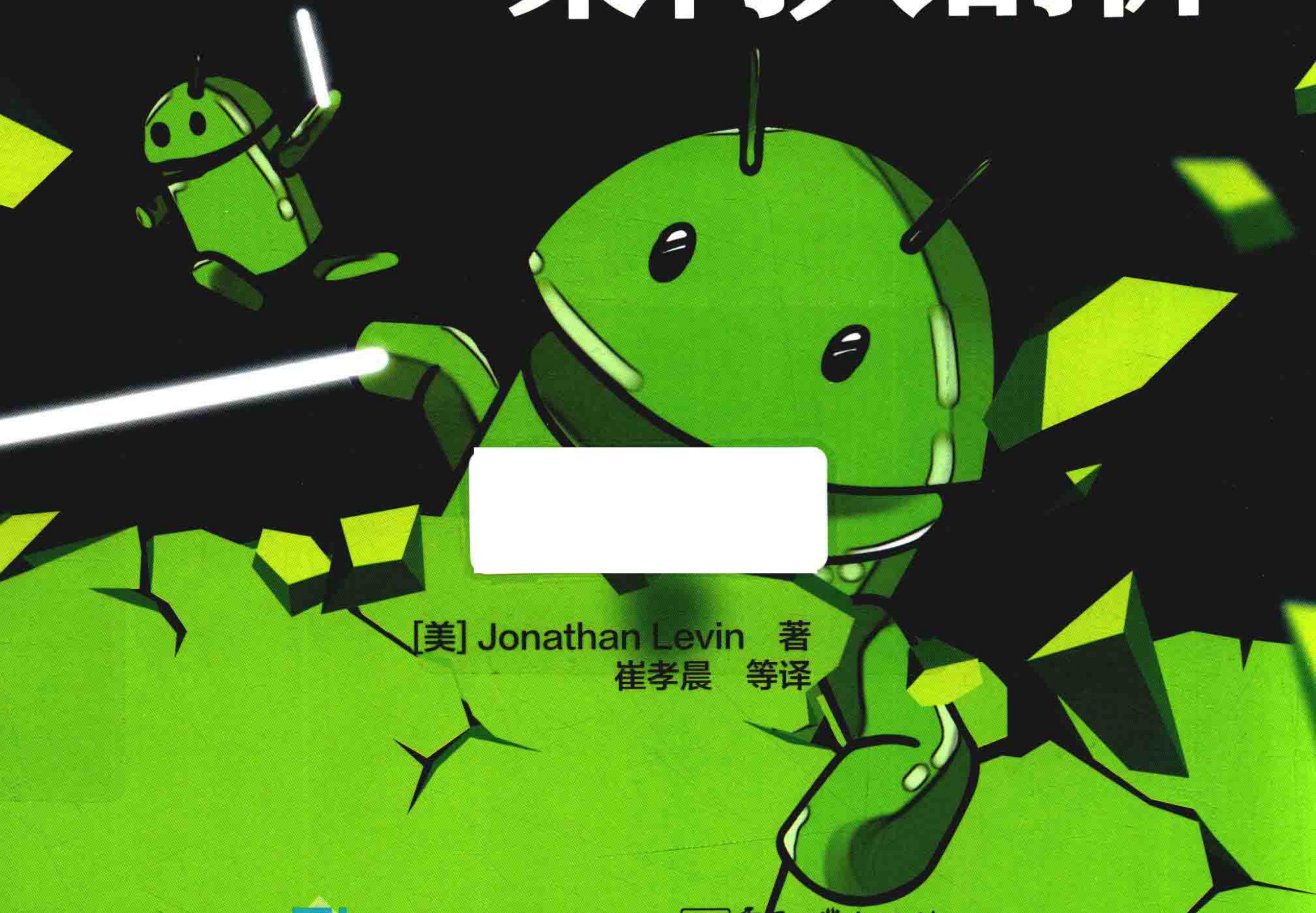
Broadview®
www.broadview.com.cn

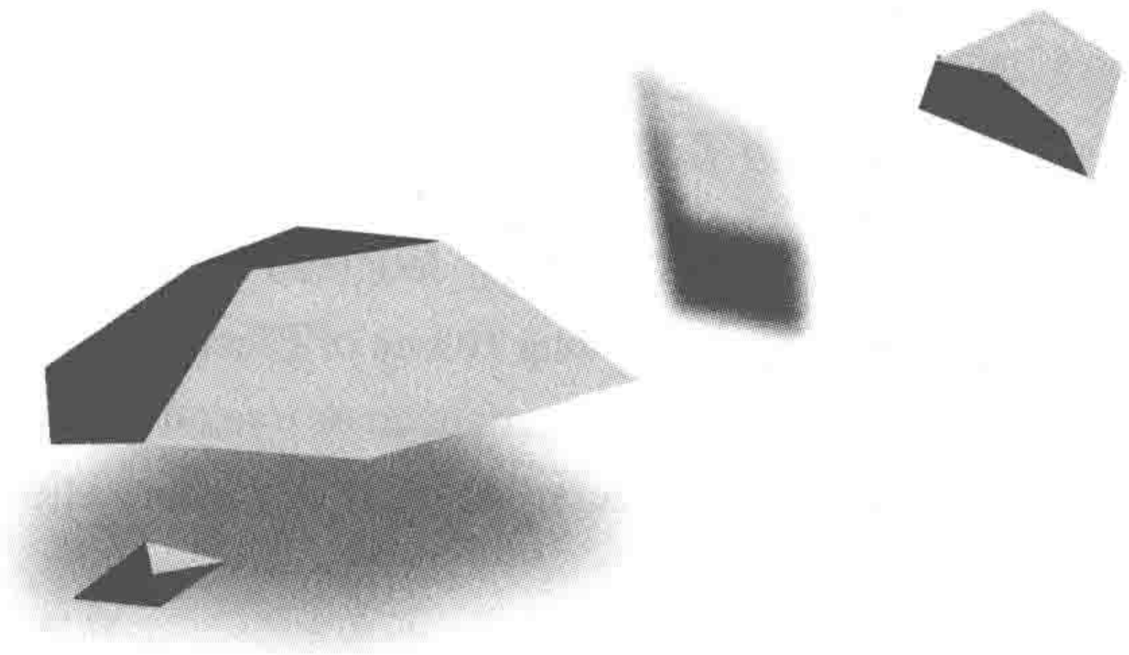
Android Internals:the Power User's View

最强Android书
架构大剖析



[美] Jonathan Levin 著
崔孝晨 等译

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn



Android Internals:the Power User's View

最强Android书
架构大剖析

[美] Jonathan Levin 著
崔孝晨 等译

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

内 容 简 介

本书通过实验而不是源码，将 Android 系统层层拆解，令读者深刻透彻地掌握 Android 系统的内部技术：以 init 进程为切入点详细阐述了 Android 的启动过程和关键服务；从 Android 作为资源协调者和服务提供者的角度，重点分析了 servicemanager 和 system_server 这两个进程。同时，作者比较了 Linux 与 Android 系统的区别，并对 Android 系统的安全性做了深入的阐述。

本书采用了大量的图表示例和实验，表达新颖清晰，让读者能直观地掌握 Android 的技术精髓。

本书适合广大移动开发者及对 Android 系统感兴趣的人员阅读。

版权贸易合同登记号　图字：01-2017-1884

图书在版编目（CIP）数据

最强 Android 书：架构大剖析 /（美）乔纳森·列维（Jonathan Levin）著；崔孝晨等译. —北京：电子工业出版社，2018.7

