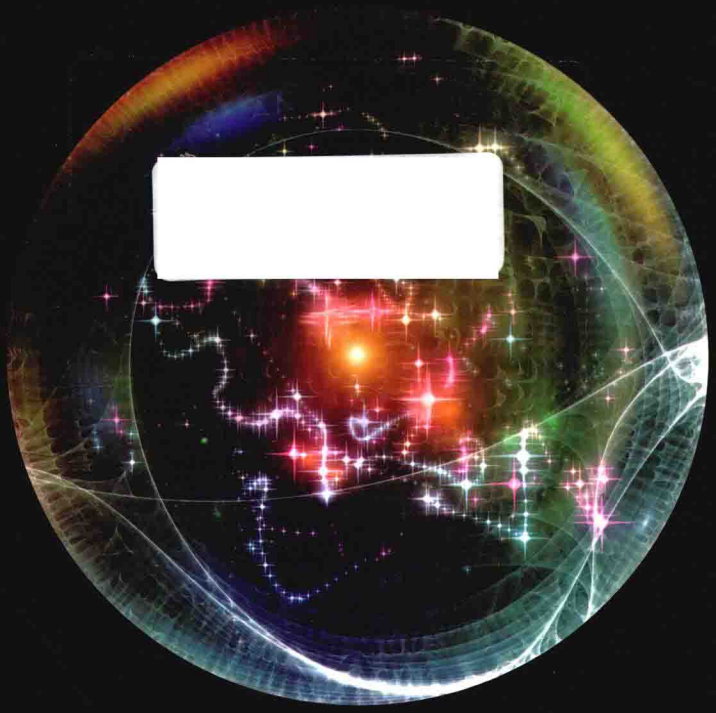
- 全面剖析进程 / 线程、内存管理、Binder 机制、GUI 显示系统、多媒体管理、输入系统、Android Dalvik/Art 虚拟机、Android 的安全机制、Gradle 自动化构建工具等核心知识在 Android 系统中的设计思想
- 通过大量图片与实例来引导读者学习，以求尽量在源码分析外，为读者提供更易于理解的思维路径
- 由浅入深，由总体框架再到细节实现，帮助读者彻底理解 Android 内核的实现原理

异步图书
www.epubit.com.cn

深入理解 Android 内核设计思想

第 2 版 | 上册

林学森 ◆ 著



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

深入理解 Android内核设计思想

第2版 | 上册

林学森 ◆ 著

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

深入理解Android内核设计思想 : 全2册 / 林学森著
. -- 2版. -- 北京 : 人民邮电出版社, 2017.7
ISBN 978-7-115-45263-4

Ⅰ. ①深… Ⅱ. ①林… 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105893号

内 容 提 要

全书从操作系统的基础知识入手，全面剖析进程/线程、内存管理、Binder 机制、GUI 显示系统、多媒体管理、输入系统、虚拟机等核心技术在 Android 中的实现原理。书中讲述的知识点大部分来源于工程项目研发，因而具有较强的实用性，希望可以让读者“知其然，更知其所以然”。本书分为编译篇、系统原理篇、应用原理篇、系统工具篇，共 4 篇 25 章，基本涵盖了参与 Android 开发所需具备的知识，并通过大量图片与实例来引导读者学习，以求尽量在源码分析外为读者提供更易于理解的思维方式。

本书既适合 Android 系统工程师，也适合于应用开发工程师来阅读，从而提升 Android 开发能力。读者可以在本书潜移默化的学习过程中更深刻地理解 Android 系统，并将所学知识自然地应用到实际开发难题的解决中。

