HZ BOOKS
华章科技

经典畅销书“深入理解Android”系列新作，资深Android系统专家邓凡平撰写，
全志、高通等公司资深专家担任技术审校并强烈推荐
从专业知识角度和Android系统代码实现角度对Netd、Wi-Fi、NFC和GPS模块代码进行深入剖析，深刻揭示其实现原理和工作机制

移动开发

邓凡平◎著

Understanding Android Internals
Wi-Fi, NFC and GPS

深入理解Android

Wi-Fi、NFC和GPS卷

机械工业出版社
China Machine Press

Understanding Android Internals
Wi-Fi, NFC and GPS

深入理解Android

Wi-Fi、NFC和GPS卷

邓凡平◎著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

深入理解 Android：Wi-Fi、NFC 和 GPS 卷 / 邓凡平著 . —北京：机械工业出版社，2014.3
（移动开发）

ISBN 978-7-111-45683-4

I. 深… II. 邓… III. 移动终端 – 应用程序 – 程序设计 IV. TN929.53

中国版本图书馆 CIP 数据核字（2014）第 024354 号

本书是经典畅销书“深入理解 Android”系列的新作，由资深 Android 系统专家邓凡平先生撰写。从通信专业知识和 Android 系统代码实现的角度，对 Netd、Wi-Fi、NFC 和 GPS 等模块的代码进行深入的剖析，旨在深刻揭示其实现原理和工作流程。其中涉及大量通信相关的专业知识，因此特意邀请全志和高通等著名芯片公司的资深专家担任技术审校。本书从实际应用的需求出发，适合所有 Android 系统工程师、Android 应用开发工程师和 BSP 开发工程师阅读。

全书共 9 章。第 1 章介绍本书的内容组成、工具使用以及参考源码的下载方法。第 2 章介绍 Netd 及相关的背景知识。第 3 ~ 5 章介绍 Wi-Fi 基础知识，重点分析了 wpa_supplicant 的实现，以及 Android 平台中特有的 Wi-Fi 服务模块 WifiService。第 6 ~ 7 章讲解了 Wi-Fi 联盟推出的两项重要技术 Wi-Fi Simple Configuration 和 Wi-Fi P2P，以及它们在 Android 平台中的代码实现。第 8 章详细介绍了 NFC 基础知识，以及 NFC 在 Android 平台中的代码实现。第 9 章讲解了 GPS 原理及 Android 平台中的位置管理服务架构。

深入理解 Android：Wi-Fi、NFC 和 GPS 卷

邓凡平 著

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：白 宇
印 刷：藁城市京瑞印刷有限公司
开 本：186mm × 240mm 1/16
书 号：ISBN 978-7-111-45683-4
版 次：2014 年 4 月第 1 版第 1 次印刷
印 张：36.75
定 价：89.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991 88361066
购书热线：（010）68326294 88379649 68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

前　言

本书主要内容及特色

本书所讲解的 Wi-Fi、NFC 以及 GPS 模块的背后都涉及非常多的专业知识，例如与 Wi-Fi 相关的 802.11 协议、Wi-Fi Alliance（Wi-Fi 联盟）定义的 Wi-Fi Simple Configuration 和 Wi-Fi P2P 协议、NFC Forum 定义的一整套与 NFC 相关的协议、与 GPS 相关的卫星导航原理、AGPS 和 OMA-SUPL 协议等。显然，如果不了解这些专业知识，就不可能真正掌握它们在 Android 平台中的代码实现。

考虑到这些专业知识的重要性，本书在讲解 Android 平台中 Wi-Fi、NFC 和 GPS 模块的实现之前，先重点介绍与代码相关的专业知识。当然，这些专业知识内容如此丰富，在一本书中无法全部涵盖。为了方便读者进一步深入学习，本书每章的最后都会列举笔者在撰写各章时所阅读的参考资料。

以下是本书的内容概述。

- 第 1 章介绍本书的内容组成、使用的工具以及参考源码的下载方法。
- 第 2 章介绍 Netd 以及相关的背景知识。
- 第 3 章介绍 Wi-Fi 基础知识。Wi-Fi 是本章的重点，而且也是当下最热门的技术。
- 第 4 章介绍 wpa_supplicant，它是 Wi-Fi 领域中最核心的软件实现。
- 第 5 章介绍 WifiService，它是 Android 平台中特有的 Wi-Fi 服务模块。
- 第 6 章和第 7 章介绍 Wi-Fi Alliance 推出的两项重要技术——Wi-Fi Simple Configuration