书名原文：Android Internals: the Power User's View

ISBN 978-7-121-31813-9

Ⅰ. ①最… Ⅱ. ①乔… ②崔… Ⅲ. ①移动终端－应用程序－程序设计 Ⅳ. ①TN929.53

中国版本图书馆 CIP 数据核字(2017)第 130062 号

策划编辑：刘　皎
责任编辑：白　涛
印　　刷：三河市双峰印刷装订有限公司
装　　订：三河市双峰印刷装订有限公司
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：787×980　1/16　印张：22.5　　字数：468 千字
版　　次：2018 年 7 月第 1 版
印　　次：2018 年 7 月第 1 次印刷
定　　价：89.00 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）88254888，88258888。

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。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

推荐语

这本书的确是目前一流的 Android 书。

——wushi（吴石），腾讯科恩实验室负责人

一本对 Android 底层架构全面、深入剖析的书，结合 Linux 有针对性地帮助读者从整体上把握 Android 架构的整体知识，并对每个模块都做了十分详尽的解读，帮助读者从细节上掌握每一个模块的要点。

——张鸿洋

也许你是刚入行的 Android 菜鸟，也许你已经是有丰富经验的 Android 高工，但是每个 Android 开发者都应该阅读一下这本书，它会让你了解真正的 Android，让你对 Android 底层系统有一个全新的认识。