◆ 著　　林学森
责任编辑　张　涛
责任印制　焦志炜

◆ 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
三河市海波印务有限公司印刷

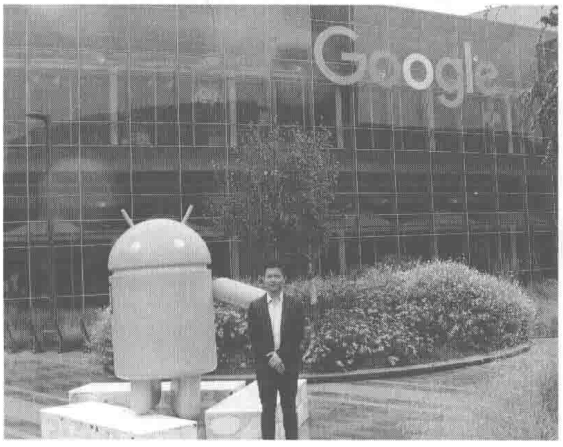
◆ 开本：787×1092　1/16
印张：63.75
字数：1681 千字　　2017 年 7 月第 2 版
印数：11 001 – 14 000 册　　2017 年 7 月河北第 1 次印刷

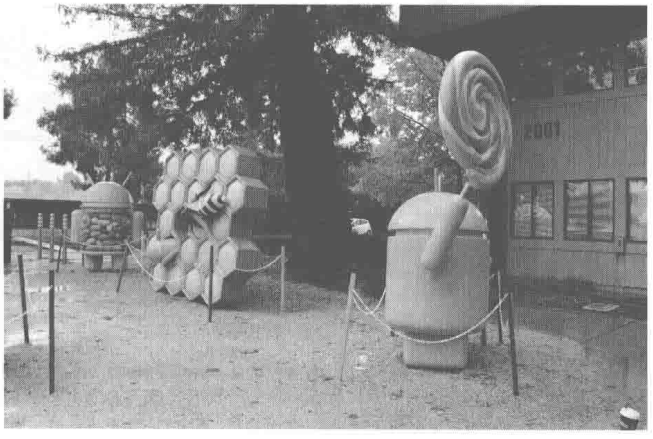
定价：158.00 元（上、下册）

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

第 2 版前言

Android 系统的诞生地——美国硅谷。





Google 大楼前摆放着 Android 的最新版本雕塑，历史版本则被放置在 Android Statues Park 中

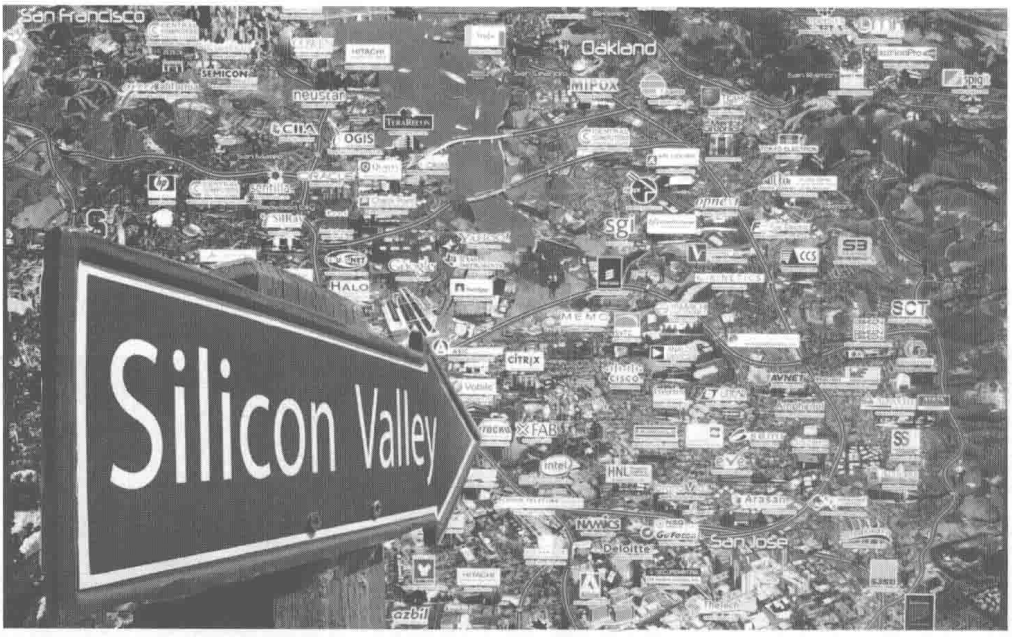
写第 2 版前言时，笔者刚好在美国加州硅谷等地公事出差访问。其间我一直在思考的问题是，美国硅谷（Silicon Valley）在近几十年时间里长盛不衰的原因是什么？技术的浪潮总是一波接着一波的，谁又会在不远的将来接替 Google 的 Android 系统，在操作系统领域成为下一轮的弄潮儿？我们又应该如何应对这种“长江后浪推前浪”的必然更迭呢？

