就是如此的有趣，而测试也是如此的有必要。

回想 2004 年我们团队第一次尝试翻译 Perl.com 的文章，那真是个很好的开始。由于亚洲 debian 用户组的聚会，我们得到了台湾 Perl 高手的指点。现在翻阅那时候在小本子上面记录的那些模块，发现原来很多都在此书中被提到，都可以用来测试或被测试验证。翻译本书是一个如此幸运的选择，定然是超出了个人的努力而得到的回报，这也可能就是所谓的被 bless 了吧。

在本书翻译过程中，我们两人之间完美地进行了合作，彼此了解了对方所熟悉的技术领域。在审阅过程中，得到了上海本地 Perl 爱好者王晖、cnhacktnt、

王亮、netcasper 的参与与评审，以及孙梦然提供的宝贵建议，在此深表感谢。我们也非常感谢东南大学出版社的张烨编辑，直接给了我们一个翻译 O'Reilly 技术书籍的机会，没有这个项目我们 2007 年的自我评估将会失色三分。

最后感谢我们的家人——父亲蒋记还、母亲张群英、妻子沈圆，父亲盛林伯、母亲高霞飞、妻子王晓冬，没有你们的鼓励与支持，我们将无法从容而自信地投入工作。感谢你们！

盛春 蒋永清
2007 年 11 月

关于作者

Ian Langworth (<http://langworth.com/>) 写了很多年的Perl程序, 并从2003年起就一直活跃在Perl社区中。他为CPAN贡献了许多模块, 大多数和Kwiki相关。他还在Perl相关的一些大会如LISA和YAPC上作过演讲。Ian同时还是广为流传的小工具Cadubi的作者, 这个软件如今被很多自由操作系统打包收纳。

Ian目前在Northeastern大学研究计算机科学和认知心理学。在进修硕士学位期间, 他参与了志愿者系统管理员团队的工作, 并为帮助他人提高代码质量和健壮度而不懈努力着。

他现在居住在马萨诸塞州的波士顿市Fenway公园附近, 并积极参与当地Boston Perl Monger小组的活动。

chromatic是O'Reilly Network的技术编辑, 主要关注开源、Linux、开发以及Perl等方面的主题。他是《Extreme Programming Pocket Guide》和《Running Weblogs with Slash》这两本书的作者, 他还是《BSD Hacks》和《Gaming Hacks》的编辑。chromatic是Test::Builder模块的原作者, 此模块是所有新版Perl 5的测试模块的根基。他还为Perl核心代码写了许多测试。他曾在多个Perl大会(如OSCON)举办过讲座和演示。在Perl.com上面也有很多他写的关于测试方面的文章。他和他的两只猫住在俄勒冈的波特兰市西面, 他家后院有条小河, 并且, 你猜得没错, 他手头确实还有些尚未结束的项目在做。

封面介绍

O'Reilly特有的封面设计风格主要来自读者的建议、我们自己的试验以及书店的经验。唯有独树一帜的封面才能凸显我们对于技术主题的阐述风格, 避免读者被枯燥的主题压得喘不过气来。

《程序高手秘笈》系列是以模仿传统实验室笔记的风格开创的, 实验室笔记对于研究员与他们的后继者而言是无价的工具。

实验室笔记的目的是要通过记录在工作中所获得的数据与结论来建立可信与真实的历史。笔记本的最前面是包括主人名字与研究主题的标题页, 笔记本中的每一页都应该标示页数并于前面加上条目。全文应该要清晰、易读并正

确标示日期；研究员应该要使用单纯、直接的语言来指出实验的名称与采取的步骤。计算过程应该被很小心地记录下来，并加上思路与想法的记录。每个写在笔记本中的实验都有引言与结论，它的目标是要做出可以作为参考文献的能被理解、有条理地组织的笔记。谨慎的文档制作能够建立起非常有价值的记录并给未来开发者提供最实用的指引。