和 Wi-Fi P2P，以及它们在 Android 平台中的代码实现。

- 第 8 章介绍 NFC 背景知识以及 NFC 在 Android 平台中的代码实现。NFC 也是历史比较悠久的技术，希望它能随着 Android 的普及而走向大众。
- 第 9 章介绍 GPS 原理及 Android 平台中的位置管理服务架构。
- 附录为笔者和审稿专家之一的吴劲良先生关于本书定位、学习方法等方面的讨论。相信这些讨论内容能引起读者的共鸣。

本书通过理论和代码相结合的方式进行讲解，旨在引领读者一步步了解 Wi-Fi、NFC 和 GPS 模块的工作原理。总之，笔者希望读者在阅读完本书后能有以下收获。

- 初步掌握 Wi-Fi、NFC 和 GPS 的专业知识。
- 根据其实现代码，进一步加深对这些专业知识的理解。

读者对象

适合阅读本书的读者包括：

- Android 系统开发工程师

系统开发工程师常常需要深入理解系统的运转过程，而本书所涉及的内容正是他们在工作和学习中最想了解的。对具体模块感兴趣的读者也可单刀直入，阅读相关章节。

- Wi-Fi、NFC 或 GPS 的 BSP 开发工程师

BSP 开发工程师更需要对 Android 平台中这些模块的工作原理及背景知识有深入的理解。虽然本书没有介绍这些模块在 Linux Kernel 层的实现，但了解它们在用户空间的工作流程也将极大帮助 BSP 开发工程师拓展自己的知识面。

- 对 Wi-Fi、NFC 和 GPS 感兴趣的在校高年级本科生、研究生和其他读者

在掌握理论的基础上，如何在实际代码中来实现或使用它们也许是众多学子最想知道的。希望这本理论与代码实现深度结合的书籍会助您一臂之力。

如何阅读本书

本书是一本专业知识和代码实现相结合的书籍，所以读者在阅读时应注意以下事项。

- 首先阅读专业知识。如果对这些内容比较了解，可以直接跳转到代码实现。
- 然后是 Android 平台中相关模块的代码实现。这些代码实现往往基于一定的专业知识，所以在阅读代码时务必和前述的专业知识相结合。
- 每章最后都列出了笔者在撰写各章时所参考的资料。资料较多，读者可根据这些内容开展进一步的研究工作。

❑ 每章开头都把本章涉及的源码路径全部列出，而在具体分析源码时，只列出该源码的文件名及所分析的函数或相关数据结构名。例如：

[-->AndroidRuntime.cpp:: 函数或数据结构名]

```
// 源码分析和一些注释
```

最后，本书在描述类之间的关系及函数调用流程上，使用了 UML 的静态类图及序列图。UML 是一个强大的工具，但它的建模规范过于烦琐，为更简单清晰地描述事情的本质，本书并未完全遵循 UML 的建模规范。如图 1 所示，外部类内部的方框用于表示内部类。另外，“外部类 A. 内部类 B”也用于表示内部类。接口和普通类用同一种框图表示。

