格蠹汇编

——软件调试案例集锦

DEBUGGING WARS

张银奎 著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

格蠹汇编

——软件调试案例集锦

DEBUGGING WARS

张银奎 著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

本书以案例形式讨论了使用调试技术解决复杂软件问题的工具和方法。全书共 36 章，分为四篇。前两篇每章讲述一个有代表性的真实案例，包括从堆里抢救丢失的博客，修复因误杀而瘫痪的系统，徒手战木马，拯救“发疯”的 Windows 7，经典阅读器的经典死锁，拯救挂死的 PowerPoint，转储分析之双误谜团，是谁动了我的句柄，寻找系统中的“耗电大王”，解救即将被断网的系统，转储分析之系统挂在 DPC，SDK 安装程序卡壳之谜等。所选案例既有深度，又有较大的广度，从平台角度看有 Windows、Linux 和 Android，从编程语言角度看有 C/C++、.NET 和 Java，从运行模式看既有内核态，也有用户态，从问题的类型来看，有多种原因导致的崩溃和挂死，也有数据混乱，启动、睡眠或者唤醒失败等。第三篇讨论了调试工具和调试系统的设计方法，包括 Windows 8 中的通过以太网络和 USB 3.0 进行内核调试的方法，Android 平台上跨机器调试 Java 应用程序的方法，.Net 调试模型的缺欠以及 CLR 4 重构调试模型的方法，通过 AMLI 调试器调试 ACPI 脚本的方法，双机调试特殊进程的方法，以及设计调试设施应该注意的海森伯效应等。第四篇收录了使用调试器探索计算机世界的若干学习笔记，包括在调试器中细品 CPU，通过调试器观察和解码堆块结构，透视 Windows 8 的新类型应用以及使用调试器监视启动、睡眠和唤醒三大基本过程等。

本书是《软件调试》一书的姊妹篇，延续了《软件调试》的深入严谨风格。但与《软件调试》重在系统介绍调试原理不同，本书重在实践，通过一个个有代表性的真实问题“现身说法”，在软件大背景下介绍调试，通过调试技术解剖软件。本书适合广大程序员、软件测试工程师、软件架构师以及相关专业的高年级学生阅读，也可供信息安全领域的工程师或者研究者参考。

图书在版编目（CIP）数据

格蠹汇编：软件调试案例集锦/张银奎著. —北京：电子工业出版社，2013.3
ISBN 978-7-121-19607-2

Ⅰ. ①格… Ⅱ. ①张… Ⅲ. ①软件－调试－案例 Ⅳ. ①TP311.5

中国版本图书馆 CIP 数据核字（2013）第 030015 号

策划编辑：赵 平
责任编辑：周宏敏
印 刷：三河市鑫金马印装有限公司
装 订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：720×1000 1/16 印张：29.75 字数：762 千字
印 次：2013 年 3 月第 1 次印刷
定 价：66.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

献给我的母亲

学必求其心得，业必贵于专精

——清·章学诚《文史通义·博约》

前 言

在过去十几年中，一种新型的材料风靡全球。它天性柔软，可以任意塑造，用途广泛，几乎无所不能。在这种材料的驱动下，优胜劣汰的规则悄然变化。善于利用这种新材料的一夜成名，抵制或者犹豫徘徊的则迅速没落。这种材料就是软件。有人说，软件在吞噬这个世界。但与其这样说，还不如说人类正在用软件重构自己的文明。

众所周知，人类对软件的依赖越来越大。但天性“软弱”的软件是否能承受如此之重呢？根据我这么多年从事软件工作的经验，这里确实需要画上一个大大的问号。

总体说来，软件的现状很糟糕。借用一句美国同行的话，“没有别的话好说，今天的软件很差劲。”更糟糕的是，很多人没有意识到这一点。

软件领域存在很多问题。略去臃肿缓慢、大量消耗资源、不够安全、稳定性差等外在问题不谈，最致命的问题是在软件行业内部，对软件的误解和错误的价值观大行其道。软件的根本在于代码，但我们却常常背离这个根本，轻视编码工作，空喊如何提高软件质量。

与看得见摸得着的硬件相比，软件生来就抽象难懂。很长一段时间里，只有少数的聪明人会编写软件。随着编程语言和开发工具的进步，编写软件看似变得简单了，但其实这只是一种误解。这种误解导致很多人低估了软件开发的真实难度。于是，错误的认识再加上软件行业的急速发展共同促成了一个可怕的现状：神圣的程序员职业被拉下神坛，沦为软件蓝领，最近几年更被打上了“码农”这样的低价标签。

