TURING 图灵程序设计丛书

Android 系统服务开发

【韩】金大佑 朴宰永 文炳元 著　邸春红 译　陈家林 审

Android Hardware Service

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

Android 系统服务开发

【韩】金大佑 朴宰永 文炳元 著　邸春红 译　陈家林 审

Android
Hardware
Service

人民邮电出版社
北京

图书在版编目（CIP）数据

Android系统服务开发 / （韩）金大佑，（韩）朴宰永，（韩）文炳元著 ；邸春红译. -- 北京 ： 人民邮电出版社，2015.2
（图灵程序设计丛书）
ISBN 978-7-115-37554-4

Ⅰ. ①A… Ⅱ. ①金… ②朴… ③文… ④邸… 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(2014)第265778号

内 容 提 要

本书分析了 Android 提供的硬件控制机制。编写团队目前均从事相关工作，直接对平台源代码及日志进行分析及测试，介绍了目前尚未普及的 Android 平台的硬件控制基本原理及实际框架的操作。

基本功扎实的开发人员可以主导顺应智能手机时代发展的“Smart”开发。通过本书可以掌握 Android 平台的操作原理，并将其直接应用于开发业务。

◆ 著 [韩] 金大佑 朴宰永 文炳元
译 邸春红
审 陈家林
责任编辑 傅志红
执行编辑 陈 曦
责任印制 杨林杰

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京缤索印刷有限公司印刷

◆ 开本：787×1092 1/16
印张：26.5
字数：715千字 2015年2月第1版
印数：1－4 000册 2015年2月北京第1次印刷

著作权合同登记号 图字：01-2013-8985号

定价：139.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第 0021 号

版权声明

作 者 序

金大佑

长达两年半的写作终于结束了，真让我耗尽心力。公司、家庭、写作齐头并进，经常让我觉得时间不够用。体力和健康状况大不如前，这也牵制了我的写作计划。我有好几十次想过就此放弃，有好几百次问过自己："为什么要自讨苦吃？为什么要做这些？"频繁的加班，甚至周末也要加班，这让我很难保证写作时间。孩子出生后，时间就更少了。尽管这样，我仍旧坚持每天更新稿件，心想着哪怕每天只写一行也好。就凭着这种不放弃的态度，本书最终得以面世。如果中途放弃的话，我一定会抱憾终身。

Android 的版本不断更新，我最初接触 Android 的时候，是 Android Cupcake（1.5）版本，现在已经升级到了 Android Jelly Bean Plus（4.2）。在此期间，Android 取得了令人瞩目的发展，是目前拥有最大用户群的智能手机 OS。Android 发布之初很难找到相关资料，而且在实际应用领域，使用 Android 的人也只是少数。我本人也曾经通过 Google 搜索 Android 相关资料，但找到的信息只停留在基础理论水平。当然，现在也经常难以找到真正需要的内容，不得不自己解决问题。因此，我开始整理开发中所需资料，希望有一天能够以某种形式公开这些资料，以帮助那些因为资料不足而伤脑筋的开发人员。一个偶然的机会，我在网络上看到了招募作者的网络招聘信息，于是决定参与其中，与团队中给我很大影响的朴宰永负责人，以及对 Android 平台有很大兴趣的文炳元先生合作，最终完成了这本书。

不知道大家是否听说过马尔科姆·格拉德威尔（Malcolm Gladwell）在《异数》（*Outsider*）一书中介绍的"一万小时定律"？"一万小时定律"是指，无论任何领域，如果想要成为一名专家，那么至少要投入一万小时。这一万小时是由什么组成的呢？只要工作时间合计超过一万小时，就可以成为该领域的专家吗？如果以每周工作 40 小时为基准，工作一年的话有 2000 小时，这样一来，如果一份工作做 5 年，那么就在该领域有了一万小时的经验，应该就可以说是一位专家了。然而我们面对的现实情况却是，从事开发工作 5 年的开发人员有时候做调试都觉得吃力，更别说做系统设计了。是的，"一万小时定律"不是单纯的工作时间总和，而是为了提高自己的实力所投资的时间总和。我想，这本书也是我提高自身实力过程中的产物。在边写书边思考如何向读者进行说明、如何更好地帮助读者理解的过程中，我也学到了很多。这种思考成为非常有用的知识及经验，可以在实际开发中解决很多问题。