——Stormzhang，公众号： stormzhang

作者用“上帝”的视角，向我们展示了一个 Android 系统的设计与架构，庖丁解牛般地让读者无须接触大量源代码就能了解整个系统的实现思想，而这是比源代码更加重要的东西。相信读者在这本书的指引下一定会对 Android 系统有更加深入的理解和认识。

——徐宜生，《Android 群英传》作者

这本书很适合用来学习、研究 Android 的系统架构。书中对比了 Android 与 Linux 系统，涵盖了文件系统，框架服务架构和安全等各个方面，为我们展现了具体且全面的 Android 系统的内部细节。此外，作者条理清晰，擅于将复杂的事情讲得简单透彻，显然造诣相当深厚。

本书可谓是了解 Android 系统内部技术的不二之选。

——段建华，技术小黑屋（droidyue.com）博主，公众号：droidyue_com

推荐序一

Android 是当今最主流的移动端操作系统，然而作为安全研究者要找到一本适合入门学习的书籍却并不容易。本人总结其原因有三：第一，Android 操作系统更新周期较短，特别是近两年 Android 自 4.4.X 更新至 7.1 版，它的系统安全特性已经发生了翻天覆地的变化，许多 Android 书籍自开始撰写到完工就需要几年时间，如果是英文书籍还涉及翻译的时间，通常读者拿到书的时候，内容已经比较过时了。第二，由于 Android 系统的复杂性，对作者的技术要求比较高。作者不仅要熟悉其原生（Native）层，对其 Java 层等组件也需要有了解。市面上的很多 Android 书籍，很少能较好地覆盖每一个面向，或者是只有一个侧重点，这导致读者即便通读全书，也无法了解 Android 的全部。第三，很多书籍从 Android 源代码入手来讲解原理，虽有足够的深度，但略嫌乏味，会给读者一种纸上谈兵的感觉，给读者的阅读增添了困难，令他们很难全部读完并完全理解。

《最强 Android 书：架构大剖析》是我见过的 Android 书籍中，最适合安全研究人员阅读的一本。此书的作者 Johnathan Levin 和译者崔孝晨都是本人的朋友，相信与这两位打过交道的朋友都会发现，他们精力非常充沛，虽然已经从事安全研究十几年，但对新技术仍然充满了热情和好奇心。在面对面交流的时候，他们经常对着一个技术点侃侃而谈、乐此不疲。而作者 Johnathan Levin 更是在本书中引入“互联网思维”，为本书设立了网站 http://newandroidbook.com，并不定期撰写文章，听取读者的反馈和建议，把 Android 最新的、读者最想了解的特性分析更新到本书中。以上特性，保证了本书与那些“拿到就过时”的 Android 书籍相比，具有明显的优势。

一本好书，光与时俱进、有技术深度还远远不够，如何把一个复杂的操作系统的内部机制和原理，合理有序、循序渐进地传授给读者，也是一个需要推敲的问题，而这就不仅仅是要求作者技术功底深厚那么简单了。Johnathan Levin 多年从事技术培训工作，这些积累的培训经验确保了本书的易读性：细心的读者很快就能发现，本书的每一章节都相对独立，无论是顺序阅

读或者跳着看都没有太大的问题；书中大量使用图表、图片来叙述，让读者更直观地掌握各个知识点，并且通过实验的方式加深对各个知识点的印象，充分掌握一些比较重要的概念。而译者崔孝晨同志更是在确保把原书的含义完整无误地传授给读者的同时，加入了许多中国元素，在表达上更为生动形象——这样的合作，无疑是中国读者的一大福音。

Android 的安全防护机制是多维的，我的团队成员何淇丹、刘耕铭在 Mobile Pwn2Own 2016 中远程攻破搭载最新 Android 7.1 的 Nexus 6p 设备，从攻破所利用的漏洞来看，很明显，安全研究者需掌握 Android 浏览器、框架组件、内核等安全特性并找出 Java 层、Web 相关、原生层甚至内核层的漏洞，并串联在一起才能对 Android 进行有效的远程攻击、突破沙盒，最终实现提权。本书对 Android 安全特性的分析也是一大亮点，很好地覆盖了目前针对 Android 的攻击面。相信阅读本书一定会对您的工作有所帮助。