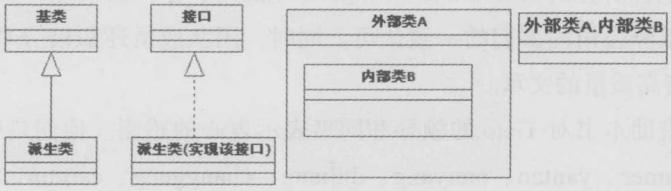

图 1　类图

图 2 所示为本书描述数据结构及成员时使用的 UML 图例。

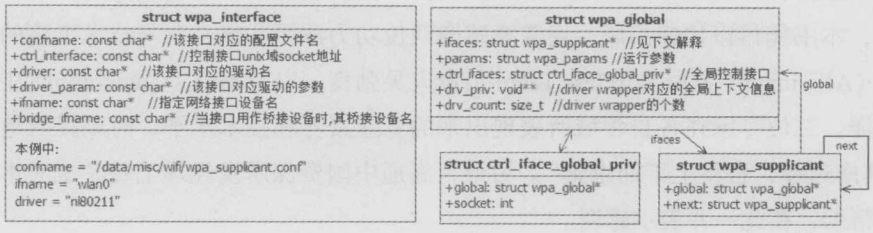

图 2　数据结构图

特别注意　本书所使用的 UML 图都比较简单，读者不必花费大量时间专门学习 UML。另外，出于方便考虑，本书所绘制的 UML 图没有严格遵守 UML 规范，这一点敬请读者谅解。

本书涉及的 Android 源码及一些开发工具的下载地址为 http://115.com/lb/5lbdugrdt4r。关于它们的使用详情，请读者阅读 1.3 节。

勘误和支持

由于作者的水平有限，加之编写时间仓促，书中难免会出现一些错误或不准确的地方，恳请读者不吝批评指正。若有问题，可通过邮箱或在博客上留言与笔者共同商讨。笔者的联

系方式如下。

- 邮箱：fanping.deng@gmail.com
- 博客：blog.csdn.net/innost 和 http://my.oschina.net/innost/blog

致谢

首先要感谢杨福川编辑的大力支持。另外，要感谢本书审稿编辑白宇严谨负责的工作。

特别感谢 Tieto 公司。Tieto 开放的企业文化、Android 团队高效的工作效率、团队成员之间默契的工作配合程度，以及领导无私而有力的支持着实让我感到幸运和自豪。在 Tieto 就职的一年中，笔者所在的 Android 团队不仅成功赢得了客户的信任，更是得到了 Tieto 公司总部和其他国家分公司同事们的一致认可。同时，团队成员还积极分享，并在《程序员》杂志上发表了六篇高质量的文章。

在此，笔者借助本书对 Tieto 的领导和同事表示衷心的感谢。他们是中国北京分公司的 Leo、hongbin、James、yantao、meiyang、dujiang、changgeng、caimin、wenjing、huaizhi、huirong、xinzhi、huimin、yuzheng、Liuxuan、Emily、Diego、jinghua、Jenny 等，中国成都分公司的 tianxiang、chengguo 等，波兰分公司的 Marcin、Marciej、Filip Matusiak 等、捷克分公司的 Vaclav、Bronislav、Petrous Jan 等、芬兰分公司的 Mikel Echegoyen。

当然，本书能得以快速出版，还需要感谢两位功力深厚并热心参与技术审稿的专家。他们是全志（Allwinner）公司 Wireless Team 负责人吴劲良，以及高通（Qualcomm）中国资深研发经理杨洋。二位专家在各自领域所表现出来的专业素养和技术水平，时刻提醒笔者应牢记“路漫漫其修远兮，吾将上下而求索”。另外，高通中国资深研发经理毛晓冬也对本书成功编写提供了帮助，在此一并表示感谢。

最后，感谢我的家人，尤其是我的妻子。希望明年上天能恩赐一个健康可爱的宝宝，这样，我将拥有更加无穷的动力编写更多书籍来回馈花费宝贵时间和精力关注本书的读者，以及所有在人生和职业道路上曾给予我指导的诸位师长。