与 Android 初创期相比，现在的 Android 开发人员有了大幅增加，然而这样就可以认为进入 Android 世界的门槛降低了吗？开源的优点就好比开卷考试的优点，开卷考试虽然为所有人都提供了相同的机会，但是在限定时间内进行考试而得到的结果却有所不同，那些对基础知识掌握透彻的人反而能够得到更好的成绩。同样，开源的优点也使得很多人可以接触到 Android 平台的核心，但是这无法降低进入 Android 世界的门槛。为什么一线的开发人员仍然会觉得 Android 开发难度很高呢？是因为 Android 的发布速度慢于其他智能手机 OS 吗？或者是因为 Android 还

不够成熟？还是因为 MSDN 等强大的开发人员帮助手册还不够完备？

当然，上面所列的原因都是影响 Android 开发的要素，但我个人认为，最主要的原因是恶劣的开发环境。极短的项目周期、接连不断的加班、依照 Man-Month 投入的人力、不完善的开发人员教育系统等，这些恶劣的开发环境消耗了开发人员的好奇心，反而使 Android 变为令开发人员感到痛苦的对象。在这种开发环境下，与其从根本上找到避免出现 Bug 的对策，不如快速修复 Bug，解决 Bug 跟踪系统的问题。我本人开始进行 Android 开发后不久，有一次阅读 Android 方面的参考书籍时，一个同事问了我这样一个问题："连开发的时间都不够用，哪儿还有时间看书啊？"我当时是这样回答的："如果想要更好地掌握开发技术，就一定要学习。如果只注重开发，那么就没有时间学习了。"我一直相信，一名透彻地理解了 Android 的工程师可以以一当十。希望有一天，人们能够在开发一线进行 smart work，获得充分的休息；开发人员可以完全发挥出自己的能力，不是用"身体"，而是用"头脑"去开发。

本书的出版得到了很多人的帮助。首先要感谢 Developerware 出版社的郑权社长和金泰然先生，没有他们就不会有这本书。感谢郑权社长的信任和耐心，感谢泰然先生的指点。另外，和我一同编写本书的朴宰永先生、文炳元先生这段时间也非常辛苦。感谢教我硬件知识的尹玄高级研究员、出色的首席 Leader 金承民先生、认真组织测试的金东旭高级研究员和李成信研究员，以及曾与我参加同一项目的李正浩研究员、朴智英研究员、Call/SMS/MMS 各位组员，还有在开发一线与我同甘共苦的尹尚浩高级研究员、与我同年进入公司的刘贤先生。另外，还要感谢百忙之中依然给我鼓励的瑞萨移动公司的金振形本部长、柳仁焕部长、李正久部长、汝成九部长、李江勋副部长、李焕承副部长、宋英伦副部长、李尚京副部长、卓泰勋副部长、金东瑾课长。

如果没有我亲爱的妻子金耀百合和可爱的儿子金河林的体贴，我也无法完成这本书。之前的很多个周末我都因为要编写本书而无法与他们一起度过，现在不用再思考修改稿件了，我想把之前欠他们的时间都补回来。另外，我还要借此机会表达我对家人的爱。

虽然第一次写作结束了，但是不知为何，我还是觉得留下了很多遗憾。在下一次的写作中，我想进一步贴近开发人员，围绕更加轻松的主题进行介绍。我相信下一部书一定会在更短的时间内完成。如果把开发当做工作，会觉得有很大的难度；而如果当做一种兴趣，那么就不会有比这更让人享受的事了。

朴宰永

我还记得 2010 年春天，那时，非智能手机已开发了 7 年。

在 iPhone 的狂潮中，LG、三星等世界级的手机制造商都为追赶智能手机这一潮流而孤军奋战，公司内部也动用了所有力量来开发 Android 手机。那时，我也厌倦了不断重复开发非智能手机，正好听说公司内部招募人员开发基于 Android 的平板电脑，就毫不犹豫地报了名。

开发平板电脑的过程中，我对 Android 平台产生了很大兴趣，想要进一步对其进行研究。当时我最喜欢并最看重的后辈金大佑高级研究员给了我一个很好的建议："您要不要和我一起写本书？"我由此开始了写作的旅程。