陈良　科恩实验室高级研究员

2018 年 6 月于上海

推荐序二

自 2008 年 Google 发布 Android 的第一版以来，时至今日，无论是在系统特性、用户规模还是生态规模上，它都取得了惊人的进展，获得了移动操作系统领域的绝对优势。Android 是开源的，这对于任何想要一探究竟的人都提供了非常大的便利，但同时由于 Android 系统本身日趋复杂，对大家也是个很大的挑战—— 一不小心就会陷进代码的汪洋大海之中。

Jonathan Levin 作为操作系统领域的专家，依赖自己深厚的技术功底和多年的研究，独辟蹊径地分别从高级用户和开发者的角度来探索 Android 系统。读者手上的这本书是从高级用户的角度开始 Android 的探索之旅的。这本书我首先接触的是英文版，现在非常高兴看到这本书的中文版面世，能让更多的读者受益。

本书特别适合高级用户（MIUI 称这部分用户为发烧友）学习使用。目前不少手机用户对各种硬件拆机评测很熟悉，这本书有如一个软件拆解，作者有如庖丁解牛一般，把运行在手机中的 Android 系统逐层拆解。在简要地介绍了 Android 的版本演化历史之后，作者先从分区和文件系统开始，详细介绍了各个分区的作用，各个分区上存储的内容和数据，还用实验详细演示了如何制作一个刷机包。在介绍了这些静态的软件组成之后，作者开始详细探索这些静态的内容是如何动态工作的。书中以关键的 init 进程作为切入点，详细阐述分析了 Android 的启动过程；接着分析了启动过程中的关键服务：原生服务和 Android 框架服务。操作系统有两个重要的角色：资源的协调者和服务的提供者。作者重点分析了 servicemanager 和 system_server 这两个进程，它们构成了 Android 系统所扮演的两个角色的基石。

对 Android 系统有一点了解的读者可能知道，Android 是基于 Linux 内核的，那么 Android 和一个常用的 Linux 系统有何不同？作者接下来就从一个 Linux 用户的视角来观察和分析 Android 系统，剥去构筑在 Linux 内核之上的那层 Android 外衣，让一个熟悉 Linux 系统的人跃跃欲试：“我也能构建一个 Android 系统”。本书最后概要性地讲述了一些 Android 的安全机制，

虽然只有短短的一章，但是非常清晰，尤其是对 selinux 的描述。从上述的脉络可以看出，作者动静结合，抽丝剥茧一般把运行在手机里的 Android 系统清晰地展示在大家眼前。

本书虽然是从高级用户的角度来探索 Android 系统的，但也很适合 Android 开发者，尤其是 Android 系统工程师学习。要想剖析一个系统，得先了解使用它。这本书有如一盏指路明灯，让我们在 Android 代码的汪洋大海之中始终明确前进的方向。略有遗憾的是这本书来得有点晚，使我们在学习 Android 系统的过程中走过一些弯路，不过今天的读者可以幸运地站在大师的肩膀上了！在小米 MIUI，我们也打算使用其中的部分内容作为内部培训材料。如果您正好打开本书看到了这篇序，诚邀您一起开始我们的 Android 系统探索之旅，这将是一个妙趣横生的旅程。谢谢！

汪文俊　MIUI 系统平台部总经理

2018 年 6 月

译者序

市面上关于 Android 的书籍可谓汗牛充栋，我甚至都不敢把书名 *Android Internals* 按照惯例译为《深入理解 Android 系统》——重名的书太多了。那么为什么还要把这本书介绍给国内的读者呢？因为市面上绝大多数的 Android 书籍都是从程序员的视角展开的，入门的门槛相对比较高。尽管开发 Android App 的程序员们自然应该对 Android 系统有一个深入的理解，但这并不意味着其他人并不需要理解 Android 系统。比如，电子取证人员，他们需要对 Android 中的文件系统及数据存放位置有一个清晰的认识，以便从中提取相关数据；喜欢折腾的技术发烧友，root 掉系统之后一般都喜欢自己修改一下系统，比如禁用一些开机启动项之类的。如果无须依赖额外的 App，只需一个文本编辑器就能完成相关修改，甚至给系统换上自己的开机动画岂不是很酷……诸如此类。但这些人中只有很少的一部分接受过正规的编程训练，因此市面上大部分的书籍对他们来说难度就太大了。

