DEBUG HACKS™

中文版

——深入调试的技术和工具

吉岡弘隆 大和一洋 大岩尚宏
安部東洋 吉田俊輔 著
马晶慧 译

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

O'REILLY®

Debug Hacks中文版

——深入调试的技术和工具

吉岡弘隆 大和一洋 著
大岩尚宏 安部東洋 吉田俊輔

马晶慧 译

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

内 容 简 介

本书是Miracle Linux的创始人吉冈弘隆和几位工程师们多年从事内核开发的经验积累。从调试器的基本使用方法、汇编的基础知识开始，到内核错误信息捕捉、应用程序调试、内核调试，本书深入浅出地讲解了Linux下应用程序和内核的调试技巧。

虽然本书的出发点是Linux内核调试，但书中的绝大部分知识在许多领域都能派上用场。如Linux应用程序开发，嵌入式Linux开发，甚至时下流行的iOS应用程序开发，只要从事应用程序开发的工作，就会涉及调试，那么读一读本书也不无裨益。

©Publishing House of Electronics Industry 2011.

Authorized translation of the Japanese edition ©2010 O'Reilly Japan, Inc. This translation is published and sold by permission of O'Reilly Japan, Inc., the owner of all rights to publish and sell the same.

本书中文简体版专有出版权由O'Reilly Japan, Inc.授予电子工业出版社，未经许可，不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号图字：01-2011-3386

图书在版编目（CIP）数据

Debug Hacks中文版：深入调试的技术和工具/（日）吉冈弘隆等著；马晶慧译.—北京：电子工业出版社，2011.9

ISBN 978-7-121-14048-8

Ⅰ.① D… Ⅱ.① 吉… ② 马… Ⅲ.① 程序开发工具Ⅳ.① TP311.52

中国版本图书馆CIP数据核字（2011）第134202号

策划编辑：张春雨
责任编辑：付　睿
封面设计：O'Reilly Japan　张健
印　　刷：
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：26.5　　字数：572千字
印　　次：2011年9月第1次印刷
印　　数：4000册　　定价：69.00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。

Debug Hacks 推荐序

凡是程序就有 bug。bug 总是出现在预料之外的地方。世界上第一个 bug 据说是继电器式计算机中飞进的一只蛾子，倒霉的飞蛾被夹在继电器之间，导致了计算机故障。由于这个小插曲，程序中的错误就被称为了 bug。传闻，后来的 COBOL 创始人 Grace Hopper 这位巾帼英雄为纪念这只飞蛾，在日记里记下了这段故事。

从那时起到现在已经过了半个世纪，而 bug 还在不断地产生。程序员之间流传的“格言”之一就是“程序的执行，是根据编写而不是想象”。而且人总是会出错的，程序中必然会混入 bug。从某种意义上来说，也许 bug 昭示了人类的极限。绝大部分 bug 都是给程序员带来小小烦恼的可爱的小东西，但最近 bug 引起的“事故”在报纸上引发热议的事情也时有发生。

程序员的工作就是写程序，然而随着计算机融入社会、程序越来越复杂，要想写出完美的程序简直难于上青天，因此任何程序都必须面对 bug。个人感觉，程序员的一大半时间都花在了寻找 bug、修改 bug 上。

但是，仅仅在黑暗中寻找 bug 是不会有任何效果的。bug 也有 bug 的寻找方法和修改方法。特别是在寻找并确定 bug 位置方面，有着各种各样的技术。“调试”这个词给人的印象是“改正 bug”，但实际上，确定了位置的 bug 完全不可怕，基本上都能立即改正。调试的精髓就是发现并确定 bug 的位置。本书汇集了身经百战的程序员们从经验中得来的 bug 寻找方法和修改方法，特别是平时看不到的有关 Linux 本身的 bug 的 hack，更是无价之宝。可能某些 hack 并非大多数程序员日常用得到的，但即便如此，其思路也有参考价值。特别感激的是本书详细地解释了 GDB、

Valgrind、oprofile 等方便的工具。此外，本书还涉及 Ruby 的两个“bug”，对此我也十分感激。

希望本书能成为诸位程序员的“启明灯”，让我们在每天与 bug 的斗争中更顺利。

2009 年 3 月于羽田空港

松本行弘[注1]

注 1：　松本行弘（Yukihiro Matsumoto），Ruby 语言创始人。——译者注

此为试读，需要完整PDF请访问：www.ertongbook.com

鸣　谢

关于作者

吉冈弘隆（Hiro Yoshioka）

Miracle Linux 的程序员，毕业于庆应义塾大学研究生院。从 1999 年开始主持名为“内核读书会”的 Linux 技术交流会，曾就职于日本数字设备公司（DEC）研发中心、日本甲骨文，后创立 Miracle Linux。大学毕业后参与了多款软件产品（COBOL 日文版、DEC Rdb、Oracle 8、MIRACLE LINUX、Asianux 等）的开发。

