



内核
Android 源码
GoldFish 下的驱动
MSM 内核和驱动
传感器、照相机
Wi-Fi、GPS

Android

驱动开发与移植 | 实战详解

李骏 陈小玉 编著

面向实战的基于驱动和移植开发的技术书

- 内容全面，涵盖了 Android 系统的所有驱动系统
- 每个实例都进行了详细讲解，力求读者易学、易用
- 凝聚一线工程师的智慧结晶，实用性强

内核
Android 源码
GoldFish 下的驱动
MSM 内核和驱动
传感器、照相机
Wi-Fi、GPS

Android

驱动开发与移植 | 实战详解

李骏 陈小玉 编著

人民邮电出版社

北 京

图书在版编目 (C I P) 数据

Android驱动开发与移植实战详解 / 李骏, 陈小玉编
著. — 北京: 人民邮电出版社, 2012. 8
ISBN 978-7-115-28361-0

I. ①A… II. ①李… ②陈… 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(2012)第105448号

内 容 提 要

Android 凭借其开源性、优异的用户体验和极为方便的开发方式, 赢得了广大用户和开发者的青睐, 目前已经发展成为市场占有率很高的智能手机操作系统。

全书分为 18 章, 依次讲解了 Android 系统的基本知识, Linux 内核的基本知识, 分析了 Android 系统的源码, 深入分析 HAL 层的基本知识, GoldFish 下的驱动、MSM 内核和驱动、OMAP 内核和驱动、显示系统驱动、输入系统驱动、振动器系统驱动、音频系统驱动和视频输出系统驱动, 多媒体框架, 传感器系统、照相机系统、Wi-Fi 系统、蓝牙系统、GPS 系统和电话系统的知识。在每一章中, 重点介绍了与 Android 驱动开发相关的底层知识, 并对 Android 源码进行了详细的分析及驱动开发实现。

本书适合 Android 程序员、研发人员及 Android 爱好者学习, 也可以作为相关培训学校和大专院校相关专业的教学用书。

Android 驱动开发与移植实战详解

- ◆ 编 著 李 骏 陈小玉
责任编辑 张 涛
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
大厂聚鑫印刷有限责任公司印刷
- ◆ 开本: 800×1000 1/16
印张: 35.25
字数: 864 千字 2012 年 8 月第 1 版
印数: 1—3 000 册 2012 年 8 月河北第 1 次印刷

ISBN 978-7-115-28361-0

定价: 79.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154

广告经营许可证: 京崇工商广字第 0021 号

前 言

经过几年的发展，Android 已经成为了移动操作系统的翘楚。目前，已有众多的设备开始选择使用 Android 系统，如智能手机、智能电视、平板电脑、上网本、MP3、MP4、智能相机、导航仪等。相信在不久的将来，还将有更多采用 Android 系统的高科技产品进入到我们的生活中。这些设备将产生各种各样的应用需求，尤其是与 Android 系统底层相关的应用，这将给开发者带来大量的机会，尤其是系统级应用开发工程师。

随着 Android 版本的更新，从最初的触屏到现在的多点触摸，从普通的联系人到现在的数据同步，从简单的 Google Map 到现在的导航系统，这都说明 Android 已经逐渐稳定，而且功能越来越强大。此外，Android 平台不仅支持 Java 和 C++ 等主流的编程语言，还支持 Ruby、Python 等脚本语言，甚至 Google 专为 Android 的应用开发推出了 Simple 语言，这使得 Android 有着非常广泛的开发群体。

本书特色

本书内容丰富，实例覆盖全面。我们的目标是通过一本图书，提供多本图书的价值，读者可以根据自己的需要有选择地阅读。在内容的编写上，本书具有以下特色。

(1) 结构合理

从用户的实际需要出发，科学地安排知识结构，内容由浅入深，叙述清楚。全书精心筛选了具有代表性的、读者关心的典型知识点，几乎包括 Android 底层和驱动技术的各个方面。

(2) 易学易懂

本书条理清晰、语言简洁，可帮助读者快速掌握每个知识点。使读者既可以按照本书编排的章节顺序进行学习，也可以根据自己的需求对某一章节进行有针对性的学习。

(3) 实用性强

本书注重实用性和可操作性，详细讲解了各个部分的源码知识，使用户在掌握相关的操作技能的同时，还能学习到相应的开发知识。

读者对象

Android 程序员

Android 底层和框架开发者

大中专院校相关专业的老师和学生

相关专业毕业设计的学生

Android 编程爱好者

相关培训机构的老师和学员

本团队在编写过程中，得到了出版社的大力支持，正是编辑的求实、耐心和效率才能使本书快速出版。南阳理工学院陈小玉编写了第 7~11 章和第 16 章。另外也十分感谢我们的家人，在我们写作的时候给予了巨大的支持。由于本团队水平有限，有纰漏和不尽如人意之处在所难免，诚请读者提出意见或建议，以便修订并使之更臻完善。编辑联系邮箱为 zhangtao@ptpress.com.cn。源程序下载地址为：www.toppr.net。