然而，仅凭非智能手机的经验和自信是无法让 Android 平台轻易向我打开秘密之门的。从内核到框架都是我未曾在原有非智能手机领域接触过的操作系统与编程语言，这使得 Android 平台更难理解，也使我经历多次历练，甚至怀疑自己连一行都写不出来。

随着时光的流逝，我的焦躁感及紧迫感日益强烈，要从哪里开始、要做什么，就好像盲人摸象一样。然而，所有的这一切都通过一个众所周知的方式解决了："勤能补拙"。

虽然每天的工作十分繁忙，但是只要时间允许，我就会对照代码进行测试与调试。我们每周在钟路的学习室编写一章的内容，或者用一整天去完成一幅图。如果由于工作繁忙而导致写作完全没有进展，那么会利用个人休假时间在学校图书馆专心写作，像明天就要参加期中考试的学生一样。就这样过了一年，曾经一行都写不出来的笔记本上，不知不觉已经写满了 100 页、200 页。阳光开始照射到曾经一片迷茫的 Android 世界。

然而，在写作速度有所提高、完成量也越来越多的时候，Android 更新了版本。开始写作时是 GB 版，中间变为 ICS 版，2012 年底变为了 JB 版。虽然从结构层面来说没有太大变化，但如果想对每一行代码的变量都进行准确说明，修改量是非常大的。

就这样，我们在 3 年时间里进行了数十次的检查与修改，终于完成了硬件服务开发这本书。虽然现在很遗憾，没有介绍 JB MR 的内容，但是很多设备都安装了 JB 版本，由此看来，本书还是非常有价值的。

本书的出版得到了很多人的帮助。

感谢金泰然先生，他像照亮黑夜的灯塔一样，3 年间为我们指明写作的目标和方向，是我们的榜样。和我一同编写本书的金大佑高级研究员、文炳元研究员在这段时间也十分辛苦。

感谢朴孝善先生，他本人工作非常繁忙，但依然利用自己的周末时间最早审读了原稿。感谢先后两次在钟路学习室用一整天时间审读本书的金成振主任、李万秀研究员、全盼基研究员。还要感谢校订终稿的文化日报编辑部的全智勉记者，以及将全智勉记者介绍给我们的首尔医院江南中心的全赫泰教授。另外，还要感谢在 3 年时间里默默等待的 Developerware 出版社的郑权社长。

最后要感谢我的灵魂伴侣、永远的爱人崔恩静，这 3 年时间里我每天加班到很晚，周末也要学习，她只能独自陪着范周，十分辛苦。

文炳元

3 年前，和我同时进入公司的泰然哥提议写本书。他当时正在写另一本著作，我经常能听到他聊起写书的情况，因此也对写作产生了一定兴趣。于是我马上接受了建议，并且拜见了将要一同写作的金大佑先生和朴宰永先生。经过 3 年的努力，这本书终于呈现在大家面前。

我们 3 名作者在技术上还有很多不成熟之处，多次的错误尝试以及其他问题都让我们觉得疲惫不堪，不过最终还是完成了本书。编写本书的过程中，我们之间出现过各种各样的争论。

其主题通常只有一个："这种内容能够让读者轻松理解并有所帮助吗？"

我们删除了很多不符合这一标准的内容，然后重新编写更为丰富的内容。就像最近苹果广告中说的："1 个 Yes 需要 1000 个 No。"编写一本书也需要无数次的自我批评与反省。

著名的软件领域专家、美国计算机名人堂代表人物杰拉尔德·温伯格在其著作《成为技术领导者》一书中介绍了处理难题时使用的特殊方法，该方法如下。

"如果想学习某个主题，可以先针对这个主题准备授课，向其他人讲授这一课程，并且充分地进行补充学习，然后围绕这一主题写书。"

我是最近才看到杰拉尔德·温伯格的这句话的，但是我在自己学习的过程中也使用了与此相似的方式。在公司准备授课并进行陈述，然后学习并编写成书。这一过程中可以学到更多内容。