从历史的长河来看，新技术、新事物的诞生往往和当时的大背景有着不可分割的关系。如果我们追溯硅谷的发展史，会发现其实它相对于美国很多传统地区来说还是非常年轻的。“硅谷”这个词是在 1971 年的“Silicon Valley in the USA”系列报导文章中才首次出现的。20 世纪四五十年代开始，硅谷就像一匹脱了缰的野马一般，“一发不可收拾”。从早期的 Hewlett-Packard 公司，到仙童、AMD、Intel 以及后来的 Apple、Yahoo!等众多世界一流企业，硅谷牢牢把握住了科技界的几次大变革，成功汇集了美国 90%以上的半导体产业，逐步呈现出“生生不息”的景象。

但为什么是硅谷，而不是美国其他地区成为高科技行业的“发动机”呢？

古语有云，“天时、地利、人和，三者不得，虽胜有殃”。

现在我们回过头来看这段历史，应该说硅谷早期的发展和当时的世界大环境有很大关系——更确切地说，正是美国国防工业的发展诉求，才给了硅谷创业初期的“第一桶金”。只有“先活下来，才有可能走得更远”。而接下来社会对半导体工业需求的爆炸式增长，同样让硅谷占据了“天时”的优势，再接再厉最终走上良性循环。

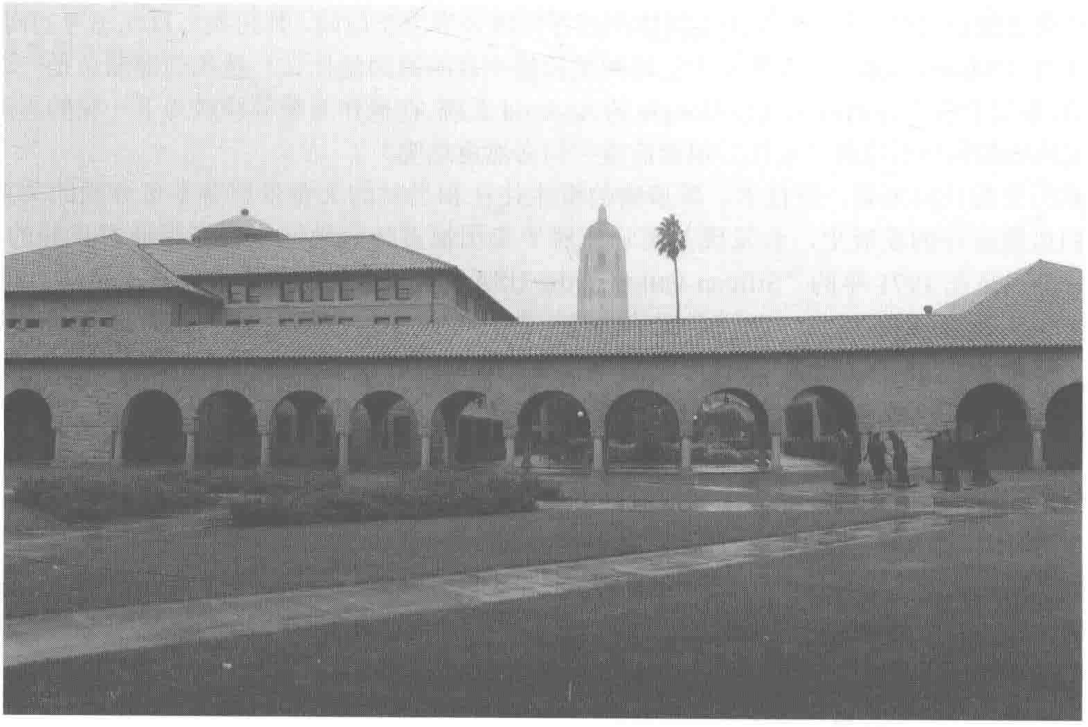


密密麻麻的硅谷大企业

（引用自 cdn.com）

硅谷的“地利”和“人和”，可能主要体现在：

（1）Stanford University



斯坦福大学校园

Stanford University 在硅谷的发展过程中起到了非常关键的作用。20 世纪 50 年代的时候，这所大学还并不是很起眼，各方面条件都比较糟糕，她的毕业生也多数会去东海岸寻求就业机会。后来她的一位教授 Frederick Terman 看到了产业和学术的接合点，从学校里划分出一大块空地来鼓励学生创业，并且指导其中两位学生创立了 Hewlett-Packard 公司。随后的几年他又成立了

Stanford Research Park，这也同时是后来全球各高科技园区的起点，并吸引了越来越多的公司加入。在那段时间里，相信起到核心催化作用的是“产”+“学”的高度结合——将科技产品不断推陈出新产生经济效益，然后再回馈到研究领域。在几十年的跨度里，很多顶尖公司（Google、Yahoo!、HP 等）的创始人都出自该校。有统计显示 Standford 师生及校友创造了硅谷一半以上的总产值，其影响力可见一斑。