目　录

第1章
准备工作

本章主要内容

- 本书的内容组成；
- 工具使用；
- 本书资源下载说明。

1.1 Android 系统架构

Android 是 Google 公司推出的一款手机开发平台。该平台本身是基于 Linux 内核的，图 1-1 展示了这个系统的架构。

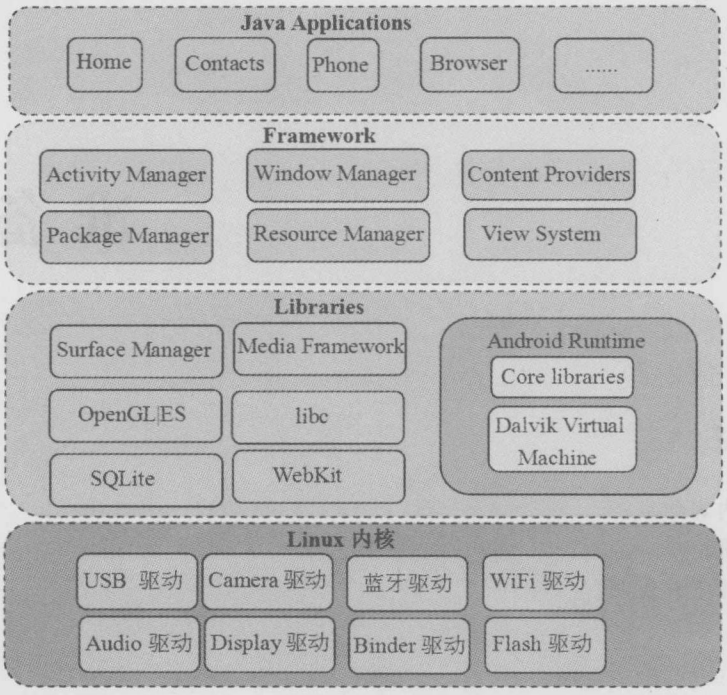

图 1-1 Android 系统架构

从图 1-1 可知，Android 系统大体可分为四层，从下往上依次如下。

- Linux 内核层，目前 Android 4.4（代号为 KitKat）基于 Linux 内核 3.4 版本。
- Libraries 层，这一层提供动态库（也叫共享库）、Android 运行时库、Dalvik 虚拟机[⊖]等。从编程语言方面来说，这一层大部分都是用 C 或 C++ 写的，所以也可以简单地把它看成是 Native 层。
- Libraries 层之上是 Framework 层，这一层大部分用 Java 语言编写。它是 Android 平台上 Java 世界的基石。
- Framework 层之上是 Applications 层，和用户直接交互的就是这些应用程序，它们都是用 Java 开发的。

1.2 工具使用

本节介绍 Android 开发和源码研究过程中的三件利器。

⊖ 4.4 版本新增了 ART 虚拟机运行时，相信它的出现能提升应用程序的运行速度。

1.2.1 Source Insight 的使用

Source Insight 是阅读源码的必备工具，是一个 Windows 下的软件，在 Linux 平台上可通过 wine 安装。下面介绍一下如何在 Source Insight 中导入源码。

使用 Source Insight 时，需要新建一个源码工程，通过菜单项 Project → New Project，可指定源码的目录。

提示 如果把 Android 所有源代码都加到工程中，将导致 Source Insight 运行速度非常慢。

实际上，只需要将当前分析的源码目录加到工程即可。例如，新建一个 Source Insight 工程后，只把源码 /framework/base 目录加进去了。另外，当一个目录下的源码分析完后，可以通过 Project → Add and Remove Project Files 选项把无须分析的目录从工程中去掉。上述步骤如图 1-2 所示。

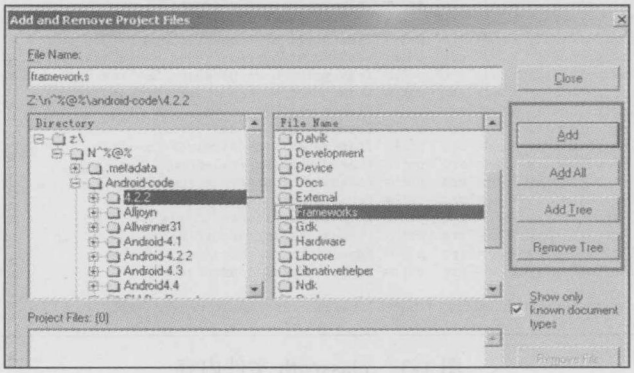

图 1-2 添加或删除工程中的目录

从图 1-2 右边的框可知，Source Insight 支持动态添加或删除目录。通过这种方式可极大减少 Source Insight 的工作负担。

提示 一般首先把 framework/base 下的目录加到工程，以后如有需要，再把其他目录加进来。另外，关于 Source Insight 其他使用技巧，可参考《深入理解 Android：卷 I》第 1 章。

1.2.2 Eclipse 的使用

笔者一般使用 Source Insight 来查看 Native 代码，而 Android 推荐的集成开发工具 Eclipse 既能查看 Java 代码和 Native 代码，也能调试系统核心进程。