编 者

目 录

第 1 章 迅猛发展的 Android 系统..... 1	2.3 Linux 内核简介 14
1.1 智能手机世界 1	2.3.1 内核的体系结构 14
1.1.1 何谓智能手机 1	2.3.2 和 Android 相关的 Linux 内核知识 17
1.1.2 当前主流智能手机系统 1	2.4 分析 Linux 内核源码 20
1.2 Android 的自身优势 3	2.4.1 源码目录结构 21
1.2.1 开源 3	2.4.2 浏览源码的工具 23
1.2.2 强大的开发团队支持 3	2.4.3 用汇编语言编写内核 代码 24
1.2.3 实行奖励机制 4	2.4.4 Linux 内核的显著特性 24
1.3 认识驱动 4	2.4.5 学习 Linux 内核的过程 32
1.4 分析 Android 架构 5	第 3 章 开始分析 Android 源码 37
1.4.1 操作系统层 (OS) 5	3.1 搭建 Linux 开发环境和工具 37
1.4.2 各种库和 Android 运行 环境 6	3.1.1 搭建 Linux 开发环境 37
1.4.3 应用程序 7	3.1.2 设置环境变量 38
1.4.4 应用程序框架 7	3.1.3 安装编译工具 38
1.5 开源的问题 8	3.2 获取 Android 源码 39
1.5.1 雾里看花的开源 8	3.3 分析 Android 源码结构 41
1.5.2 从选择 Java 开始谈为什么 不开源驱动程序 8	3.4 编译 Android 源码 46
1.5.3 对驱动开发者的影响 9	3.5 运行 Android 源码 48
第 2 章 简要分析 Linux 内核 10	3.6 实践演练——演示两种编译 Android 程序的方法 49
2.1 Linux 基础 10	3.6.1 编译 Native C 的 helloworld 模块 49
2.1.1 Linux 历史简介 10	3.6.2 手工编译 C 模块 50
2.1.2 主要版本 10	3.7 编译 Android Kernel 53
2.1.3 Linux 的发展机遇 11	3.7.1 获取 Goldfish 内核代码 53
2.2 Android 和 Linux 的关系 11	3.7.2 获取 MSM 内核代码 56
2.2.1 Android 继承于 Linux 12	3.7.3 获取 OMAP 内核代码 56
2.2.2 Android 和 Linux 内核的 区别 12	

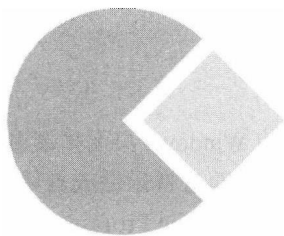
3.7.4 编译 Android 的 Linux 内核.....	56	5.4 移植总结.....	104
3.8 运行模拟器.....	58	5.4.1 移植各个 Android 部件的 方式.....	104
3.8.1 Linux 环境下运行模拟器 的方法.....	58	5.4.2 辅助工作.....	105
3.8.2 模拟器辅助工具——ADB.....	59	第 6 章 常见的驱动平台.....	112
第 4 章 驱动移植.....	62	6.1 专用驱动.....	112
4.1 Android 移植.....	62	6.1.1 Binder 驱动程序.....	112
4.1.1 移植的任务.....	62	6.1.2 Logger 驱动.....	122
4.1.2 移植的内容.....	62	6.1.3 组件 Lowmemorykiller.....	123
4.1.3 驱动开发需要做的工作.....	63	6.1.4 Timed Output 驱动程序.....	123
4.2 Android 对 Linux 的改造.....	64	6.1.5 Timed Gpio 驱动程序.....	125
4.2.1 Android 的核心驱动.....	64	6.1.6 唤醒和休眠.....	126
4.2.2 为 Android 构建 Linux 操作系统.....	67	6.1.7 Ashmem 驱动程序.....	130
4.3 内核空间 and 用户空间接口.....	68	6.1.8 Pmem 驱动程序.....	133
4.3.1 实现系统和硬件之间的 交互.....	68	6.1.9 Alarm 驱动程序.....	134
4.3.2 实现内核到用户空间的 数据传输.....	70	6.1.10 USB Gadget 驱动程序.....	136
4.4 三类驱动程序.....	74	6.1.11 Android Paranoid 驱动 程序.....	137
4.4.1 字符设备驱动.....	74	6.2 Goldfish 设备驱动.....	138
4.4.2 块设备驱动.....	82	6.3 MSM 内核和驱动.....	146
4.4.3 网络设备驱动.....	86	6.3.1 MSM 基础.....	146
第 5 章 深入详解 HAL 层.....	87	6.3.2 移植 MSM 内核.....	148
5.1 初识 HAL 层.....	87	6.3.3 移植 MSM.....	150
5.1.1 HAL 层简介.....	87	6.3.4 高通特有的组件.....	156
5.1.2 比较 HAL_legacy 和 HAL.....	89	6.4 OMAP 内核和驱动.....	157
5.2 分析 HAL 层源码.....	89	6.4.1 OMAP 基础.....	157
5.2.1 分析 HAL module.....	89	6.4.2 OMAP 内核.....	159
5.2.2 分析 mokoid 工程.....	92	6.4.3 移植 OMAP 体系结构.....	161
5.3 Sensor 在 HAL 层的表现.....	101	6.4.4 移植 Android 专用驱动 和组件.....	168
5.3.1 HAL 层的 Sensor 代码.....	101	6.4.5 OMAP 的设备驱动.....	169
5.3.2 总结 Sensor 编程的流程.....	103	第 7 章 输入系统驱动.....	175
		7.1 输入系统介绍.....	175
		7.1.1 Android 输入系统结构元素 介绍.....	175