（2）便利的地理环境

整个硅谷地区面积并不是很大，属于温带海洋性气候，全年平均温度在 13℃～24℃，污染很小。同时，它依林傍海，陆、海、空都可以很好地与外界相连，这样一来自然有利于人才的引入。

（3）鼓励创新，完善的专利保护机制

从法律上讲，硅谷每年有超过 4000 项的专利申请，工程师和律师的比例达到了 10∶1。在创新点得到保护的同时，也使得初创公司能够得到进一步的发展，从而避免它们被扼杀在摇篮中。从观念上来说，硅谷人对知识产权还是非常尊重的，他们大多认为剽窃是没有技术含量的，相当于“涸泽而渔”。

（4）完善的风投体系，并容忍失败

事实上在硅谷创业，其成本和失败率都很高——其中能存活 3～5 年的公司只有 10%～20%。一方面，风险投资方需要高度容忍这样的失败率；另一方面，在允许快速试错的同时，风险投资方又可以从某些成功中获得巨大收益——硅谷就是一个可以达到这种矛盾平衡的神奇所在地。

“三十年河东，三十年河西”，技术的浪潮总是在不断演进的。从 Symbian、Black Berry，到 Android、iOS，历史经验告诉我们没有一项技术是会永远一成不变的。所以我们在技术领域的探索过程中，既要拿“鱼”，更要学会“渔”——前者是为当前的工作而努力，后者则是为我们的未来做投资。以 Android 操作系统为例，事实上我们除了“知其然”外，还更应该学习它的内部设计思想——即“知其所以然”。当我们真正地理解了那些“精华”所在以后，那么相信以后再遇到任何其他的操作系统，就都可以做到“触类旁通”了。也只有这样，或许才能在快速变革的科技领域中把握住脉搏，立于不败之地。

林学森

于美国硅谷

关于本书第 2 版

在第 1 版上市的这两年时间里，不断有读者来信分享他们阅读本书时的感想和心得，笔者首先要在这里衷心地向大家说声感谢！正是你们的支持和肯定，才有了《深入理解 Android 内核设计思想》（第 2 版）的诞生。

其中有不少读者提到了他们希望在本书后续更新中看到的内容，包括 Android 虚拟机的内部实现原理、Android 的安全机制、Gradle 自动化构建工具等——这些要求都在本次版本更新中得到了体现。