本书的作者 Jonathan Levin，也是畅销书 *Mac OS X and iOS Internals: To the Apple's Core*（中文版为《深入解析 Mac OS X & iOS 操作系统》）一书的作者。按 Jonathan 自己的说法，*Mac OS X and iOS Internals: To the Apple's Core* 一书的读者反馈中，反映最激烈的问题是：太技术化了！许多读者读起来感到头大！所以在这本后继的 *Android Internals* 中，他把不需要代码就能表达清晰以及与开发人员关系不太紧密的部分放在这一本书中，而把剩下的、与开发紧密相关的部分放在了另一本书中。这一点从本书英文版的副标题“*Power User*”就可以看出来。那么什么是“Power User”呢？如果一定要和传统的桌面系统的用户相对应，这个“Power User”就相当于系统管理员（administrator）的角色。相对于普通用户，他需要对系统有更加深入的理解，能对系统进行更加详细的配置，因而也被认为可以拥有较高的权限（比如 root）——本书的部分实验确实需要拥有 root 权限，且第 8 章中也专用了一个小节讨论 root 这一主题。

有人问，既然是讲系统内部实现，不讲编程又是如何把它讲清楚的呢？答案是使用实验。

本书的内容是根据作者多年讲课的讲义，整理、精选[1]而来，通过在 ADB（Android 调试桥）中执行各种命令的方式（相对于阅读代码），比较直观地向读者揭示 Android 内部的工作原理。效果如何呢？别看广告，看疗效。上次曝出的 CIA Value7 的相关内容显示，这本书已经被 CIA 私下盗版，用于 CIA 特工的内部培训了。而可怜的 Jonathan Levin 既不敢告 CIA 侵权，又不能告 WikiLeaks……，只好在本书官网上提供了已经被泄密的 2015 年 6 月版的英文版的免费下载链接——与其去 WikiLeaks 下载，不如上官网下载。不过读者也不必沮丧，自我开始本书的翻译以来，几乎每个季度都会收到 Jonathan Levin 发来的大量更新——其中包括历次 Android 系统更新的新内容，以及书中已经发现的一部分错误的更新（包括一些我发现的错误:)），使得我也不得不多次将译稿做一些必要的返工，目前出版的中文译本是以 2016 年 11 月底的最新版本为准（更新至 Android Marshmallow PR1 版）的，您大可不必担心白花银子。

在本书的翻译过程中，我们力求将原文准确、清晰地翻译成中文。有模糊不清之处，我们尽量通过与作者沟通、阅读源码和实验的方式搞清楚。但各类缩写还是本着忠实原文的原则，沿用原文的写法。如在本书中，Android JellyBean 版会被缩写成 J 版或 JB 版，Android Lollipop 版会被缩写成 L 版，Android Marshmallow 版会被缩写成 M 版等。

本书由上海公安学院的教师教官完成翻译，第 1 章由殷方老师翻译，第 2 章由王宏老师翻译，其余章节由我翻译，全书由我统一校对，并经本书作者 Jonathan Levin 及其国内合作培训公司的同志审校。

最后感谢电子工业出版社刘皎老师在本书翻译过程中给予我们的有力帮助，感谢腾讯公司科恩实验室吴石、陈良、赵泽光等老师给本书初稿提出的宝贵意见。

囿于译者水平有限，书中必然存在疏漏之处，敬请读者不吝指正。

崔孝晨

2018 年 6 月

---

1 确实是精选而来的，书中第 7 章的“wchan 和 syscall 文件”一节中有“你也可以使用*上个实验*中给出的方法，解析程序计数器（program_counter）的值”字样，但之前的那个实验与这一节毫无关系。后经与作者确认，这里原本另有一个实验，在正式出版时删除，但是忘记相应地修改这里的语句，而遗留下了这个问题。

致中国读者

此刻没有什么事情比本书中文版的出版（经过几年的努力）更令我开心的了！在过去的几年中，上海已经成了我的第二故乡——我经常过来给开发者们培训 Linux、Android 和 MacOS 相关的内容。

本书的翻译历时较长（尽管还没有长到和本系列第二本书的写作一样）。在此过程中，Android 在全世界尤其是在中国的受欢迎程度与日俱增——华为、小米和其他手机厂商已经渐渐成为一流的智能手机生产商——他们打造了更好的设备并将这种能力扩展成一个丰富的生态体系。

我希望本书能帮助更多的人加深对 Android 系统的理解；我希望它能启发与操作系统交互时的创新的、激动人心的思考——令 Android 系统更优秀、更强大而且更有效率。