7.1.2	Android 输入系统驱动	176	8.4	实现电话系统驱动	219
7.1.3	Input 系统的层次结构	177	8.4.1	RIL 中消息队列的建立	219
7.1.4	移植工作	177	8.4.2	与底层 Modem 通信	224
7.2	Input 驱动源码分析	178	8.5	深入分析实现 Android 电话系统的流程	229
7.2.1	文件 input.h	178	8.5.1	初始启动流程	229
7.2.2	文件 KeycodeLabels.h	182	8.5.2	接收信息流程	231
7.2.3	文件 KeyCharacterMap.h	186	8.5.3	等待硬件响应	233
7.2.4	KI 格式文件	187			
7.2.5	kcm 格式文件	188	第 9 章	显示系统驱动	236
7.2.6	文件 EventHub.cpp	188	9.1	显示系统基础	236
7.3	Input 设备的运作过程	191	9.1.1	Android 的版本	236
7.3.1	Input 设备的注册 (硬件驱动层)	192	9.1.2	不同的显示系统	237
7.3.2	Input 子系统的加载过程 (子系统核心层)	192	9.1.3	FrameBuffer 驱动的 使用基础	237
7.3.3	Input 子系统的事件处理 (事件处理层)	194	9.2	移植 Android 显示系统	238
7.4	模拟器的输入驱动	198	9.2.1	FrameBuffer 驱动程序	238
7.5	高通平台的输入驱动实现	199	9.2.2	硬件抽象层	242
7.5.1	触摸屏驱动	199	9.3	实现显示系统的驱动程序	254
7.5.2	按键和轨迹球驱动	205	9.3.1	Goldfish 中的 FrameBuffer 驱动程序	254
7.6	OMAP 处理器中的输入驱动 实现	208	9.3.2	使用 Gralloc 模块的驱动 程序	257
7.6.1	触摸屏驱动程序	208	9.4	MSM 中显示驱动的实现	266
7.6.2	键盘驱动程序	208	9.4.1	MSM 中的 FrameBuffer 驱动程序	267
第 8 章	电话系统驱动	210	9.4.2	MSM 中的 Gralloc 驱动程序	269
8.1	电话系统基础	210	9.5	OMAP 中显示驱动的实现	276
8.1.1	Android 电话系统简介	210	9.5.1	文件 omapfb-main.c	276
8.1.2	深入分析电话系统的 实现文件	211	9.5.2	文件 omapfb.h	278
8.1.3	电话系统结构	212	9.6	6416 中 FrameBuffer 的工作 原理	279
8.2	移植 Modem 驱动和 RIL 硬件 抽象层	213	第 10 章	音频系统驱动	286
8.3	移植和调试	214	10.1	音频系统结构	286
8.3.1	驱动程序	214	10.2	音频系统的层次	287
8.3.2	RIL 接口	216			