需要特别说明的是，第 2 版中的所有新增和有更新的部分都是基于 Android 最新的 N 版本展开的。由于 Android 版本的更新换代很快，且版本间的差异巨大，导致书中很多内容几乎需要全部重写。另外笔者写书都是在下班后的业余时间进行的，所以即便是每晚奋笔疾书到深夜，再加上周末和节假日时间（如果没有加班工作的话），最后发现更新全书所需时间依然要大于 Android 系统的发布间隔。为了让读者可以早日阅读到大家感兴趣的内容，本次版本的部分章节保留了第 1 版的原有内容——本书下一次再版时会争取将它们更新到 Android 的最新版本。这一点希望得到大家的谅解，谢谢！

致谢

感谢我目前任职公司的领导和同事们，是你们的帮助和支持，才让我更快地融入到了这个大家庭中。在一个到处都是“聪明人”和具有“狼性奋斗者”精神的公司里，每天的进步和知识积累都是让人愉悦的。

感谢人民邮电出版社的编辑，你们的专业态度和处理问题的人性化，是所有作者的“福音”。

感谢我的家人林进跃、张建山、林美玉、杨惠萍、林惠忠、林月明，没有你们的鼓励与理解，就没有本书的顺利出版。

感谢我的妻子张白杨的默默付出，是你工作之外还无怨无悔地在照顾着我们可爱的宝宝，才让我有充足的时间和精力来写作。

感谢所有读者的支持，是你们赋予了我写作的动力。另外，因为个人能力和水平有限，书中难免会有不足之处，希望读者不吝指教，一起探讨学习，作者的联系方式是：**xuesenlin@alumni.cuhk.net**。编辑联系和投稿邮箱是：zhangtao@ptpress.com.cn。本书读者交流 QQ 群为 216840480。

作者

第1版前言

写本书的原因

4次大幅改版，N次修订，前后历时近3年，本书终于要与读者见面了。

在这3年的时间里，Android系统不断更新换代，书本内容也尽可能紧随其步伐——我总是会在第一时间下载到工程源码，然后系统性地比对和研究每次改版后的差异。可以说本书伴随着Android的高速发展，完整地见证了它给大家带来的一次又一次惊喜。

在这么长的写作跨度中，有一个问题始终萦绕在我的脑海中，即“为什么写这本书”？

市面上讲解操作系统的著作很多，主要风格有两种。

- 理论型

高校中采用的操作系统教材多数属于这种类型。它们主要阐述通用的计算机理论与原理，一般不会针对某个具体的操作系统做详细剖析。这类书籍是我们进入计算机科学的“敲门砖”。只有基础打得扎实，研究市面上任何一款操作系统才能做到“有的放矢”。

- 实用型

这类书籍以讲解某个具体的操作系统为主，如市面上就有非常多的关于Windows和Linux系统的。前者因为不开源，谁也不可能深入代码级别进行讲解；而后者则恰恰相反，任何人都能轻松获取到完整的内核源码。在Linux之父经典名言“Read the f***king Source Code”的鼓励下，无数有志之士投入到“代码汪洋”的分析中，从中细细感受大师们的设计艺术。

那么本书属于什么类型呢？个人认为更贴切地说，就是上面两种的结合。

本书的一个主要宗旨是希望读者可以由浅入深地逐步理解Android系统的方方面面。因而在每章节内容的编排上，采用由整体到局部的线索铺展开来——先让读者有一个直观感性的认识，明白“是什么”“有什么用”，然后才剖析“如何做到的”。这样的一个好处是读者在学习过程中不容易产生困惑；否则如果直接切入原理，长篇大论地分析代码，仅一大堆函数调用就可能让人失去学习的方向。这样的结果往往是，读者花了非常多的时间来理清函数关系，但始终不明白代码编写者的意图，甚至连这些函数想实现什么功能都无法完全理解。

本书希望可以从更高的层次，即抽象的、反映设计者思想的角度去理解系统。而在思考的过程中，大部分情况下我们都将从读者容易理解的基础知识开始讲起。就好比画一张素描画一样——先给出一张白纸，勾勒出整体框架，然后针对重点部位细细加工，最后才能还原出完整的画面。另外，本书在对系统原理本身进行讲解的同时，也最大程度地结合工程项目中可能遇到的难点，理论联系实际地进行解析。希望这样的方式既能让读者真正学习到Android系统的设计思想，也能学有所用，增加一些实际的项目开发经验和技巧。