本书中的内容同样如此，我们将学到的内容编写成书，将自己了解的内容以最易于理解的方式整理出来。相信各位阅读本书后一定会觉得有所获益。

感谢

我想感谢编写本书过程中帮助过我的各位朋友。感谢金大佑先生在漫长的编写时间里一直出色地承担着编辑负责人及作者任务、朴宰永先生像大哥一样一直是我的榜样、郑权社长对原稿进行编辑并一同参与完成写作。还要感谢我 20 多年的朋友智勋仔细地检查原稿，以及泰然哥给了我这个写作的机会。另外，我还要感谢在我进入 LG 电子后手把手悉心指导我的东万哥、成勋哥、朱玉高级研究员、世赫主任、形硕高级研究员，以及我所在团队的所有同伴。正是有了大家的帮助，我才能学到丰富的知识并快速成长。最后，我还要感谢在长时间的写作过程中一直鼓励我、最支持我的（甚至还帮我校订书稿！）亲爱的妻子美丽，以及她肚子里的儿子星星，我爱你们。


思静、范周
我爱你们


美丽
&星星
ME?... WE!

目　录

01

第 1 章　Android 系统服务开发

Android Hardware Service

作为一款移动设备软件平台，Android 支持各种硬件，并为硬件控制提供高水平的 API。

本章介绍 Android 系统的基本结构，以及 Android 提供的系统服务开发类型和硬件控制方法。除此之外，还对 Android 提供的进程间通信（Inter-Process Communication，IPC）和线程间通信（Inter-Thread Communication，ITC）方法进行讲解。Android 的守护进程使用 Linux 提供的默认 IPC 方法 UNIX 套接字进行 IPC，Android 的 Java 进程使用已实现的消息通信方法进行 ITC。

Android 是由 Google 开发的基于 Linux 的移动设备平台，提供 OS、中间件、用户界面（UI）以及 API 等。

Android 默认移植于 ARM 架构，也可以移植于 MIPS 及 x86 架构。作为核心的本地库由 C 语言及 C++ 语言编写而成，除此之外的框架层、UI 及应用程序由 Java 编写而成。其优势在于，如果选择 Java 作为编程语言，则无需创建架构应用程序，使用一个应用程序即可跨平台运行。编写的 Java 位元组码不是通过 Sun Jana VM 执行的，而通过 Google 自己开发的 Dalvik VM 执行。

Android 提供 SDK（Software Development Kit，软件开发套件），提供各种用于开发用户应用程序的 API 及开发工具（调试器和模拟器等）。Android 操作系统使用 Linux 内核，包含了各种 Android 系统构成要素中使用的 C 语言及 C++ 语言库。Android 还提供 NDK（Native Development Kit，本地开发套件），使开发人员可以令应用程序调用由 C 语言和 C++ 语言编写的本地模块。

1.1　Android 系统服务开发简介

Android 已植入移动电话、平板电脑、导航、相机、电视机及其他生活家电等各种移动设备。Android 遵循开源协议 Apache v 2 许可，向所有芯片供应商敞开大门。用作 Android OS 的 Linux 内核遵循 GPL 协议，承担向所有开发商公开 Linux 内核源码的义务；然而，遵循 Apache v 2 许可的 Android 平台代码则不需要承担公开开发代码的义务。由于使用 Android 不需要支付协议的费用，因此，芯片供应商与手机制造商可以毫无限制地生产 Android 设备。另外，各制造商可以根据 Android 的新功能接入顺应最新趋势的硬件技术，所以能够对性能进行优化也是其一大优点。

作为 Android OS 的 Linux 内核支持各种不同的硬件，因此与其他平台相比，可以快速将最新硬件及各种硬件架构商业化，推出新的产品。Android 为原 Linux 内核不支持的移动设备添加了电源管理、LMK（Low Memory Killer，低内存管理）、Alarm、内核调试器 / 记录器、Binder 等。另外，通过其极具灵活性的结构，不仅可以制造 PMP 等单一设备，还可以制造高端旗舰级手机设备。Android 对最新硬件的支持为开发人员开发各种 UX 及最新功能的应用程序提供了基础。

Android 每次发布新版 API 的同时，都会提供支持最新硬件的 API，这有助于反映最新硬件技术的新概念应用程序的出现，满足了用户的需要。本章将介绍理解 Android 硬件服务开发必备背景知识。