10.2.1	层次说明	287	11.2	移植的内容	346
10.2.2	Media 库中的 Audio 框架	288	11.3	分析硬件抽象层	346
10.2.3	本地代码	291	11.3.1	Overlay 系统硬件抽象层的 接口	346
10.2.4	JNI 代码	294	11.3.2	实现硬件抽象层	349
10.2.5	Java 代码	295	11.3.3	实现接口	350
10.3	移植工作	296	11.4	实现 Overlay 硬件抽象层	351
10.3.1	两个任务	296	11.5	在 OMAP 平台实现 Overlay 系统	353
10.3.2	Audio 的硬件抽象层	296	11.5.1	实现输出视频驱动 程序	353
10.3.3	实现 AudioFlinger 中的 Audio 硬件抽象层	298	11.5.2	实现 Overlay 硬件 抽象层	354
10.3.4	真正实现 Audio 硬件 抽象层	304	11.6	系统层调用 Overlay 模块	360
10.4	MSM 平台实现 Audio 驱动系统	304	11.6.1	测试文件	360
10.4.1	实现 Audio 驱动程序	304	11.6.2	在 Android 系统中创建 Overlay	361
10.4.2	实现硬件抽象层	305	11.6.3	管理 Overlay HAL 模块	363
10.5	OSS 平台实现 Audio 驱动系统	309	11.7	抽象层实现 (V4l2 驱动实现方式)	364
10.5.1	OSS 驱动程序介绍	309			
10.5.2	mixer	310	第 12 章	振动器系统驱动	369
10.6	ALSA 平台实现 Audio 系统	317	12.1	振动器系统结构	369
10.6.1	ALSA 基础	317	12.1.1	硬件抽象层	371
10.6.2	注册音频设备和音频 驱动	318	12.1.2	JNI 框架部分	372
10.6.3	ALSA 的底层接口	318	12.2	开始移植	373
10.6.4	放音流程	320	12.2.1	移植振动器驱动程序	373
10.6.5	录音流程	325	12.2.2	实现硬件抽象层	374
10.6.6	在 Android 中使用 ALSA 声卡	328	12.3	在 MSM 平台实现振动器驱动	375
10.6.7	在 OMAP 平台移植 Android 的 ALSA 声卡驱动	337	第 13 章	Android 多媒体插件框架	379
10.7	6410 中的 ALSA 驱动	340	13.1	Android 多媒体插件	379
10.7.1	ALSA 的设备文件	340	13.2	需要移植的内容	380
10.7.2	创建声卡和 PCM 设备	341	13.3	OpenCore 引擎详解	381
			13.3.1	OpenCore 的层次结构	381
第 11 章	视频输出系统驱动	344	13.3.2	OpenCore 的代码结构	382
11.1	视频输出系统结构	344	13.3.3	OpenCore 的编译结构	383

13.3.4	OpenCore OSCL	387	15.2.3	实现上层部分	456
13.3.5	实现 OpenCore 中的 OpenMax 部分	389	15.3	实现传感器	460
13.3.6	OpenCore 的扩展	401	第 16 章	Wi-Fi 系统、蓝牙系统 和 GPS 系统	466
13.4	Stagefright 引擎	408	16.1	Wi-Fi 系统的应用和移植	466
13.4.1	代码结构	408	16.1.1	Wi-Fi 系统的结构	466
13.4.2	实现 OpenMax 接口	408	16.1.2	移植的内容	468
13.4.3	Video Buffer 传输流程	412	16.1.3	移植和调试	468
第 14 章	Camera 照相机驱动	418	16.1.4	OMAP 平台实现 Wi-Fi	476
14.1	Camera 系统的结构	418	16.1.5	配置 Wi-Fi	478
14.2	移植的内容	421	16.1.6	SDIO 设备的移植	481
14.2.1	fimc 驱动模块的加载	421	16.1.7	移植 Wi-Fi 驱动的注意事项	485
14.2.2	V4l2 驱动的用法	425	16.2	蓝牙系统的应用和移植	486
14.3	移植和调试	426	16.2.1	蓝牙结构	487
14.3.1	V4L2 驱动程序	426	16.2.2	移植的内容	489
14.3.2	硬件抽象层	434	16.2.3	具体移植	490
14.4	实现 Camera 系统的硬件 抽象层	438	16.2.4	MSM 平台的蓝牙驱动	492
14.4.1	Java 程序部分	438	16.2.5	本地适配器连接过程	494
14.4.2	Java 本地调用部分	439	16.2.6	远程适配器连接过程	498
14.4.3	本地库 libui.so	440	16.2.7	分析 6410 的蓝牙驱动	500
14.4.4	Camera 服务 libcameraservice.so	441	16.3	定位系统	510
14.5	实现 Camera 系统	445	16.3.1	系统结构	510
14.5.1	在 MSM 平台实现 Camera 系统	445	16.3.2	移植的内容	512
14.5.2	OMAP 平台实现 Camera 系统	448	16.3.3	移植和调试	512
14.6	借助 Sensor 驱动使用照相机 系统	449	16.3.4	GPS 的串口驱动和数据 读取	525
第 15 章	传感器系统驱动	452	第 17 章	振动器驱动和警报器驱动	529
15.1	传感器系统的结构	452	17.1	Alarm 系统基础	529
15.2	移植 Sensor 驱动	454	17.1.1	Alarm 系统的结构	529
15.2.1	移植驱动程序	454	17.1.2	移植的内容	532
15.2.2	移植硬件抽象层	455	17.2	移植和调试	532
			17.3	实现 Alarm 驱动	535
			17.4	MSM 平台实现 Alarm	542