1. 导入 Android Framework Java 源码

注意，这一步必须编译整个 Android 源码才可以实施，步骤如下。

1）将 Android 源码目录 /development/ide/eclipse/.classpath 复制到 Android 源码根目录。

2）打开 Android 源码根目录下的 .classpath 文件。该文件是供 Eclipse 使用的，其中保存的是源码目录中各个模块的路径。

由于我们只关心 Framework 相关的模块，因此可以把一些不是 Framework 的目录从该文件中注释掉。同时，去掉不必要的模块也可加快 Android 源码导入速度。图 1-3 所示为该文件的部分内容。

```
-->
<classpathentry kind="src" path="frameworks/base/cmds/am/src"/>
<classpathentry kind="src" path="frameworks/base/cmds/input/src"/>
<classpathentry kind="src" path="frameworks/base/cmds/pm/src"/>
<classpathentry kind="src" path="frameworks/base/cmds/svc/src"/>
<classpathentry kind="src" path="frameworks/base/core/java"/>
<classpathentry kind="src" path="frameworks/base/drm/java"/>
<classpathentry kind="src" path="frameworks/base/graphics/java"/>
<classpathentry kind="src" path="frameworks/base/icu4j/java"/>
<classpathentry kind="src" path="frameworks/base/keystore/java"/>
<classpathentry kind="src" path="frameworks/base/location/java"/>
<classpathentry kind="src" path="frameworks/base/location/lib/java"/>
<classpathentry kind="src" path="frameworks/base/media/java"/>
<classpathentry kind="src" path="frameworks/base/media/mca/effect/java"/>
<classpathentry kind="src" path="frameworks/base/media/mca/filterfw/java"/>
<classpathentry kind="src" path="frameworks/base/media/mca/filterpacks/java"/>
<classpathentry kind="src" path="frameworks/base/nfc-extras/java"/>
<classpathentry kind="src" path="frameworks/base/obex"/>
<classpathentry kind="src" path="frameworks/base/opengl/java"/>
<classpathentry kind="src" path="frameworks/base/packages/FusedLocation/src"/>
<classpathentry kind="src" path="frameworks/base/packages/SettingsProvider/src"/>
<classpathentry kind="src" path="frameworks/base/packages/SystemUI/src"/>
<classpathentry kind="src" path="frameworks/base/policy/src"/>
<classpathentry kind="src" path="frameworks/base/sax/java"/>
<classpathentry kind="src" path="frameworks/base/services/java"/>
<classpathentry kind="src" path="frameworks/base/telephony/java"/>
<classpathentry kind="src" path="frameworks/base/test-runner/src"/>
<classpathentry kind="src" path="frameworks/base/voip/java"/>
<classpathentry kind="src" path="frameworks/base/wifi/java"/>
<classpathentry kind="src" path="frameworks/ex/carousel/java"/>
<classpathentry kind="src" path="frameworks/ex/chips/src"/>
<classpathentry kind="src" path="frameworks/ex/common/java"/>
<!--
  classpathentry kind="src" path="frameworks/ex/photoviewer/src"/
-->
```

图 1-3 .classpath 文件内容

然后，单击 Eclipse 菜单栏 New → Java Project，弹出如图 1-4 所示的对话框。设置 Location 为 Android 4.2 源码所在路径。

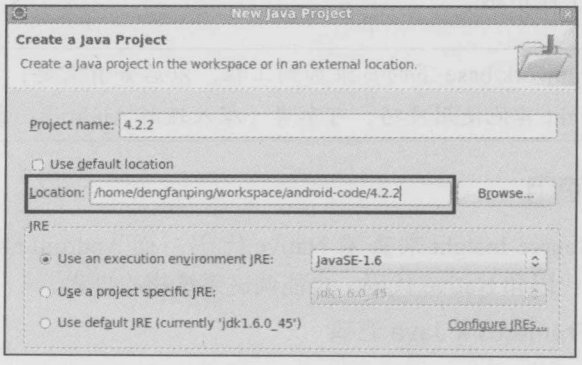

图 1-4 导入 Android 源码

由于 Android 4.2 源码文件较多，导入过程会持续较长一段时间。