我本人非常乐意接受各种改进，所以请记得访问 http://NewAndroidBook.com/，让我知道你的所思所想！

关于本书

概览

购买了本书的朋友，毫无疑问你已经意识到了 Android 的重要性。这个启动于 2003 年的操作系统，在被谷歌收购之后，现在已经成为谷歌最得力的产品。它迎头赶上了苹果公司的 iOS 操作系统（也有人说是很接近了），不仅取得了移动操作系统领域内的绝对优势（截至本书付印时，它的市场占有率已经达到了令人惊异的 82%了），而且还渗透到了其他平台上，成为可穿戴设备、TV 和嵌入式设备上的操作系统。

Android 是开源的，而且是可以免费获得的，这也就意味着任何人都能获得它，并对它进行修改，使之能够运行在任何一种平台上——事实上，这也是它能够力压群雄，占据市场主流的原因。不过，令人吃惊的是，尽管已被广为接受，但是至今仍然没有一本书来完成探究其内部工作原理并将其文档化的任务。前几年有一本名叫《构建嵌入式 Android 系统：移植、扩展和定制》[1]的书，作者是 Karim Yaghmour——这本书给出了大量关于该操作系统通用结构的细节信息，但是其着眼点在于如何创建和修改源码，使之能运行在各种新的平台上，而没能给出操作系统本身的结构。事实上，在此书“内部结构入门”一节中，Yaghmour 声称“要想完全理解 Android 系统服务的内部结构，无异于蛇吞象”。

我认为这还算是一种保守的说法，这也就是为什么本内容需要由好几本书组成而不是只有

---

1 该书的英文版书名是 *Embedded Android: Porting, Extending, and Customizing*，中文版书名为《构建嵌入式 Android 系统》（秦云川、肖淇译，中国电力出版社，2015 年 8 月出版），英文版书名中的“*Porting, Extending, and Customizing*”在中文版书名中没有译出，但为了方便读者理解下文，这里将其译出。——译者注

一本的原因。第 1 本（也就是你现在正在读的这本书）主要是从高级用户或者管理员的角度讨论 Android。在这一本中，我试图从各种不同的角度，如 Android 的设计、文件系统结构、启动顺序、原生服务再加上 Linux 基底以及 Linux 基底对操作的影响来讨论这一操作系统。这一切都不涉及代码，只是试图尽可能地给你一个大致的概念和鸟瞰图。从某种程度上说，本书可以算作 Yaghmour 那本著作的后续之作，Yaghmour 的那本书本身也是极好的资料来源，我强烈建议你找一本来读。

本系列的第 2 本（将于不久后出版）将会对 Android 讨论得更深，而且会把视线转向 Android 框架服务（framework）的结构——这显然对开发者更有吸引力：通过使用 Java 层上各种丰富的框架，开发者可以拥有把输入设备、传感器、图形图像之类的东西抽象化的强大能力。当然这一抽象化的能力也并非是没有代价的——复杂性隐藏在“水面之下”，只是大多数开发者对此安之若素（更有甚者还满足于这一状态）罢了。不过知识是力量的源泉，深入熟悉各类框架（及其底层实现机制）对于任何想要进行底层开发或者在性能调优、支持更多的硬件、安全研究等方面有所建树的人士都是至关重要的。

Android 是一个不断飞速更新的系统。在本书开始编写时，最新版的 Android 系统还是 KitKat，然后（尽管中间有过几次跳票）最新版就变成 Lollipop 了。而且这一趋势还在不断加快——由于 Lollipop 版被发现有不少 Bug，谷歌又宣布将要推出 Android Marshmallow 版。不过，截至本书付梓时，Lollipop 版显示出了它稳定的一面——所以我也可以骄傲地说，这本书已经反映了最新版……好吧，是截至出版之日。幸运的是，借本书自媒体出版的东风，我可以不断地紧跟 Android 系统的更新而修订本书的内容，读者现在看到的这一版已经更新到 Marshmallow Preview Release 1 版（2015 年 6 月）了[1]。

我还试图从我的上一本书 *Mac OS X and iOS Internals* 中吸取一点“经验教训”。我收到的读者对那本书的主要批评之一是：那本书太技术化了，充斥着大量源码，非开发人员身份的读者读起来实在是太累了。我个人的信条是“读一下源码吧，淡定些！”——因为源码不像自然语言，（几乎）是不会有歧义的，因此也是用来描述系统的正确方式。虽然我还是坚持我的想法不动摇，但在这本书中，我还是在不牺牲细节信息的前提下，尽可能多地改用图表的形式来表达意思。[我把这一做法也用在了 *Mac OS X and iOS Internals* 一书的第 2 版上（这一版将于 2016 年下半年出版）——这倒也不完全是因为我想通了，心甘情愿地这么做，还有一个重要因素也在促使我这么做，那就是在那本书里更加深入地探讨了 Mac OS X 和 iOS 系统更底层、更隐秘的部