第 18 章	光系统驱动和电池系统驱动	544
18.1	Lights 光系统的应用和移植	544
18.1.1	Lights 系统的结构	544
18.1.2	移植的内容	545
18.1.3	移植和调试	546
18.1.4	MSM 平台实现光系统	547
18.1.5	深入分析 Android 的光系统	548
18.2	Battery 电池系统的应用和移植	549
18.2.1	Battery 系统的结构	549
18.2.2	移植的内容	551
18.2.3	移植和调试	551
18.2.4	模拟器中实现电池系统	554



第1章 迅猛发展的 Android 系统

Android（安卓）是 IT 巨头谷歌公司推出的一款手机系统，是建立在 Linux 内核基础之上的，能够迅速建立手机软件的解决方案。Android 的功能十分强大，作为一个新兴的热点，已经成为了软件行业中的一股新兴力量。据市场数据统计，Android 系统已经成为当今市场占有率第一位的智能手机系统。本章将简单介绍 Android 的发展历程和背景，让读者了解 Android 的辉煌之路。

1.1 智能手机世界

智能手机是指具有像个人电脑那样独立的操作系统，用户可以在上面安装自己需要的第三方软件或游戏程序，并且可以通过移动通信网络接入无线网络。在 Android 系统诞生之前已经有很多优秀的智能手机产品，例如家喻户晓的 Symbian（塞班）系列和微软的 Windows Mobile 系列等。

1.1.1 何谓智能手机

当今公认的智能机必须具备下面的功能标准：

- (1) 操作系统必须支持新应用的安装；
- (2) 高速度处理芯片；
- (3) 支持播放式的手机电视；
- (4) 大存储芯片和存储扩展能力；
- (5) 支持 GPS 导航。

后来手机联盟制定了一个标准，在该标准中列出了智能手机的如下主要特点：

- (1) 具备普通手机的全部功能，例如可以进行正常的通话和发短信等手机应用；
- (2) 是一个开放性的操作系统，在系统平台上可以安装更多的应用程序，从而实现功能的无限扩充；
- (3) 具备上网功能；
- (4) 具备 PDA 的功能，实现个人信息管理、日程记事、任务安排、多媒体应用、浏览网页；
- (5) 可以根据个人需要扩展机器的功能；
- (6) 扩展性能强，并且可以支持很多第三方软件。

1.1.2 当前主流智能手机系统

当今最主流的智能机系统有微软的 Windows Phone 系列、塞班系列、PDA、黑莓、苹果的 iOS

和本书的主角 Android。

1. 微软的 Windows Mobile

Windows Mobile 是微软公司的一款杰出产品，Windows Mobile 将熟悉的 Windows 桌面扩展到了个人设备中。使用 Windows Mobile 操作系统的设备主要有 PPC 手机、PDA、随身音乐播放器等。Windows Mobile 操作系统有三种，分别是 Windows Mobile Standard、Windows Mobile Professional、Windows Mobile Classic。当前的最新版本是 Windows Phone 7，Windows Phone 8 也即将发布。

2. 塞班系统

塞班系统 Symbian 是一款久负盛名的手机系统，最初是由诺基亚、索尼爱立信、摩托罗拉、西门子等几家大型移动通信设备商共同出资组建的一个合资公司，职责是专门研发手机操作系统。Symbian 有着良好的界面，采用内核与界面分离技术，对硬件的要求比较低，支持 C++、Visual Basic 和 J2ME。目前根据人机界面的不同，Symbian 体系的 UI (User Interface, 用户界面) 平台分为 Series60、Series80、Series90、UIQ 等。其中 Series60 主要是给数字键盘手机用，Series80 是为完整键盘所设计，Series90 则是为触控笔方式而设计。

2010 年 9 月，诺基亚宣布将从 2011 年 4 月起从 Symbian 基金会 (Symbian Foundation) 手中收回 Symbian 操作系统控制权。

