

软件调试修炼之道



[软件调试修炼之道_下载链接1](#)

著者:Paul Butcher

出版者:人民邮电出版社

出版时间:2011-6

装帧:平装

isbn:9787115252647

本书主要讲述如何运用方法和调试工具在客户投诉之前自动检测程序中的bug，紧紧围绕问题重现、问题诊断、缺陷修复、反思四个中心环节，并将调试置于软件开发与运行的大环境中，为我们道出了软件调试修炼之道。

本书适用软件开发、调试一线人员及一切热爱软件调试之道的有志者。

作者介绍:

Paul Butcher

资深程序员，涉猎广泛，从单片机编码到高级声明式编程无所不精。Paul是一位少年天才，8岁时就已经开始在8位机上编写游戏。最近几年他开始痴迷于赛车，认为自己是可以和汉密尔顿比肩的赛车手。

目录: 第一部分 问题的核心

第1章 山重水复疑无路 2

1.1 调试不仅是排除缺陷 2

1.2 实证方法 4

1.3 核心调试过程 5

1.4 先澄清几个问题 6

1.4.1 你知道要找的是啥吗 6

1.4.2 一次一个问题 7

1.4.3 先检查简单的事情 7

1.5 付诸行动 8

第2章 重现问题 9

2.1 重现第一，提问第二 9

2.1.1 明确开始要做的事 10

2.1.2 抓住重点 10

2.2 控制软件 11

2.3 控制环境 11

2.4 控制输入 13

2.4.1 推测可能的输入 13

2.4.2 记录输入值 15

2.4.3 负载和压力 19

2.5 改进问题重现 20

2.5.1 最小化反馈周期 20

2.5.2 将不确定的缺陷变为确定的 22

2.5.3 自动化 25

2.5.4 迭代 26

2.6 如果真的不能重现问题该怎么办 27

2.6.1 缺陷真的存在吗 27

2.6.2 在相同的区域解决不同的问题 27

2.6.3 让其他人参与其中 27

2.6.4 充分利用用户群体 28

2.6.5 推测法 28

2.7 付诸行动 29

第3章 诊断 30

3.1 不要急于动手——试试科学的方法 30

3.2 相关策略 35

3.2.1 插桩 36

3.2.2 分而治之 37

3.2.3 利用源代码控制工具 38

3.2.4 聚焦差异 39

3.2.5 向他人学习 39

3.2.6 奥卡姆的剃刀 40

3.3 调试器 40

3.4 陷阱 41

3.4.1 你做的修改是正确的吗	41
3.4.2 验证假设	42
3.4.3 多重原因	43
3.4.4 流沙	44
3.5 思维游戏	45
3.5.1 旁观调试法	45
3.5.2 角色扮演	46
3.5.3 换换脑筋	47
3.5.4 做些改变，什么改变都行	47
3.5.5 福尔摩斯原则	48
3.5.6 坚持	49
3.6 验证诊断	49
3.7 付诸行动	50
第4章 修复缺陷	51
4.1 清除障碍	51
4.2 测试	52
4.3 修复问题产生的原因，而非修复现	54
4.4 重构	56
4.5 签入	57
4.6 审查代码	58
4.7 付诸行动	59
第5章 反思	60
5.1 这到底是怎么搞的	60
5.2 哪里出了问题	61
5.2.1 我们已经做到了吗	62
5.2.2 根本原因分析	62
5.3 它不会再发生了	63
5.3.1 自动验证	63
5.3.2 重构	64
5.3.3 过程	65
5.4 关闭循环	65
5.5 付诸行动	66
第二部分 从大局看调试	
第6章 发现代码存在问题	68
6.1 追踪缺陷	68
6.1.1 缺陷追踪系统	68
6.1.2 怎样才能写出一份出色的缺陷报告	69
6.1.3 环境和配置报告	70
6.2 与用户合作	72
6.2.1 简化流程	72
6.2.2 有效的沟通	73
6.3 与支持人员协同工作	77
6.4 付诸行动	78
第7章 务实的零容忍策略	79
7.1 缺陷优先	79
7.1.1 早期缺陷修复可以大大降低软件运行的不确定性	79
7.1.2 没有破窗户	80
7.2 调试的思维模式	81
7.3 自己来解决质量问题	83
7.3.1 这里没有“灵丹妙药”	83
7.3.2 停止开发那些有缺陷的程序	84
7.3.3 从“不干净”的代码中将“干净”的代码分离出来	84
7.3.4 错误分类	85
7.3.5 缺陷闪电战	86