1.1.1　Android 的出现

人们曾经尝试利用嵌入式 Linux 或 LIMO 等 Linux 平台开发智能手机，最终都失败了。失败的原因有很多，但主要的原因是缺乏共同开发环境，以及未能为开发人员创建一个可持续发展的生态系统平台。不过，Android 于 2007 年首次登场后，吸引了大量开发人员，并形成了 Google Play Store 等 Android App 商店。另外，很多制造商都选择 Android，这样就成功构建了一个良性循环的结构，以围绕 Android 的生态系统为基础，在移动 OS 竞争中占据优势地位。

图 1-1 整理出了 Android 的演进。Android 始于 2003 年由安迪・鲁宾（Andy Rubin）、里奇・米纳尔（Rich Miner）、尼克・西尔斯（Nick Sears）及克里斯・怀特（Chris White）创立的 Android Inc.。Google 于 2005 年 7 月收购了这间位于美国加利福尼亚的小软件公司。此后，在 2007 年 11 月，为了开发能够应用于移动设备的标准型 OS，包括 Google 在内的各国厂商组

建了开放手机联盟（OHA）。

2008 年 10 月 22 日，由 HTC 负责生产的世界上第一款 Android 手机 G1 发布了。以 HTC “G1”为起点不断发展，很多手机制造商开始在自己生产的设备上搭载 Android。2008 年 9 月，Android 1.0 版本上市。*

Google 依照开源协议 Apache v2 许可发布所有 Android 源码，企业或用户可以独立开发并搭载 Android 程序。此外还提供了可以进行应用程序交易的 Google Play，与此同时，各制造商及通信公司还共同运营应用程序市场。市场中提供的应用程序分为付费及免费两类。

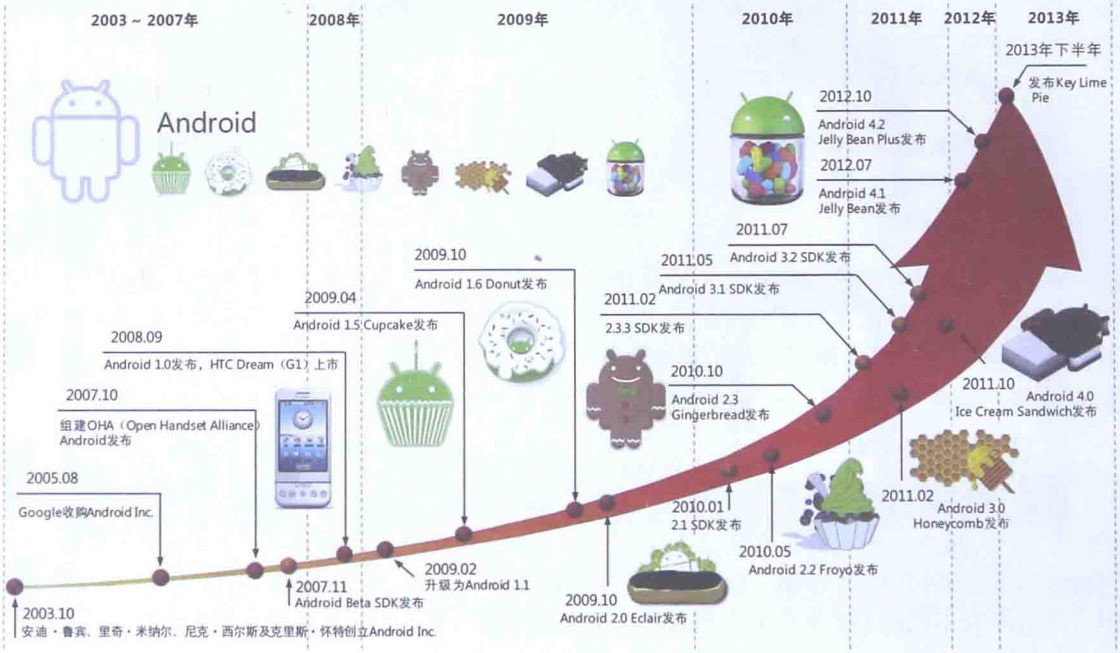

图 1-1 Android 的主要沿革

Android 在基于原有 Linux 内核运行的内嵌式 Linux 及 LIMO 上失败了，但在智能设备中成为了非常具有优势的平台。那么，其成功主要取决于哪些原因呢？