3. Palm

Palm 是流行的个人数字助理 (PDA，又称掌上电脑) 的传统名字。从广义上讲，Palm 是 PDA 的一种，是 Palm 公司发明的。而从狭义上讲，Palm 是 Palm 公司生产的 PDA 产品，区别于 SONY 公司的 Clie 和 Handspring 公司的 Visor/Treo 等其他运行 Palm 操作系统的 PDA 产品。其显著特点之一是写入装置输入数据的方法，能够点击显示器上的图标选择输入的项目。2009 年 2 月 11 日，Palm 公司 CEO Ed Colligan 宣布以后将专注于 WebOS 和 Windows Mobile 的智能设备，而将不会再有基于 “Palm OS” 的智能设备推出，除了 Palm Centro 会在以后和其他运营商合作时继续推出。

4. 黑莓 BlackBerry

BlackBerry 是加拿大 RIM 公司推出的一种移动电子邮件系统终端，其特色是支持推动式电子邮件、手提电话、文字短信、互联网传真、网页浏览及其他无线资讯服务，其最大优势在于收发邮件。

5. iOS

iOS 是苹果手机产品 iPhone 的操作系统，iPhone 是一款手机产品，由苹果公司在 2007 年 1 月 9 日举行的 Macworld 宣布推出，并在 2007 年 6 月 29 日在美国上市。iPhone 实现了将创新的移动电话、可触摸宽屏 iPod 以及具有桌面级电子邮件、网页浏览、搜索和地图功能的突破性因特网通信设备这三种产品完美地融为一体。每一件 iPhone 产品都是一件艺术品，无论是外观还是品质和性能，都拥有超高的人气。推出的每一款新 iPhone，都马上成为市面智能手机的标杆。在 2011 年 10 月 4 日推出了当前最新的 iPhone 4S。

6. Android

Android 是本书的主角，是一款基于 Linux 内核的开源手机操作系统的名称，该平台由操作系统、中间件、用户界面和应用软件组成，号称是首个为移动终端打造的真正开放和完整的移动软件。

1.2 Android 的自身优势

Android 采用了 WebKit 浏览器引擎，具备触摸屏、高级图形显示和上网功能，用户能够在手机上查看电子邮件、搜索网址和观看视频节目等，同时 Android 还具有比 iPhone 等其他手机更强的搜索功能，可以说是一种融入全部 Web 应用的平台。正是因为其特有的巨大优势，在 2010 年下半年，Android 规模便超越了苹果 iPhone，大有一统手机系统之势！并且终于在 2011 年超越了塞班，成为市场占有率最高的智能手机系统。

1.2.1 开源

Android 出身于 Linux 世家，是一款开源的手机操作系统。Android 功成名就之后，各大手机联盟纷纷加入，这个联盟由包括中国移动、摩托罗拉、高通、宏达电子和 T-Mobile 在内的 30 多家技术和无线应用的领军企业组成。通过与运营商、设备制造商、开发商和其他有关各方结成深层次的合作伙伴关系，希望借助建立标准化、开放式的移动电话软件平台，在移动产业内形成一个开放式的生态系统。

1.2.2 强大的开发团队支持

Android 的研发队伍阵容强大，包括摩托罗拉、Google、HTC（宏达电子）、PHILIPS、T-Mobile、高通、魅族、三星、LG 以及中国移动在内的 34 家企业。它们都将基于该平台开发手机的新型业务，应用之间的通用性和互联性将在最大程度上得到保持。并且还成立了手机开放联盟，联盟中的成员由手机制造商、半导体公司和软件公司组成，具体名单如下。

1. 手机制造商

台湾宏达国际电子（HTC）（Palm 等多款智能手机的代工厂），摩托罗拉（美国最大的手机制造商），韩国三星电子（仅次于诺基亚的全球第二大手机制造商），韩国 LG 电子，中国移动（全球最大的移动运营商），日本 KDDI（2900 万用户），日本 NTT DoCoMo（5200 万用户），美国 Sprint Nextel（美国第三大移动运营商，5400 万用户），意大利电信（Telecom Italia，意大利主要的移动运营商，3400 万用户），西班牙 Telefónica（在欧洲和拉美有 1.5 亿用户），T-Mobile（德意志电信旗下公司，在美国和欧洲有 1.1 亿用户）。

2. 半导体公司

Audience Corp（声音处理器公司），Broadcom Corp（无线半导体主要提供商），英特尔（Intel），Marvell Technology Group，Nvidia（图形处理器公司），SiRF（GPS 技术提供商），Synaptics（手

机用户界面技术)，德州仪器（Texas Instruments），高通（Qualcomm），惠普 HP（Hewlett-Packard Development Company, L.P）。

3. 软件公司

Aplix, Ascender, eBay 的 Skype, Esmertec, Living Image, NMS Communications, Noser Engineering AG, Nuance Communications, PacketVideo, SkyPop, Sonix Network, TAT-The Astonishing Tribe, Wind River Systems。