吉冈弘隆还担任了 JIS X0208:1990/X0212:1990 的标准化，U-20 编程大赛的评委，以及安全编程夏令营的部门主审。

2008 年，吉冈弘隆获得了经济产业省商务情报政策局长的感谢信和乐天技术奖 2008 年金奖。

博客“ユメのチカラ”（梦的力量）：http://blog.miraclelinux.com/yume/。

“未来のいつか/hyoshiok の日記”（有朝一日/hyoshiok 的日记）：http://d.hatena.ne.jp/hyoshiok/。

大和一洋（Kazuhiro Yamato）

任职于 Miracle Linux 的软件工程师。之前主要工作内容为 Linux 内核和 GLIBC 周边，最近也开始着力于 gstreamer 等媒体相关 hack。不善使用 GUI 工具，开发环境完全使用 VIM + GCC + GDB。

大岩尚宏（Naohiro Ooiwa）

任职于 Miracle Linux 的软件工程师。从事开发业务，但更多的是调查并分析内核，负责解决驱动程序、网络、文件系统等大范围的许多 bug，号称任何 bug 都能解决，近期在验证 Linux 的节能功能。擅长绘画，但不善日语和英语。经常被上司催促给社区出补丁，与日本的 Linux 开发者代表人物见面的机会很多。

安部东洋（Toyo Abe）

任职于 Miracle Linux 的软件工程师。初入 Linux 内核界时连 Hello World 程序都写不好，那段经历真是让人几欲落泪。到目前只涉及了 x86 架构，写完本书之后打算挑战 ARM。

吉田俊辅（Shunsuke Yoshida）

任职于 Miracle Linux 的系统工程师，不论在哪里都自称为普通人。曾先后就职于地方性软件企业和制造业系统集成企业，后加盟 Miracle Linux。主要负责操作系统、数据库、网络、虚拟化等基础设施。参加了小江户 LUG [注1]、YLUG（横滨 Linux User Group）、USAGI 补完计划[注2]等关东附近的开源社区。参加过各种活动、展出，以及写书。

博客“第三のペンギン”（第三只企鹅）：http://blog.miraclelinux.com/thethird/。

关于贡献者

岛本裕志（Hiroshi Shimamoto）

软件工程师。主要通过解决问题来学习 Linux，特长为分析 core、crash。活跃在 x86、任务调度（scheduler）、实时系统等领域的 Linux 内核社区中。

注 1：小江户らぐ，全称小江户 Linux User Group，以日本崎玉县川越市为中心的 Linux 用户组。此处译为小江户 LUG。——译者注

注 2：USAGI（UniverSAl playGround for Ipv6）项目致力于开发 Linux 上的高质量 IPv6 和 IPsec（用于 IPv4 和 IPv6）的协议栈，而 USAGI 补完计划是改善该项目的俱乐部。——译者注

美田晃伸（Akinobu Mita）

Fixstars 公司程序员。不太明白调试器的用法，调试时使用 printf[注3]。

致谢

本书的灵感来源于传世之作《BINARY HACKS——ハッカー秘伝のテクニック 100選》[注4]一书。执笔本书的不仅有 Miracle Linux 的员工，贡献者岛本裕志、美田晃伸也提供了很有意思的 hack。在此表示衷心的感谢。

此外，本书的推荐序来自于 Ruby 编程语言的缔造者松本行弘。在此表示真挚的感谢。

——吉冈弘隆

注 3：Linux 内核的 Fault Injection 的作者和维护者。

注 4：译者注：本书的中文版《Binary hacks——黑客秘笈 100 选》已由中国电力出版社出版，ISBN：9787508387932。

前 言

本书主要内容是程序员在编程过程中无法避免的调试过程。虽然调试技术不依赖于任何编程语言和开发环境，任何编程都无法避免，但调试领域却很难总结，也几乎没有任何参考书。

编程入门读物多如牛毛，为何调试的入门读物却屈指可数？

思考一下编程中的设计、编码、测试和调试等过程，就会发现在调试上花费大量时间的情况并不罕见。实际上，软件开发成本中的大部分并没有用在设计新软件上，而是用在扩展、改变软件及修改软件错误上。

与设计新软件相比，有时会感到调试更加困难。尽管设计、编码和测试相关的最佳实践以书籍的形式广为流传，但软件开发的重要一环——调试，相关的入门书籍却寥寥无几，真是有点奇怪。