一方面是人们对软件的要求越来越高，软件的复杂度在提高，另一方面是轻视编码，程序员的素质在降低。于是便形成了今天软件行业中的一个普遍存在的根本问题：软件的复杂度超出了软件开发者可以驾驭的程度，软件质量在最重要的开发源头危机四伏。

程序员是真正为软件大厦“添砖加瓦”的人，程序员的水平高低直接关系着软件的质量优劣。高水平程序员对整个计算机系统融会贯通，写代码时有成竹在胸。

他们写出的代码框架挺拔，细节缜密，而且短小精悍，不枉费“一滴笔墨”。这样的代码给人看时有条有理，如读美文，提交给计算机执行时，轻快顺畅，一蹴而就。而水平差的程序员则相反，很多概念交织脑海中，不清不楚，只知其一，不知其二。写代码时畏首畏尾，东拼西凑。写出的代码杂乱无章，驴唇不对马嘴，冗长拖沓，又臭又长。这样的代码给人看时摸不着头脑，如坠云里雾中，提交给计算机执行时，磕磕碰碰，到处卡壳。一般说来，程序员的水平越低，写出来的冗余代码越多，这是今天软件普遍患有“肥胖症”的一个根本内因。而软件的复杂度和软件的大小密切相关，软件越庞大，软件的复杂度便越高。软件的复杂度越高，会让程序员越发畏首畏尾，惧怕改动现有代码，加入更多的冗余代码，继续增加复杂度，陷入可怕的恶性循环。

代码是软件的根本。写代码是值得修炼一生的一门技艺。提高对代码的感知力和驾驭力是所有软件工作者都必须要修炼的基本功夫。如何修炼呢？学编程语言，学操作系统，学硬件原理，学面向对象，学虚拟机，学云计算，学软件工程……把所有东西通通学一遍？即使都一一学过了，我觉得也还不够，还缺少最重要的融会贯通。

写作目标和书名由来

关于融会贯通，宋明理学之集大成者朱熹早有精彩的论述。据说，融会贯通这个成语就是他发明的。宋代人黎靖德编辑的《朱子语类》收集了大量朱熹与他的学生问答的语录。在卷九《论知行》中记载了这样一段精彩的对话。朱熹教导学生说：“学者喫（吃）紧是要理会这一个心，那纸上说底（的），全然靠不得。”意思是不能只停留在书本上，要用心去理会。但学生们听了后，却有人没有顺着这个思路去理解，天马行空般说出一套空话来：“心之体与天地同其大，而其用与天地流通。”于是朱熹就顺着这个反面典型继续说道：“又不可一向去无形迹处寻，更宜于日用事物、经书指意，史传得失上做工夫。即精粗表里，融会贯通，而无一理之不尽矣。”教导大家不要一味去追求空理论，与其那样，还不如在平常事物上下功夫。

好一个“精粗表里，融会贯通”。反复读这句话，我欣然有所悟，真正从国学中汲取到营养。虽然这句话本来不是关于软件的，但这个道理完全可以用在软件上。在我看来，很多做软件的同行都学了不少的书本知识，但却缺少用心理会，尤其缺少针对实际问题的钻研探索和刨根问底。也就是缺少宋儒们所说的穷理精神。关于“穷理”和“贯通”，《朱子语类》中还有一段很精彩的论述：

穷理者，因其所已知而及其所未知，因其所已达而及其所未达。人之良知，本所固有。然不能穷理者，只是足于已知已达，而不能穷其未知未达，故见得一截，

又不曾见得一截，此其所以于理未精也。然仍须功夫日日增加。今日既格得一物，明日又格得一物，工夫更不住地做。如左脚进得一步，右脚又进一步；右脚进得一步，左脚又进，接续不已，自然贯通。

——《朱子语类》（卷一八）

“治”软件并不比治学问简单，必须把编程语言、操作系统、硬件基础、编译工具、调试器等等一大堆东西理解透彻，“化”为己有，然后才能游刃有余，独当一面。很多人浅尝辄止，“足于已知已达”，“见得一截”，不曾见得另一截。于是练就的只能是“三脚猫”功夫，做做打零的工作或者在人群中当当南郭先生是可以的，但成不了大器。如果能像朱熹说的那样，今日格得一物，明日又格得一物，“左脚进得一步，右脚又进一步”，那么便离成功越来越近了。