1.2.3 实行奖励机制

安卓（Android）为了提高程序员的开发积极性，不但为其提供了一流硬件的设置，还提供了一流的软件服务。并且采取了振奋人心的奖励机制，定期召开比赛，创意和应用夺魁者将会得到重奖。

1. 开发 Android 平台的应用

在 Android 平台上，程序员可以开发出各式各样的应用。其中 Android 应用项目是通过 Java 语言开发的，只要具备 Java 开发基础，就能很快地上手并掌握。作为单独的 Android 开发，Java 编程门槛并不高，即使没有编程经验的门外汉，也可以在突击学习 Java 之后而学会 Android。另外，Android 完全支持 2D、3D 和数据库，并且和浏览器实现了集成。所以通过 Android 平台，程序员可以迅速、高效地开发出绚丽多彩的应用，例如常见的工具、管理、互联网和游戏等。

2. 奖金丰厚的 Android 大赛

为了吸引更多的用户使用 Android 开发，已经成功举办了奖金为 1000 万美元的开发者的竞赛。鼓励开发人员创建出有创意的实用软件。这种大赛对于开发人员来说，不但能练习自己的开发水平，并且有机会获得高额的奖金。

3. 在 Android Market 上获取收益

Android Market 地址是 <http://www.Android.com/market/>。为了能让 Android 平台吸引更多的关注，谷歌开发了自己的 Android 软件下载店 Android Market，允许开发人员将应用程序在上面发布，也允许 Android 用户随意下载获取自己喜欢的程序。开发者需要申请开发者账号，有账号后才能将自己开发的作品上传到 Android Market，并且可以对自己的软件进行定价。只要开发的软件作品足够吸引人，就可以获得好的金钱回报。

1.3 认识驱动

驱动是硬件和软件之间的媒介和载体，是计算机等电子产品运行的根本。生活中总会遇到这样的场景：买了一个新 USB 鼠标，插在电脑上后会提示安装新的驱动；买了一台新的打印机，也需要安装驱动后才能使用。驱动含有推动和发动之意，计算机领域中的驱动也含有推动之意。

当我们在电脑中安装新硬件时，总会被要求放入“这种硬件的驱动程序”。其实在 Windows 系

统中,在安装主板、光驱、显卡、声卡这些硬件产品时都对应着一套完整的驱动程序。如果需要外接别的硬件设备,也还需要安装相应的驱动程序,例如外接游戏硬件要安装手柄、方向盘、摇杆、跳舞毯等的驱动程序,外接打印机要安装打印机驱动程序,上网或接入局域网要安装网卡、Modem 甚至 ISDN、ADSL 的驱动程序。

和 Windows 系统一样,在 Android 手机中也经常需要使用一些外部硬件设备,例如蓝牙耳机、存储卡和摄像头。要想使用这些外部辅助设备,也需要安装对应的驱动程序。驱动程序是添加到操作系统中的一段代码,通常这段代码比较简短,但是在里面包含了和硬件相关的设备信息。有了这些信息,计算机就可以与设备进行通信,从而使用这些硬件。驱动程序是硬件厂商根据操作系统编写的配置文件,可以说没有驱动程序,计算机中的硬件就无法工作。操作系统不同,对应的硬件驱动程序也不同。硬件厂商为了保证硬件的兼容性及增强硬件的功能,会不断更新、升级驱动程序,例如显卡芯片公司 Nvidia 平均每个月会升级驱动程序 2 到 3 次。

驱动程序是硬件的一个构成部分,当我们安装新的硬件时,也必须安装对应的驱动程序。凡是安装一个原本不属于我们电脑中或手机中的硬件设备时,系统就会要求你安装驱动程序,将新的硬件与电脑或手机系统连接起来。驱动程序在此扮演了一个沟通的角色,负责把硬件的功能告诉电脑系统,并且也将系统的指令传达给硬件,让它开始工作。

手机中的驱动和电脑中的驱动一样,当在手机中使用数据线、蓝牙、红外等连接方式连接电脑时,在一般情况下需要驱动程序。而且在一部分手机中,通过数据线、蓝牙、红外方式连接电脑后还需要软件才能传输数据到电脑,或者传输数据到手机。此时可以使用购买手机时的随机光盘中的驱动程序解决问题,驱动程序也可以在手机网站或论坛上下载。

14 分析 Android 架构

Android 作为一个移动设备的平台,其软件层次结构包括操作系统(OS)、中间件(MiddleWare)和应用程序(Application)。根据 Android 的软件框图,其软件层次结构自下而上分为以下 4 层。