目录

《程序高手秘笈》系列	1
------------------	---

前言	5
----------	---

第一章 测试起步	11
----------------	----

安装 Test 模块	11
运行测试	15
解析测试结果	17
编写你的第一个测试	20
装载模块	22
改进测试比对	26

第二章 编写测试	31
----------------	----

跳过测试	31
标注测试项目为 TODO	35
简单数据结构比较	38
数据组成	43
测试警告信息	46
测试异常	49

第三章 管理测试	51
测试的组织	51
检查覆盖率	55
编写测试库	60
对测试库的测试	63
编写测试用具	66
通过网络运行测试	68
自动运行测试	69
 第四章 与代码一起发布测试	 73
测试文档覆盖率	75
测试整个模块	80
让用户决定	81
让用户决定（续）	84
模块测试打包	85
收集测试报告	88
确保技量	91
 第五章 应对难以测试的代码	 95
重载内建操作符和函数	95
冒名模块	100
冒名对象	104
部分实现的冒名对象	110
重载运行时的代码	114
全局重载函数和操作符	119
 第六章 测试数据库	 123
置换测试数据库	123
使用临时数据库	131
冒名数据库	135

第七章 测试网站	141
测试后端逻辑	141
测试前端界面	146
录制和回放浏览过程	149
测试 HTML 的有效性	152
运行自己的 Apache 服务器	154
用 Apache-Test 进行测试	157
发布带有 Apache-Test 的模块	162
 第八章 使用 Test::Class 进行单元测试	165
编写测试用例	165
创建测试夹具	169
继承测试	172
用 Test::Class 跳过测试	175
用 Test::Class 标注 TODO 测试	176
 第九章 其他类型的测试	179
编写可测试的程序	179
发生了什么事?	182
测试程序	183
测试交互式程序	185
测试共享库	188

《程序高手秘笈》系列

噢，你挑中了这本书。酷。我真感到兴奋！当然啦，你一定会怀疑为何这些书有着不一样的外观，封面做得好像是在学校用的笔记本（所以，本系列也称为笔记本系列）。我的意思是说，这是奥莱理的书对吧？那动物到哪里去了？还有，说真的，有必要弄出另外一个系列吗？这不能就归作 cookbook 吗？还是说 nutshell，或者是那一种五花八门又很酷的 hacks？事实上，我们的构想是要给出包括了封面、字体、甚至是页边的笔记这样全新的外观与感受。这都是为了要给你一些在手边真的能用得着的东西。

我深深地认为20世纪90年代被描述成每个人想学会每件事（有何不可？反正我们都有来自网络公司的六位数收入），而千禧年则是“信息痛苦”（information pain），人们没有时间（或者金钱）去读完六百页的书，定期学习两百件事务，但其中只有四件可以用到目前的工作上。如果有办法坐在程序高手的旁边看着他做事，问问这些专家们为什么要选择某个特定方法，那个绝招是怎么办到的，或者说他在处理管道流的时候是怎样避免线程的问题，那不是比较好吗？这样的特殊想法是没有书本可以帮上忙的。书本可以指导你，让你决定，但一个程序设计师的思路最终还是无法真的就用只字片纸来抓住。

我们要用这个系列来证明这个想法的假设是错的。

程序高手秘笈如同它所宣称的：书中的潦草手稿与注记正是王牌技客（alpha geek）在玩新语言、API 或做项目时的心血结晶。那是解决问题直指要害的程序代码，删掉了那些更像是充页数的注释。有意思的是，这里并没有把焦点放在怎样才是漂亮的或者是在你有闲功夫来做的时候可能还挺有趣的事情上（上

一次是什么时候了？)，而是着眼于你就是得“交卷”的需求。各位，这可不是在上课，这是实战。如果你想看的是一大堆概念、体系结构与UML图表，本人在此郑重为您推荐我们的动物封面与nutshell系列丛书；如果你要的是太阳底下所有新鲜事的答案，我们的cookbook系列是个中极品；再如果你觉得身陷云里雾里而技术曲折又坎坷，hacks系列就是你的救星。但如果你是个程序设计师，你写程序，并且你就是想搞定它，那么你需要的就是程序高手秘笈。书中的字字句句，是一个开发者想要传达到你心中的想法，而这些只能被厘清到勉强够格出书。我们写的很享受……也希望你能喜欢它。

笔记本系列是……

范例导向的指引

如同你于稍后的“本系列书的编排”一节中会看到的，开发者的笔记本系列全书都是以范例程序代码为中心。你会看到几乎每一页都有程序代码，并且程序代码都是有作用的，而不是那种无聊到根本不值得用纸张去印的“Hello World!”程序。