上面一段话中，还提到了一个重要的理学概念，那就是格物。何谓格物呢？朱熹曾这样定义：“格物者，格，尽也，须是穷尽事物之理。若是穷得三两分，便未是格物。须是穷尽得到十分，方是格物。”简单理解，格就是穷尽，或者说研究透彻的意思，物即事物和道理。所谓格物，就是要推究事物的机理，与上面的“穷理”很类似。与格物常常一起出现的另一个概念是致知。致知的含义是要不断推进自己的知识，由已知而推及未知。

格物致知都出自儒家的经典著作《大学》，也就是那段广为流传的处世名言：“欲治其国者，先齐其家；欲齐其家者，先修其身；欲修其身者，先正其心；欲正其心者，先诚其意；欲诚其意者，先致其知；致知在格物。物格而后知至，知至而后意诚，意诚而后心正，心正而后身修，身修而后家齐，家齐而后国治，国治而后天下平。”

对于这几句“经文”中的格物致知，朱熹在他的名著《大学章句》中有一段非常好的诠释，摘录如下：

所谓致知在格物者，言欲致吾之知，在即物而穷其理也。盖人心之灵莫不有知，而天下之物莫不有理，惟于理有未穷，故其知有不尽也。是以《大学》始教，必使学者即凡天下之物，莫不因其已知之理而益穷之，以求至乎其极。至于用力之久，而一旦豁然贯通焉，则众物之表里精粗无不到，而吾心之全体大用无不明矣。此谓物格，此谓知之至也。

——朱熹《大学章句》

这段话常被称为“格物补传”，是朱熹在整理《大学》一书时，为缺失章节所做的补充。我非常喜欢这段话，曾经反复阅读和背诵过。在写作这一段内容时，禁不住又读了几遍。不过颇为遗憾的是，我在上大学时并没有读过这段话。香港大学将“明德格物”作为校训，或许那里的学生都读过《大学》吧。

其实，与做学问有所不同的是，做软件有一个得天独厚的优越方法，那就是我

一直主张的基于调试的方法。调试设施是现代计算机与生俱来的固有部分。发明现代计算机的前辈们设计这些设施的目的就是为了帮助人类驾驭这些高速度的机器。利用这些设施，计算机可以暂停、可以慢速前进、可以任由人类摆布。调试设施的一个直接用途就是帮助人们发现计算机软硬件的臭虫（bug）。但如果认为调试设施只是用来抓臭虫的，那么便大错特错了。以调试器为核心的调试设施是征服计算机世界的强大武器，除了用来侦错外，还有很多用途，包括帮助我们探索和学习计算机系统。借助调试设施，我们可以深入到计算机系统的每一个角落，“精粗表里”无处不到。坚持使用这种方法，便可以像朱熹说的那样，“今日既格得一物，明日又格得一物”，“接续不已”，不断坚持，便“自然贯通”了。

过去十来年中，我几乎每天都会用到调试器，我把它比喻为随身携带的一把剑。借助这把剑，我领会了计算机系统的奥秘。借助这把剑，我轻松追赶层出不穷的新技术。借助这把剑，我可以直抵软件的最深处，洞悉其中的精华与糟粕。借助这把剑，我更清楚的看到了软件的现状。

但让我诧异的是，调试技术并没有得到应有的普遍重视。当我向一些同行介绍调试方法时，常常听到惊讶的声音：“还可以这样玩（软件）啊！”

我希望更多的同行能学会用调试器这把剑，因此写了《软件调试》一书分享我所知道的调试技术。因为篇幅限制，《软件调试》偏重理论，实例较少。当时计划以后再分册写案例性的内容。本来以为一两年可以出一本。但实际进展远远没有当初想象的快。好的案例是可遇不可求的。写案例有点像写小说，更需要灵感触发。就这样，差不多五年时间过去了，才积攒出第一册调试案例。在思考书名时，曾经想到用“捉虫记”或者“调试战役”这样的名字，但最终还是受格物一词启发，取名“格蠹汇编”。蠹（音“杜”）的原意是蛀虫，借来指臭虫，代表软件调试。

关于名字中的“汇编”，意思是很多案例汇集在一起，并不是指汇编语言。虽然本书的部分章节中有汇编代码出现，而且附录中列出了常用的汇编指令，但是这不是专门讨论汇编语言的书。寻找汇编语言专著的读者到此赶紧打住，另选他书。这几句话故意另起一段，全为醒目。

总的来说，格蠹是作者新创的一个词，既可以当名词用，又可以当动词用。做名词时代表调试之学，做动词时代表钻研和实践这门学问。之所以新造这个词，旨在希望越来越多的软件同行不再把调试技术看作是捉虫小技，像古人重视格物那样重视这门技术，把它当作一门学问来学习。《格蠹汇编》是作者过去五年中探索和实践这门学问的成绩汇报和学习笔记。