- (1) 操作系统层(OS)。
- (2) 各种库(Library)和 Android 运行环境(RunTime)。
- (3) 应用程序框架(Application Framework)。
- (4) 应用程序(Application)。

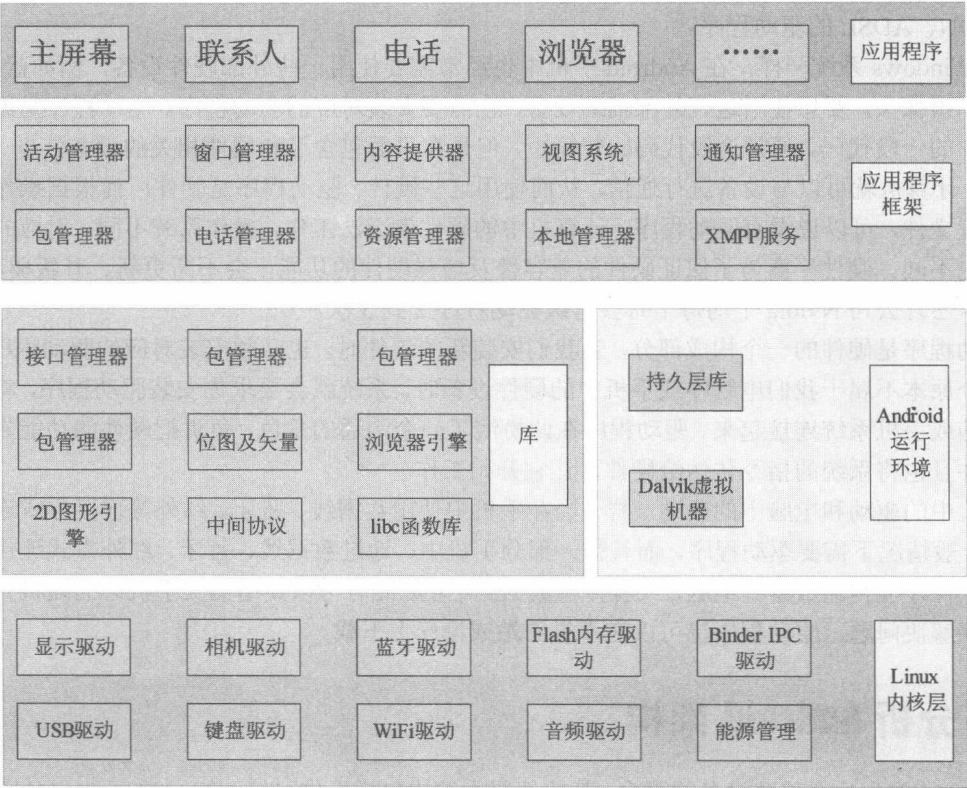
上述各个层的具体结构如图 1-1 所示。

1.4.1 操作系统层(OS)

Android 使用 Linux2.6 作为操作系统, Linux2.6 是一种标准的技术, Linux 也是一个开放的操作系统。Android 对操作系统的使用包括核心和驱动程序两部分, Android 的 Linux 核心为标准的 Linux2.6 内核, Android 更多的是需要一些与移动设备相关的驱动程序。主要的驱动如下所示。

- 显示驱动(Display Driver): 常用基于 Linux 的帧缓冲(Frame Buffer)驱动。
- Flash 内存驱动(Flash Memory Driver): 是基于 MTD 的 Flash 驱动程序。
- 照相机驱动(Camera Driver): 常用基于 Linux 的 v4l (Video for Linux) 驱动。

- 音频驱动 (Audio Driver): 常用基于 ALSA (Advanced Linux Sound Architecture, 高级 Linux 声音体系) 驱动。



▲图 1-1 Android 操作系统的组件结构图

- WiFi 驱动 (Camera Driver): 基于 IEEE 801.31 标准的驱动程序。
- 键盘驱动 (KeyBoard Driver): 作为输入设备的键盘驱动。
- 蓝牙驱动 (Bluetooth Driver): 基于 IEEE 801.35.1 标准的无线传输技术。
- Binder IPC 驱动: Android 一个特殊的驱动程序，具有单独的设备节点，提供进程间通信的功能。
- Power Management (能源管理): 管理电池电量等信息。

1.4.2 各种库和 Android 运行环境

本层次对应一般嵌入式系统，相当于中间件层次。Android 的本层次分成两个部分，一个是各种库 (Library)，另一个是 Android 运行环境 (RunTime)。本层的内容大多是使用 C++ 实现的。其中包含的各种库如下。

- C 库: C 语言的标准库，也是系统中一个最为底层的库，C 库是通过 Linux 的系统调用来实现的。