- 7.3.6 专项小组 87
- 7.4 付诸行动 87
- 第三部分 深入调试技术
- 第8章 特殊案例 90
 - 8.1 修补已经发布的软件 90
 - 8.2 向后兼容 91
 - 8.2.1 确定你的代码有问题 92
 - 8.2.2 解决兼容性问题 93
 - 8.3 并发 95
 - 8.3.1 简单与控制 95
 - 8.3.2 修复并发缺陷 96
 - 8.4 海森堡缺陷 97
 - 8.5 性能缺陷 98
 - 8.5.1 寻找瓶颈. 99
 - 8.5.2 准确的性能分析 99
 - 8.6 嵌入式软件 100
 - 8.6.1 嵌入式调试工具 100
 - 8.6.2 提取信息的痛苦路程 102
 - 8.7 第三方软件的缺陷 102
 - 8.7.1 不要太快去指责 103
 - 8.7.2 处理第三方代码的缺陷 103
 - 8.7.3 开源代码 104
 - 8.8 付诸行动 106
- 第9章 理想的调试环境 107
 - 9.1 自动化测试 107
 - 9.1.1 有效的自动化测试 107
 - 9.1.2 自动化测试可以作为调试的辅助 108
 - 9.1.3 模拟测试、桩测试以及其他的代替测试技术 109
 - 9.2 源程序控制 110
 - 9.2.1 稳定性 110
 - 9.2.2 可维护性 111
 - 9.2.3 与分支相关的问题 111
 - 9.2.4 控制分支 112
 - 9.3 自动构建 113
 - 9.3.1 一键构建 114
 - 9.3.2 构建机器 115
 - 9.3.3 持续集成 115
 - 9.3.4 创建版本 116
 - 9.3.5 静态分析 117
 - 9.3.6 使用静态分析 119
 - 9.4 付诸行动 120
- 第10章 让软件学会自己寻找缺陷 121
 - 10.1 假设和断言 121
 - 10.1.1 一个例子 122
 - 10.1.2 等一下——刚才发生了什么 124
 - 10.1.3 例子，第二幕 124
 - 10.1.4 契约，先决条件，后置条件和不变量 125
 - 10.1.5 开启或关闭断言 125
 - 10.1.6 防错性程序设计 126
 - 10.1.7 断言滥用 128
 - 10.2 调试版本 129
 - 10.2.1 编译器选项 130
 - 10.2.2 调试子系统 130
 - 10.2.3 内置控制 132

- 10.3 资源泄漏和异常处理 133
 - 10.3.1 在测试中自动抛出异常 133
 - 10.3.2 一个例子 134
 - 10.3.3 测试框架 136
- 10.4 付诸行动 139
- 第11 章 反模式 140
 - 11.1 夸大优先级 140
 - 11.2 超级巨星 141
 - 11.3 维护团队 142
 - 11.4 救火模式 144
 - 11.5 重写 145
 - 11.6 没有代码所有权 146
 - 11.7 魔法 146
 - 11.8 付诸行动 147
- 附录A 资源 148
- 附录B 参考书目 157
- [\(收起\)](#)

[软件调试修炼之道_下载链接1](#)

标签

- 调试
- 软件调试
- 软件开发
- 计算机
- 编程
- 程序设计
- Programming
- 修炼之道

评论

内容不仅只有调试（事实上调试只有不多的[相对]一部分..）还涵盖了软件工程的诸多方面，但是面广不深入。不是说这书不好（不过中文翻译也是在渣，好多词句都要反复看几遍），只是没有达到我的预期要求。

不知是翻译的不好，还是咋地，读得吃力...为读完而读完...

（1）一次只做一个修改 （2）记录改变 （3）二分法

其实程序员有很多的时间都用来调试，看两本这方面的书，应该有所帮助

事实上这不是一本工具书，它不教你用具体的某种方法/工具进行调试，相反，传达的是一种调试的思路和理念。个人感觉读起来不是非常顺畅，可能跟我没有学过软件工程有关系。

对薄书一向有好感。干货不少，发现要学的东西很多～

一本不错的书，但翻译不怎么样。

调试的神书

系统的讲的调试软件的方法, 不过, 对我而言没太大作用----这些策略之前就已经会了. 在去坝上草原团建的路上看完的, 感觉一般

调试，不仅仅是修复BUG。

总结软件调试的通用方法，重现问题->诊断->修复->反思。同时指出，自动化测试可以作为调试的辅助，我去……

讲大道理的书籍，可以略读，如果做过一些项目的话，作者说的差不多都能遇到。

内容其实还可以的，虽然比较基础但说得挺实在，日常周围能把基础的事情踏踏实实做好的其实真不多，中文翻译扣一星。

翻译的人真不知道有没有做过开发，bug，check-in这些能不能不翻译

它确实不会发生 -> 它不会再次发生 -> 让软件学会自己寻找缺陷

主要还是方法论的说明..对于我这种项目经验不多的,帮助不大..

与Hack
Debug不同，主要注重整体调试的思路、预防bug的方法、开发中需要引入的方法以及人力在调试中的相关分配与士气把握。
<http://blog.wpc.fun/it/technical/debug/2013/04/21/HowToDebug> 读书笔记见
<http://www.5wpc.info/it/thought/experience/technical/2013/04/21/HowToDebug/>

看完还是要实践, 找bug之路

广

分析问题，模拟真实环境，自动报告运行状态，记录日志，单元测试……

[软件调试修炼之道_下载链接1](#)

书评

很多人光看标题，以为这本书只是在讲blackbox/whitebox testing, unit testing, regression testing, etc. 其实，作者根本没专门把这些Testing 101的内容拿出来讲。书的内容除了debug，还涵盖了很多『好』的软件开发方法。所谓的『好』的开发方法，就是要保证软件质量，保证开...

我看过了 我看过了 我看过了 我看过了 我看过了 我看过了 我看过了 我看过了 我看过了
我看过了 我看过了 我看过了 我看过了 我看过了 我看过了 我看过了 我看过了 我看过了
我看过了 我看过了 我看过了 我看过了 我看过了 我看过了 我看过了 我看过了 我看过了
我看过了 我看...

If you develop software, sooner or later you're going to discover that it doesn't always behave as you intended. Working out why it's misbehaving can be hard. Sometimes very hard. Debug It! is here to help! All bugs are different: there is no silver bullet....

[软件调试修炼之道_下载链接1](#)