主要内容和阅读方法

这本书讲了很多故事，大多是作者使用调试器这把剑在软件世界中“作战”的故事。故事的时间基本在过去五年中。故事内容的跨度很大，大多数故事是相互独

立的，互不相干。但所有故事的主角基本相同，总是两个：一个是作者本人，另一个是调试器，大多时候是两个主角一起出场，但也有少数例外。

本书正文共 36 章，分为 4 篇，每篇包含 9 章。篇章不是按时间划分的，而是按用途——读者阅读的用途来划分的。4 篇的主要内容略述如下。

第一篇选取了笔者在工作之余偶遇的几个有趣故事，旨在说明如何于“日用事物”中活学活用调试技术，传达宋儒的“知行合一”思想，所谓“学者实下功夫，须是日日为之”，故取名为“笃行”。笃字之意与诚相近，用朱熹的话解释“真实无妄之谓诚”。简单理解，笃行就是“一心一意，坚持不懈地实践”。笃行需要恒心和毅力，如果说有窍门的话，那么便是挖掘乐趣。因为一旦有乐趣，就会“我自乐此，不知疲也”，坚持起来就容易了。本着这样的目的，第一篇所选的故事，重在趣味性，目的是激发大家的学习兴趣。例如，第 1 章讲的是通过搜索内存抢救丢失的博客，第 8 章讲的是联合使用用户态调试和本地内核调试解救挂死的 PowerPoint 程序，第 9 章讲的是 PDF 阅读器的经典死锁。为了让大家可以亲自动手操练一番，本书还安排了 10 个动手实验，第一篇包含 3 个。

如果说第一篇旨在激发兴趣和热身的话，那么第二篇便是真的开战了。第 10 章一开始讲的便是不太好理解的“双误”异常。随之是 Linux 下的后台服务与驱动程序通信时数据混乱（第 11 章），因为补丁安装失败而即将被断网的系统（第 12 章），SDK 安装程序挂死（第 13 章），句柄异常导致的随机崩溃（第 14 章）。而后是两个系统级的挂死，一个是挂在 DPC（第 15 章），另一个是因为驱动程序处理电源事件不当挂死在唤醒途中（第 16 章）。最后是两个应用层的死循环（第 17、18 章）。《中庸》中有“博学之，审问之，慎思之，明辨之，笃行之”（《中庸》第 20 章），又说：“有弗辨，辨之弗明弗措也”，意思是要么不辨别，如果辨别了，不清楚就不停止。调试复杂的软件问题就像走迷宫，也像警察办案，需要分辨真伪，更需要这种“辨之弗明弗措”的探索精神，所以第二篇取名为“明辨”。

第三篇的内容换了个角度，直接以调试工具和调试方法为目标，讲述调试工具本身的设计思路、存在问题和解决方法，包括 Windows 系统内核调试的通信问题和 Windows 8 的解决方法（第 19 章），Android 系统跨机器调试 Java 应用程序的方法（第 20 章），.Net 调试模型的缺欠（第 21 章）以及 CLR 4 重构调试模型的思路（第 23 章），通过 AMLI 调试器调试 ACPI 脚本的方法（第 24 章），双机调试特殊进程的方法（第 25 章），以及设计调试工具需要注意的海森伯效应问题（第 27 章）。这一篇的主要目的是帮助大家深入理解我们手中的调试工具，了解它们的内部构造，熟悉它们的长处和短处。就像战士要了解枪的构造一样，学习调试器是学习软件调试的必修课。熟悉手中的武器，才可能游刃有余，打起仗来得心应手。因此这一篇取名为“器用”，意为武器和工具。

用兵作战，除了武器精良外，熟悉战场地形和拥有丰富的天文地理知识也很重要。软件调试也是一样，只有深入了解计算机世界的“地形地貌”，熟悉其中的“张

三李四王二麻子”，才知道从哪里入手，往哪里发兵。本书第四篇的目的便在于此。篇中收录了笔者最近几年中使用调试器探索计算机世界的学习笔记，分为两类。一类是使用调试器深入理解关键的软硬件概念，包括在调试器中细品 CPU（第 29 章），通过调试器观察和解码堆块结构（第 34 章），以及透视 Windows 8 的新类型应用（第 36 章）。另一类是把调试器当作侦探，监视复杂的系统过程，包括计算机系统的启动（第 30 章）、睡眠（第 31 章）和唤醒（第 32 章）这三大基本过程，以及颇有些神秘的 Windows 7 打电话“回家”的过程（第 35 章）。阅读这一篇将有助于扩大读者的知识面，并且了解关键的细节，让知识既有广度又有深度，所谓“致广大而尽精微”，因此这一篇取名为“致知”。