---

1 原文如此，就是苦了我这个译者，翻译的过程中不断地修订，原文都在不断地改，更何况译文……，简直是一个不断返工的死循环“哭”。截至本书中文版出版时，最新版为 2016 年 11 月底更新的版本，最新的内容更新至 Android Nougat 版。——译者注

分——这些东西可是没有源码的……]。

本书也十分强调动手实践，我从我们的 Android 培训课程里抽取了一些动手练习，并把它们改编成了书中的实验。如果你想要对相关章节讨论的主题有深入了解的话，这些实验对你来说无疑是极为重要的。Android 是 UNIX（实际上是 Linux）的一种衍生品，而学习 UNIX 的唯一正确方式是用我们的手指，而不是用我们的眼睛或耳朵。在这些实验里演示了 Android 命令行接口（CLI，command-line-interface）中的一些非常有用的命令，以及深入了解操作系统内部结构的技术。更进一步说，有些实验在不同的 Android 环境下会产生不同的输出结果——这也使它们非常值得你在自己的手机/平板电脑上亲手做一遍，以体验不同厂商或操作系统版本在架构和实现上的不同之处。

全书内容鸟瞰

本书的内容编排，既可以让你能按部就班地逐页阅读，也能让你随便翻开一页就能读下去。书中的每一章都是独立成章的，在你阅读本书的电子版时，文中所提及的相关主题都是以超链接的形式给出的——直接点击它，你就可以跳转到相关章节进一步阅读，但是对于纸质版的读者，我就只能给出相应的章节编号（引用书中的内容时）或者 URL（引用其他资料时）了。我也会在相应的地方附上所引用的 AOSP 文件的路径（尽管为了节省空间，是以缩略的形式给出的），要不然本书的主要用途就变成“防身”了……

第 1 章介绍了 Android 操作系统：介绍了它各个不同版本的演化史（从 Froyo 版开始，这是你目前在市场上能找到的最老版本的 Android 系统了，一直到 Lollipop 版[1]）。同时，在这一章里（从较高的视角）也通过逐一比较 Android 软件栈中的各个层（layout），阐述了 Android 的体系架构以及它的 Linux 基础。紧接其后，这一章还介绍了 Android 的衍生产品——既包括谷歌自己的，也包括其他厂商的（比如亚马逊的 FireOS）。最后，这一章将对 Android 未来发展方向的设想和思考作为整章的小结。

从第 2 章开始，我们开始深入探索各种技术细节——第 2 章的主题是 Android 的分区和文件系统。我们先讨论 Android 使用的分区架构（不幸的是，各家厂商远远没有对此达成一致），以及文件系统——EXT4 和 F2FS。然后，我们将探究文件系统中存放的内容——如果你想要知道某个特定的系统目录或文件中存放的是什么数据，这将是非常有用的。此外，本章还会涉及一些内置应用的数据存放目录，如果你对电子取证感兴趣，这些内容无疑也是非常重要的。在这一章中还会讨论 Android 受保护的文件系统（OBB 和 ASEC）——尽管在系统被 root 之后，

1　事实上 2016 年 12 月的更新版中已经介绍到 Nuogat 版了。——译者注

这些保护措施就会失效。最后，我们还会阐述 Linux 伪文件系统（cgroupfs, debugfs, procfs, sysfs 等）在系统中扮演的角色。

第 3 章是在前一章的基础上展开讨论的——因为涉及分区。它解释了在 Android 系统启动过程中，各个分区所起的作用。我们先会讨论 Android 的启动镜像（尽管有时会有些不正确地把它称为 ROM），以及怎样把它刷到设备的各个启动分区里去。Android 默认使用的 Boot Loader 也会予以阐述（本书官网上还有一篇从更加技术的角度开展讨论的进阶阅读文章），以及启动镜像的其他一些组件［内核（kernel）、设备树（device tree）和 initramfs］也会被详细讨论。本章中的相关实验还演示了如何把这些组件从启动镜像中解出来，修改其中的内容，然后再把它们重新打包回去——制作一个你自己的刷机包（当然安装这种刷机包的前提是：在你的移动设备上 Boot Loader 已经解锁了）。此外，在这一章中还讨论了通过无线网络发送更新镜像进行（OTA）升级，以及设备备份、重置和关机的操作过程。