1. 构建面向应用程序的生态系统

原有的非智能手机使用的不是通用的 OS，而是 RTOS，手机制造商的 UI 平台各不相同，无法激活应用程序市场。嵌入式 Linux 将 Linux 内核用作 OS，但没能实现共同 UI 平台，故无法提供适合用户使用的应用程序。为了解决这些问题，Android 发布了在 Linux 内核中运行的 Java 平台，以及可以用于制作应用程序的 SDK。这令编程人员可以利用 SDK API 开发 Android 应用程序，使之能够在所有使用 Android 平台的设备上运行。

Android 设备需要通过 CTS（Compatibility Test Suite，兼容测试）这一认证过程才能发布。通过 CTS 认证的 Android 设备保障了与 Google Palyer 商店中应用程序的兼容性，完善了在有些设备上无法运行的问题。

* 在此之后，Google 已于 2013 年 9 月和 2014 年 10 月先后发布了 Android 4.4（kitkat）和 Android 5.0（Lollipop）。——编者注

2. Google 的强大支持

Android 出现之前，LIMO 被视为下一代平台而备受瞩目。然而 LIMO 没能满足大家的期待，最终从市场上消失了。LIMO 是基于 Linux 内核创建的平台，有各类厂商参与其中，但没有理顺各厂商之间的利害关系。也没有人组织进行软件开发，无法快速应对市场需要，最终没能获得移动 OS 的地位而搁浅。Bada OS 作为一种新的移动 OS 出现，但也没能取得显著成果，与 TIZEN 合并后就从历史舞台上消失了。为了确定一个新的平台，需要有强大的韧性和资本，还需要改善向下兼容性和平台，并不断根据定位对平台升级。

Google 于 2005 年收购了当时的安卓公司，并不断升级版本、改进功能、构建生态系统，最终创建出了现在我们所认识的安卓。如果没有 Google 强大的领导力，安卓也不会取得如此大的发展。

3. 开源

平台最强大的支持者就是开发人员。Android 从一开始就是开源的，得到了开源社区的支持，得以迅速发展。开发人员的自发性是平台获得成功的最大资本，来自于开发人员对新平台的好奇，以及对新机会的期待。

Android 通过开源使很多芯片供应商和 OEM 企业参与其中，对选作 OS 的 Linux 内核未提供的标准 HAL 进行定义并发布，使芯片供应商与 OEM 企业可以利用共同接口控制硬件进行开发，并且可以方便地使用 Linux 支持的各种设备。另外，Android 得到开源社区的支持，除 Google 外，很多其他开源社区的开发人员也可以报告问题或提交修订。

4. 提供共同的 UX

Android 之前的嵌入式 Linux 等平台没能成功提供共同的 UX。虽然各嵌入式 Linux 也提供 UX 平台，但是由于不同的制造商会选择不同的 UX（QT、GTK 及制造商的 Custom GUI 等），因此无法提供共同的 UX 环境，必须制作专用应用程序。因此，虽然是相同的嵌入式 Linux 平台，但应用程序在不同的 UX 平台上不能兼容，最终成为嵌入式 Linux 进行普及的绊脚石。

5. 提供标准化的开发环境

一个新平台出现后，向开发人员普及时最重要的因素之一就是支持集成开发环境（Integrated Development Environment，IDE）。MS 为了使开发更加方便、高效，提供了 Visual Studio 等 IDE 工具，吸引了大量 Windows 开发人员。

Android 出现前的嵌入式 Linux 环境中没有提供可用的 IDE，必须利用 vim 编辑源码，在命令窗口利用 make 和 gcc 创建源码。Android 提供了 Eclipse 等开源 IDE，以及用于创建源码的标准化环境，使开发人员工作起来更方便高效。如果没有提供标准化的源码创建环境，就无法统一每个制造商创建源码的方法，每当开发环境有所变化，就要学习相应的源码创建方法。

开发人员调试一个新程序的时间并不少于设计和编码的时间。Android 提供了 adb（Android Debugging Bridge）和 ddms（Dalvik Debug Monitor Server）等标准化调试工具，改善了应用程序调试环境。过去，不同的制造商或解决方案供应商使用不同的调试方法，所以需要很多时间学习开发环境。然而 Android 提供了标准调试方法，自此就不必再学习不同开发环境下的调试方法了。