调试是种极端个性化的工作，10 个人就会有 10 种调试方法。而且，调试中有高手，也有菜鸟，也有像拥有魔法一样，哼着歌就能找到 bug 并改正的 hacker。

执笔本书之际，心底的想法之一就是以我们遇到过的事情为中心，明确地介绍具体的调试方法。

我们写下我们自己的调试方法，同时也加深了对自己的调试方法的理解。为什么使用这条命令调试？最初是怎样找到这个 bug 的？经过这种反反复复的自问自答，调试的过程也经历了考验。

每条 hack 都是实际工作中遇到的事情。有的例子中为了简化说明，重新书写了测试程序等，但那些也都是在实际遇到的 bug 的基础上进行的说明。举出实际例子，就可以写出经验之谈，避免纸上谈兵了。

在写出的调试过程中，哪怕我们的调试方法还有很多改善余地，也要明确地写出来，起到抛砖引玉的作用，这种写法尽管质朴却十分重要。倒不如说，我们相信正是不断积累这种拙劣的做法，调试方法才会进步。

比我们的方法更好的方法一定存在。讨论好的调试方法需要一个舞台作为基础，这正是本书永恒不变的愿望。特别是希望诸位 hacker 们（大牛程序员）与自己的方式对比着阅读，针对我们的方法的适用范围、优缺点等各种观点进行讨论，这才是我们专业程序员对调试更深入的理解。

调试方法，与其说是本书中明确写出的东西，不如说是从各自的经验中得来的、某种秘籍一样的存在。作为程序员，在不断的钻研和积累中掌握调试技术，才是我们专业程序员的基本能力。

此外，像我们这种把自己的调试方法广泛公开的行为，如果能普遍流传的话，就能与更多的人共享最佳实践，而这些实践是我们程序员的无价之宝。

随着工具和开发环境的进步，今后调试方法也会发生变化，而且调试方法也会随着我们自身对于调试和编程风格的理解而逐步进化。为了主动地学习这些知识，我们希望通过本书，和诸位程序员一起学习进步。

阅读本书的必要知识和读者对象

本书的读者对象是主要用 C/C++等编程语言进行开发的应用程序程序员和 Linux 内核开发者。本书并没有限定语言和开发环境，但在示例中均使用 Linux。要进行底层调试，计算机体系的基础知识、编程语言的基础知识是必要的。此外，开发环境方面，UNIX 编程环境的基础知识也是必要的。除此之外的知识并没有特别要求。

本书读者对象为自己进行编程设计、实现、测试和调试工作的初级到中级程序员。本书面向那些希望加强自己的编程技术的人。不仅是 C/C++程序员，对于使用 Perl/PHP/Python/Ruby 等脚本语言进行编程的人们来说，哪怕语言和工具各不相同，本书中的许多方法也具有参考价值。此外，对于使用 Windows、Mac 等不同平台的人来说，本书的思考方式同样可以参考。

我们特别希望诸位学生们阅读本书。如果在读过一遍编程语言入门书之后，希望更深入地理解编程的话，本书中的 hack 可以用来参考。我在编辑本书时就在想，要是在学生时代就有这样一本书该多好啊。

使用脚本语言编程的人们，平常几乎不会意识到计算机体系和机器语言等，但是，如果 Ruby 处理程序发生 segmentation fault 而突然崩溃，为改正错误，就要用到本书中的知识和技术。对于想把程序员之路走得更宽的人来说，本书是个好的起点。

此外，我们也希望拥有各自调试风格的熟练的 hacker 们阅读本书，并提出宝贵的意见。特别是，市面上几乎没有正面讨论 Linux 内核调试的参考书，因此在我们给本书设置的范围、读者对象及书籍结构等方面，如果能获得您的意见反馈，真是不胜荣幸。

本书内容范围

我们选择的例子主要是 Linux 上的应用程序和 Linux 内核本身，其原因只是我们在从事这方面的工作。

说起编程，还有 Web 应用程序、嵌入式、游戏、中间件等各种各样的应用领域。尽管各个领域都有特殊的调试方法，但本书并不涉及。虽然包罗万象的全能调试方法并不存在，但本书主要讨论更为通用的调试方法。因此，许多情况下这种思路可以灵活运用。

本书包含的内容

本书介绍调试的基本思路和方法。不仅包括对应用程序的调试，也包括对操作系统（Linux 内核）的调试。此外也会涉及 GDB 等调试器的使用方法、转储文件（dump）的查看方法、crash 的使用方法，以及 kprobes 和 oprofile 等调试工具。

本书涉及的工具之外，还有许多优秀工具，例如 ftrace、LTTng、dmalloc、blktrace、lockdep、kgdb、KDB、utrace、lockmeter、mpatrol、e1000_dump、git-bisect、kmemcheck 等。关于这些工具，我们期待看到诸位读者的 Debug hacks。