本书的主要内容

细心的读者会发现本书章节中包含了“Android和OpenGL ES”“信息安全基础概述”等看似与本书无关的内容——有些人可能会产生疑问，是否有此必要？

根据我们多年的 Android 项目开发和培训经验，答案就是“非常有必要”。举个例子，Android 的显示系统是围绕 OpenGL ES 来展开的，后者是它的“根基”。但另外，并非所有开发人员都深谙 OpenGL ES。这样导致的结果就是他们在学习显示系统的过程中，有一种“四处碰壁”的感觉——实践证明，正是这些因素直接打击到了大家学习 Android 系统的信心。

因此我们在讲解系统实现原理之前，会最大程度地为读者提炼出所需的背景知识。有了这样的铺垫，相信对大家学习 Android 内核大有裨益。

本书在内容选择上依据的是“研发人员（包括系统开发和应用程序开发）参与实际 Android 项目所需具备的知识”，因而具有较强的实用性。全书共分为 4 篇，涵盖了编译、系统原理、应用原理和系统工具等多个方面。

其中第一篇不仅详细介绍了 Android 源码的下载及编译过程，为读者呈现了“Hello World”式的入门向导——更为重要的是，结合编译系统的架构和内部原理，为各厂家定制自己的 Android 产品提供了参考范例。

Android 本质上只是市面上众多主流的操作系统之一。所以在系统原理的讲解过程中，我们将首先引导读者从计算机体系结构、经典的操作系统理论（比如进程/线程管理、进程间通信等）的角度来思考问题——包括 Android 在内的任何操作系统内核在实现过程中都“逃”不出这些经典的理论范畴。本书虽然是剖析 Android 系统的，但更希望读者可以从中学到“渔”，而不仅仅是“鱼”。

从动态运行的角度来理解，Android 内核是由众多系统服务组成的，如 ActivityManagerService、GUI 系统中的 SurfaceFlinger、音频系统中的 AudioFlinger、输入系统 InputManagerService 等。而各服务之间通信的基础就是 Binder 机制。本书在阐述它们错综复杂的关系中，遵循由“整体到局部”“由点及面”的科学方法，将知识点深入浅出地铺展开来，希望为读者全面理解 Android 内核提供“思维捷径”。

与其他讲解 Android 应用程序的书籍不同，本书在分析 APK 应用程序时的立足点是它的内部实现原理。如 Intent 匹配规则、应用程序的资源适配过程、字符编码的处理、Widget 机制、应用程序的编译打包等都是应用开发人员在工作中经常会遇到的难题。通过系统性地解析隐藏在这些实现背后的原理，有助于他们彻底摆脱困惑，加深对应用开发的理解。

不论是系统工程师还是应用开发人员，Android 调试工具都至关重要。但我们在实际工作中发现，不少研发人员对这些工具“只知其一，不知其二”。因而系统工具篇中将针对常用调试工具进行全面解析，希望由此可以让大家学习到如何“举一反三”，真正把它们的作用发挥得“淋漓尽致”。

本书的主要特点

（1）通过大量情景图片与实例引导读者学习，以求尽量在源码分析外为读者提供更易于理解的思维路径。

（2）作者在展开一个话题时，通常会由浅入深、由总体框架再到细节实现。这样可以保证读者能跟得上分析的节奏，并且“有根有据可循”，尽可能防止部分读者阅读技术书籍时“看了后面忘了前面”的现象。

（3）目前市面上不少 Android 书籍仍停留在 Android 2.3 或者更早期的版本。虽然原理类似，但对于开发人员来说，他们需要与项目研发相契合的技术书籍。本书希望尽可能紧随 Android 的更新步伐，为读者了解最新的 Android 技术提供帮助。