以开发者为目标

你曾经读过那些充斥着时髦术语，感觉上不像是程序设计，反倒比较像广告宣传，似乎是专门给呆伯特式的老板看的书吗？我们也读过——本系列丛书正是那些书的反方。事实上，一个好的笔记本对不会写程序的人来说是本无字天书（别说我们没有警告你！），并且本来就应该是这样。但是对于开发者……真是超级棒。

真正值得体会玩味的

你真的有时间坐下来阅读完全无趣的东西吗？如果有的话，可能你正在看的是大部分的语言参考手册。但是如果你跟我们其他人一样，还是看笔记本比较好。实用的程序代码范例，针对应用范例的精要讨论，甚至还掺杂些幽默成分，这才是组成优良笔记本的好材料。

起而行，不坐而谈

如果你在夜深人静的时候想远离计算机好好看本书，这些书帮不了什么忙。这是打算给你在写程序跟bytecode奋战的时候用的。因此，笔记本讲的除了程序还是程序。使用前请先启动好你的编辑程序。

【译注】原文并未直接提到呆伯特这个人物，但pointy-haired bosses也无其他人选能出其右。

笔记本系列并不是……

讲义

我们可不随便让任何人都来出一本程序高手秘笈。作者得是货真价实的程序设计师，最好是写起程序来废寝忘食的那一种。全职作家、学术研究与理论大师在某些方面是一流的，但本系列是关于真枪实弹的程序设计，里面所谈的都是指令，而不是一些破碎的理论概念。

充斥着概念图形与类继承

这可不是 nutshell 系列（老早就跟你说过了）。你不会找到一百页的索引条目列出所有的 method，你也别想看到整页都是 UML 图表与 method、继承树与流程图，你能看到的都是一页又一页的程序源代码。你有没有开始认识这个循环的主题？

一大堆的解释，一点点的应用程序

目前，许多的程序设计书在给你看到有用的程序代码之前似乎都有三四章以上的章节。我不确定到底是哪个家伙被作者说服，以为可以让读者等这么久，但铁定不是这系列里面的人。我们认为如果你在十页以内没有写程序的话，一定是出了什么问题。本系列丛书也满载了实用的应用程序，以最快的速度将你从书中范例带到实际工作中。

本系列书的编排

开发者的笔记本系列试着要传达与绝大多数其他书籍不同的信息，因为这样，组织编排也不相同。书中是有章节的安排，但是其间的差异也如同程序高手秘笈与传统程序设计书所相去之远是一样的。首先，你会发现每个章与节中的标题都被安排围绕着一个特定任务。要注意我们说的是“任务”而不是“概念”。这可是攸关于本系列的重点，首要工作就是去完成某件事。每个标题都代表着一个实作 (lab)。实作的意思如同它的名字所说的那样，是指完成特定目标的步骤。事实上，在每个实作之下，你会看到的第一个标题就是：“我该怎么做？”这是每个实作的核心问题，你也会在章节中看到够低级的 (down-and-dirty) 程序代码与细节。

有些实作中的事务是不要你去做（你在高中的化学实验课曾经把玩过钾吗？），只是为了帮你避免落入同样的陷阱。有些实作告诉你需要在一开始就得注意某些议题的充分理由，因此我们称此段落为“我为何要注意？”的原因

显而易见。你会在“发生什么事？”段落中找到一些程序范例没能清楚沟通的事物。在这些段落中你会发现概念与理论也是牢牢地盯住手上正在解决的任务，而不是一些用来增加页数的解释文字。最后，有些实作提出了替代方案以及采用不同手法对付类似问题的疑难解答。这些都属于“关于……”段落，能够帮助交代清楚每个任务于程序设计大方向下的一些来龙去脉。

还有最后一件事情——你会在页边发现草写的眉批。那不是装饰用的，而是来自于产品开发者的提示、绝招与顿悟，有时也可能是个帮你解闷的冷笑话。这些注记代表着整体对话过程的一部分，尽可能让你贴近开发者的想法。希望这可以让你明显地感受到你确实是在向大师讨教。

——Brett McLaughlin, 丛书设计者

【译注】本系列原文经常使用俚语与市井典故来凸显此系列之与众不同，故译文也以此为乐，读者应不至于误解为译者水平低下。