本书不包含的内容

本书不包含编程的一般方法，例如软件设计、容易调试的编码方法和测试方法论等。TDD（Test-driven development，测试驱动开发）是集测试和调试于一身的开发过程，但并不在本书的讨论范围。

此外，本书也不涉及一般疑难解答（troubleshooting）中众所周知的故障发生时的问题区分、迂回策略（workaround）的提出等。

此为试读，需要完整PDF请访问：www.ertongbook.com

本书主要着眼于发现 bug 之后的修改过程，这个狭义的调试。

本书结构

“第 1 章 热身准备” 概要地讲述调试是怎样的过程，并介绍本书的整体情况。

“第 2 章 调试前的必知必会” 介绍调试的基本知识，包括调试器（GDB）的用法、Intel 架构的基础知识、栈的基础知识、函数调用时的参数传递方法、汇编语言的学习方法等。

“第 3 章 内核调试的准备” 介绍 Linux 内核调试方法的基础，包括 Oops 信息的阅读方法、串口控制台的使用方法、通过网络获取内核信息的方法、SysRq 键、各种转储的获取方法、crash 命令的使用方法、用 IPMI 和 NMI watchdog 获取崩溃转储、内核特有的汇编语言等内核调试的基础知识。

“第 4 章 应用程序调试实践” 讲述用户应用程序的实践性的调试方法。通过实例介绍各种调试方法，包括栈溢出导致的 segmentation fault（SIGSEGV）、backtrace 无法正确显示、数组非法访问导致的栈破坏、灵活应用监测点检测内存非法访问、malloc()/free()中发生的故障、应用程序停止响应等。

“第 5 章 实践内核调试” 介绍内核故障的基本调试方法，包括 kernel panic（NULL 指针访问、链表破坏、竞态条件）、内核停止响应（死循环、自旋锁、信号量、实时进程）、运行缓慢、CPU 负载过高等各种异常情况的调试方法。

“第 6 章 高手们的调试技术” 汇集了广泛的内容，如调试时的各种工具、小技巧等。介绍的工具、技巧有 strace、objdump、Valgrind、kprobes、jprobes、KAHO、systemtap、proc 文件系统、oprofile、VMware Vprobe、错误注入、Xen 等。其他还有 OOM Killer 的运行和原理、通过 GOT/PLT 进行函数调用的原理和理解、initramfs、用 RT Watchdog 检测实时进程停止响应的方法、调查手头的 x86 机器是否支持 64 位的方法等。

“附录 Debug hacks 术语的基础知识” 介绍本书出现的术语。阅读各篇 hack 时如果遇到不理解的术语，可以参照该附录。

本书的使用方法

本书除了第 1 章之外，并没有假设阅读顺序。有一定基础的读者可以随意翻阅感兴趣的章节。如果想掌握基础知识，可以认真阅读第 1 章和第 2 章，并阅读参考文献等。而对于玩转内核的高手们，阅读本书所介绍的工具的使用方法等，也可能会感到甘之如饴。如果读者能告诉我们一些 hack 技巧，我们会感到十分幸运。

本书的体例

等宽字体（sample）

表示文件名、源代码、输出、命令等。

粗体等宽字体（sample）

表示用户输入的内容。

表示提示、建议等。

表示注意和警告等。

各 hack 左侧的温度计图标表示各 hack 的难易度。

初级

中级

高级

中文版书中订口处的“ ”表示原书页码，便于读者与原日文版图书对照阅读，本书的索引所列页码为原日文版页码。

意见和疑问

我们尽最大努力对本书的内容进行验证和确认，但难免出现错误或容易导致误解的说法，也可能有印刷错误。阅读本书时如发现问题，请联系我们，以便再版时改正。

日本：

株式会社 O'Reilly Japan

东京都新宿区坂町 26 番地 27 Intelligent Plaza Building 1F 邮编 160-0002
电话：03-3356-5227
FAX：03-3356-5261
电子邮件 japan@oreilly.co.jp

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）
奥莱利技术咨询（北京）有限公司

有关本书的技术问题和意见，请发送至以下电子邮件：

info@mail.oreilly.com.cn
japan@oreilly.co.jp

本书的网页、示例代码[注1]、勘误表和追加信息位于：

http://www.oreilly.co.jp/books/9784873114040/

关于 O'Reilly 的其他信息请访问下面的网站。

http://www.oreilly.com.cn/（中文）
http://www.oreilly.co.jp/（日语）
http://www.oreilly.com/（英语）

注 1： 这些示例代码是笔者在执笔时使用的程序，并不保证能在任何环境下正常运行，而且可能会在没有预先通知的情况下做出修改。另外，本书不为示例代码提供任何支持。

目 录

此为试读，需要完整PDF请访问：www.ertongbook.com