（4）本书的出发点仍是操作系统的经典原理，并以此为根基扩展分析 Android 中的具体实现机制——贯穿其中的是经久不衰的理论知识。

（5）本书所阐述的知识点大部分来源于工程项目研发的经验总结，因而具有较强的实用性，希望可以让读者“知其然，更知其所以然”。做到真正贴近读者，贴近开发需求。

致谢

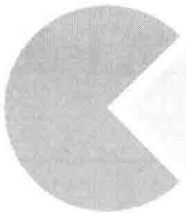
感谢王益民董事长、钟宝英女士长期以来的关心、信任和支持——你们在很多方面都是我们学习的楷模。衷心祝愿王总企业蒸蒸日上、再创辉煌；衷心祝愿钟小姐事事顺心如意、永葆青春。

感谢人民邮电出版社的编辑，你们的专业态度和处理问题的人性化，是所有作者的“福音”。

感谢我的家人、长辈和朋友林进跃、林美玉、林惠忠、刘冰、林月明、温艳，感谢你们长期以来对我工作和生活上无微不至的关心和支持。

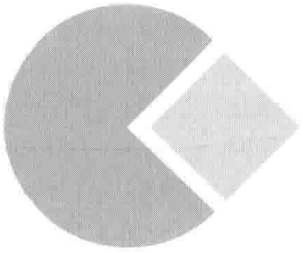
感谢所有读者的支持，是你们赋予了我写作的动力。另外，因为个人能力和水平有限，书中可能还有不足之处，希望读者不吝指教，一起探讨学习，作者的联系方式是：**xuesenlin@alumni.cuhk.net**。编辑联系和投稿邮箱是：zhangtao@ptpress.com.cn。

目　录



第 1 篇

Android 编译篇



第 1 章　Android 系统简介

美国当地时间 2015 年 5 月 28 日，“Google I/O 2016”大会在旧金山市的 Moscone Center 举行。会议公布的官方数据如下：

- 全球已经激活的 Android 设备达到 9 亿次；
- Google Play 中收录了超过 70 万的应用程序；
- 应用程序安装量达到 480 亿次；
- 132 个以上的国家或地区销售 Android 设备；
- 超过 190 个国家或地区可以下载到免费的 Android 应用程序。

2016 年，Google 则直接把大会地址从传统的 Moscone Center 改到了 Shoreline 公园的户外，吸引了成千上万来自全球各地的科技爱好者。

从 2008 年 9 月 Google 发布 Android 1.0 版本开始，Android 已经走过了 8 个年头。在这短短的几年间，这个以机器人为 Logo 的操作系统不仅席卷了全球各地的手机市场，而且与 iOS、Windows Phone 形成三足鼎立之势，更渗透到传统与新兴电子产业的方方面面。越来越多的电子产品已开始采用 Android 系统，如 Android 电视、平板电脑、MP4 等与人们日常生活息息相关的电子设备。

那么，Android 势不可当的魅力从何而来呢？本章将试着以 Android 系统的发展历史为主线，先为读者提供最直观的背景知识，从而为以后的“透过现象看本质”打下一定的基础。

1.1 Android 系统发展历程

“Android”一词先天就充满着天才们改变世界的梦想味。虽然一件杰出的作品并不能只靠“名号”，但毋庸置疑的是，一个叫得响又耐人寻味的名称总会使人产生不自觉的亲近感。这或许就是每个 Android 版本都会有个代号的原因。下面来看看各个版本对应的 Android “外号”，如表 1-1 所示。

表 1-1　　Android 各版本的代号

Code name	Version	API level
(no code name)	1.0	API level 1
(no code name)	1.1	API level 2
Cupcake（纸杯蛋糕）	1.5	API level 3, NDK 1
Donut（甜甜圈）	1.6	API level 4, NDK 2
Éclair（松饼）	2.0	API level 5
Éclair	2.0.1	API level 6
Éclair	2.1	API level 7, NDK 3
Froyo（冻酸奶）	2.2.x	API level 8, NDK 4