第 4 章专门讨论一个进程/init。这个进程和它在 UNIX 系统中的同名进程一样，是负责在用户态中启动系统的。我们会详细解释启动的过程，并解释/init.rc 文件中使用的语法。/init 的其他一些作用，比如维护系统属性和监视硬件改变（以 ueventd 进程的身份），也会详细地加以讨论。

在第 5 章中讨论的是原生服务，也就是列在/init.rc 文件中的，由/init 进程启动的 Linux 二进制可执行文件（与之相对应的是 Dalvik 级的框架服务，这些服务是以 system_server 进程中的线程的形式被加载起来的，我们会在第 2 本中讨论这些框架服务）。在这一章中逐个详细介绍了你可以在自己的移动设备上找到的每一个守护进程——说实话，还真不少。

在第 6 章中简略介绍了 Android 框架服务的大致架构，解释了 servicemanager 和 system_server 进程在其中所扮演的角色——这两个进程共同构成了其余所有构建在其上的 Android 框架服务的基石。Binder 也是这一章里的重头戏，我们会简略地对它进行一番描述，但是大部分细节信息还是要留待第 2 本讨论补充。我希望这一解释足以让你能更深一步地理解 Android 进程间通信和远程过程调用的内部工作机制。

第 7 章以 Linux 的视角来看待 Android，也就是通过/proc 伪文件系统以及使用 Linux 系统中的工具，观察 Android 系统中的进程及应用。这一章还有一个“一箭双雕”的作用——你可以把这一章讨论的绝大多数工具，用在你自己的 Linux 系统原生代码的调试工作中。

作为本书的最后一章，第 8 章是专门用来讨论安全的。这一章在本书的官网上有预览版（只不过在预览版中的编号中，它是第 21 章——当时我曾经天真地认为可以在一本书中把所有的问题都讲清楚），在这一章里将逐一详细讨论 Android 的所有安全特性——既包括 Linux 层上的，也包括 Dalvik 虚拟机层上的。同时，在这一章中还有一个小节专门来讲述 Android 设备的 root

问题——既讨论了“被厂商认可的”在设备启动时 root 的方式，也讨论了那些通过安全漏洞 root 设备的方法。

本书使用的排版约定

本书采用如下排版约定：

- 以 filename 这种格式表示文件名。
- 命令、系统调用名称以及框架类名都是以 `command(1)`、`systemCall(2)` 以及 `classes` 这种格式表示的。命令和系统调用名称后面跟的数字是指：在使用 Linux 的 man 命令打开的手册中，该命令或系统调用所在的章节编号。

此外，本书还有许多插图、代码清单和输出结果。插图是由系统组件或消息传递流程组成的图片，相对于输出结果，代码清单中给出的一般是内容固定的文件中的内容，而输出结果中给出的则是一连串命令的执行结果——它通常是某个实验的一部分。我制作输出结果的目的是：显示各条命令的执行顺序及其用法，所以输出结果中一般都是带注释的（如输出结果 0-1 所示）。

```
# Comment, explaining what's being done
user@hostname   (directory)  User input
Output...
Output..  # Annotation, explaining output
Output..
```

输出结果 0-1　一个输出结果样例

请注意上面这个输出结果中的细节——用户名（上面这个输出结果中的 user）（以及命令行提示符是$还是#）能够告诉你，这条命令是在 shell 中就能执行，还是必须要有 root 权限才能执行的。主机名（上面这个输出结果中的 hostname）则可以告诉你这条命令是在哪台设备上执行的——如果它是 generic，表示是在一台模拟器中；如果是 flounder，表示是在一台 Nexus 9(L) 中；如果是其他移动设备的名称（s3、s4、kindle 或 Nexus 5 之类的），则表示是在一台对应的移动设备中；如果它是 Forge，则表示是在作者自己的 Linux 计算机上运行的。我尽量避免在书中出现大段的代码（至少在本书中是这样的），在迫不得已的情况下，我也尽量只给出最关键的代码，此外，我也会在代码中加上帮助你理解代码的注释。代码的字体颜色也调整为能够同时兼顾彩色（如果你读的是 PDF 版）和黑白（如果你阅读的是纸质版）两种打印方式的颜色。