纵观四篇内容，如果套用兵书里的话，前两篇是战例，第三篇是兵器，第 4 篇是练将——将领指挥作战所需的广泛知识。从针对的问题来讲，前两篇求解的是故障性的问题，即常说的故障处理（Troubleshooting）。后两篇求解的是学习性的问题，也就是探索新知。不管如何划分，4 篇内容的总目标是一致的，就是利用调试方法深入理解软件和计算机系统，温故知新，打通障碍，让知识“融会贯通”。

下面谈一下如何读这本书。首先，因为本书的各章内容相对独立，所以没有必要从第 1 章依着顺序来读，完全可以根据自己的兴趣选择中间的某一章开始读。也可以按照实际遇到的问题来找要读的内容。为了方便大家“对症用药”，附录 C 特意给出了面向问题的一张索引表，比如.Net 应用程序挂死问题对应的是第 13 章和 21 章，驱动程序导致的系统挂死问题对应的是第 15 和 16 章。

第二条阅读建议是希望大家边读边做，也就是遵循“笃行”精神。为了帮助大家顺利上手，我们特意设计了 10 个“亲自动手”实验，附在某些章的末尾，清晰地写出了实验的步骤，并在附录 A 和 B 描述了搭建实验环境的方法。

第三条建议是希望大家制定一个读书计划，然后按计划坚持阅读和做实验。这本书不算太厚，大家可以在一年内轻松读完。全书四篇，正好每个季度读一篇，每个月读三章，如果一周能读一章的话，那么一个月中还可以有一周休息。这样坚持不懈，便可以能像朱熹说的那样“左脚进得一步，右脚又进一步”，离功夫练成那一天越来越近。

在线资源和动手实验

可以通过以下链接访问本书的网站，包括动手实验所需的材料、问题讨论以及勘误信息等。

http://www.advdbg.org/books/dbgwars/

除了下载实验资料和阅读在线信息外，您也可以通过网站提交您阅读本书时遇到的疑问或者报告您发现的问题。

关于封面

这本书写了很多“见鬼了”的问题，所以封面上的主角是民间传说中的捉鬼能手——钟馗道士。这样的设计与我的第一本书《软件调试》一脉相承，代表了这两本书的“姊妹”关系。

感谢

首先感谢《程序员》杂志，为我开设了“调试之剑”专栏，本书的不少文章都是为这个专栏而写的，初稿曾经在杂志上发表过。感谢副社长蒋涛先生大力支持我将这些稿件重新整理和结集出版。

感谢提供本书案例的各位同行和朋友。他们非常大度地允许我将他们的问题公之于众，供大家学习。为了避免给他们带来麻烦，书中故意抹去了可能被“对号入座”的内容。

感谢本书的策划编辑赵平老师，她总是那么支持我的想法，对我一再拖延交稿时间也是那么宽容。感谢本书的编辑周宏敏老师，她是本书的第一位真诚读者，一字不落地阅读了全部书稿，甚至把有些内容读了很多遍。感谢周老师所做的繁重编辑工作，而且纠正了我的不少错误。

感谢我的同事和票友王科平参与了本书取名的过程。科平在部门同事中享有“国学大师”的雅号，他帮助我去粗存精，直到最终想出目前的书名。

感谢郑明华先生将“格蠹汇编”四个字转化为漂亮的行书。我不懂得书法，但很喜欢他写的这四个字，奉为至宝。

感谢我的家人给我的大力支持，他们承担了本该属于我的繁重家务，让我有时间专心写作。

最后，感谢您在茫茫书海中选择了这本书，软件需要大智慧，编程是神圣的工作，让我们用这句话共勉，希望本书对您有所帮助。

张银奎

2013 年 1 月 1 日于上海

反侵权盗版声明

举报电话：（010）88254396；（010）88258888
传　　真：（010）88254397
E-mail：　dbqq@phei.com.cn
通信地址：北京市万寿路 173 信箱
　　电子工业出版社总编办公室
邮　　编：100036

目录 CONTENTS

笃行第一

明辨第二

器用第三

致 知 第 四

笃行第一

知与行，功夫须著并到。知之愈明，则行之愈笃；行之愈笃，则知之益明。二者不可偏废。如人两足相先后行，便会渐渐行得到。若一边软了，便一步也进不得。

宋·朱熹《朱子语类》（卷一四）