揭示各种驱动的具体实现过程　分享驱动开发的移植技巧

Broadview
www.broadview.com.cn

Android

底层开发技术实战详解

——内核、移植和驱动　第2版

王振丽 等编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Android移动开发技术丛书

Android 移动开发技术丛书

Android 底层开发技术实战详解
——内核、移植和驱动
（第 2 版）

王振丽　等编著

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

内 容 简 介

本书从底层原理开始讲起，结合真实的案例向读者详细介绍了 Android 内核、移植和驱动开发的整个流程。全书分为 21 章，依次讲解驱动移植的必要性，何为 HAL 层深入分析，Goldfish、MSM、OMAP 内核和驱动解析，显示系统、输入系统、振动器系统、音频系统、视频输出系统的驱动，OpenMax 多媒体和多媒体插件框架，传感器，照相机，Wi-Fi，蓝牙和 GPS，电话系统，时钟系统，USB Gadget 驱动，Lights 光系统和 Battery 电池系统等。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 源码进行了剖析。

本书适合 Android 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。

图书在版编目（CIP）数据

Android 底层开发技术实战详解：内核、移植和驱动 / 王振丽等编著. —2 版. —北京：电子工业出版社，2015.3
（Android 移动开发技术丛书）
ISBN 978-7-121-25441-3

Ⅰ. ①A… Ⅱ. ①王… 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（2015）第 012300 号

责任编辑：张月萍
印　　刷：北京京师印务有限公司
装　　订：北京京师印务有限公司
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：787×1092　1/16　　印张：37.00　　字数：898 千字
版　　次：2012 年 8 月第 1 版
　　2015 年 3 月第 2 版
印　　次：2015 年 3 月第 1 次印刷
印　　数：3500 册　　定价：79.00 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。
质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件到 dbqq@phei.com.cn。
服务热线：（010）88258888。

前　言

随着 3G/4G 的到来，无线带宽越来越高，使得更多内容丰富的应用程序装入手机成为可能，如视频通话、视频点播、移动互联网冲浪和内容分享等。为了承载这些数据应用及快速部署，手机功能将会越来越智能，越来越开放。为了实现这些需求，必须有一个优秀的开发平台来支持，在此由谷歌（Google）公司发起的 OHA 联盟走在了业界的前列，2007 年 11 月推出了开放的 Android 平台，任何公司及个人都可以免费获取源代码及开发 SDK。由于其开放性和优异性，Android 平台得到了业界广泛的支持，其中包括各大手机厂商和知名的移动运营商等。继 2008 年 9 月第一款基于 Android 平台的手机 G1 发布之后，三星、摩托罗拉、索尼爱立信、LG 等主流手机制造商都推出了自己的 Android 平台手机。在 2011 年年底，Android 超越了塞班和 iOS，雄踞智能手机市场占有率榜首的位置。根据国际数据公司（IDC）5 月公布的新数据，在 2013 年第一季度，Android 和 iOS 系统占的装机量到所有智能手机出货量的 92.3%。在 2013 年头三个月，安装 Android 系统的新智能手机数量跃升至 1.621 亿部，大大超过去年同期的 9030 万部。这意味着，在销往世界各地的所有新智能手机中，谷歌的移动操作系统的市场占有率已经达到 75%，与第一季度的 59.1%相比有显著提高。

毕竟 Android 平台被推出的时间才短短六七年，了解 Android 平台软件开发技术的程序员还不多，如何迅速地推广和普及 Android 平台软件开发技术，让越来越多的人参与到 Android 应用的开发中，是整个产业链都在关注的一个话题。为了帮助开发者更快地进入 Android 开发行列，笔者特意精心编写了本书。本书系统地讲解了 Android 底层驱动开发和移植的基本知识，图文并茂地帮助读者学习和掌握各种驱动的开发常识，详细讲解了 Android 源代码的方方面面。

从技术角度而言，Android 是一种融入了全部 Web 应用的平台。随着版本的更新，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，从基本的网页浏览到现在的 HTML 5，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、C、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。

本书内容

在本书的内容中，详细讲解了 Android 底层技术和驱动开发的基本知识。本书内容新颖、知识全面、讲解详细，全书分为 21 章，具体内容分布如下。

章	主要内容
第 1 章	什么是驱动以及 Linux 内核源代码简单剖析
第 2 章	搭建 Linux 开发环境，分析及编译 Android 源代码，在 Linux 环境下运行模拟器

（续表）

章	主要内容
第 3 章	Android 移植的内容、驱动开发所要完成的任务，三种类型的驱动程序
第 4 章	传感器在 HAL 层的表现，HAL 层的源代码与移植
第 5 章	Goldfish 下的 staging、Ashmen、Pmem、Alarm 和 Android Paranoid 驱动
第 6 章	MSM 内核与 MSM 的移植
第 7 章	OMAP 内核与移植
第 8 章	显示系统的移植、调试与驱动程序实现
第 9 章	MSM 处理器和 OMAP 处理器平台中输入驱动的实现
第 10 章	振动器的系统结构与移植
第 11 章	音频系统的层次、移植与不同平台下的实现
第 12 章	视频输出系统 Overlay 的分析、实现、调用
第 13 章	OpenMax 多媒体框架的层次和实现
第 14 章	OpenCore 引擎和 Stagefright 引擎的代码结构与扩展
第 15 章	传感器系统的结构、移植与实现
第 16 章	照相机系统的结构、移植与实现
第 17 章	Wi-Fi 系统、蓝牙系统和定位系统的移植
第 18 章	开发电话系统
第 19 章	分析时钟系统驱动，实现时钟驱动程序的开发、移植和模拟器测试
第 20 章	分析 USB Gadget 驱动，依次讲解分析软件结构、层次整合和 USB 设备枚举
第 21 章	Lights 光系统和 Battery 电池系统

全书内容都采用了理论加实践的教学方法，每个实例先提出制作思路及包含知识点，在实例最后补充总结知识点并出题帮助读者举一反三。

本书特色

本书内容相当丰富，实例内容覆盖全面，满足 Android 技术人员成长道路上的方方面面。编者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，以完善自身的知识和技能结构。在内容的编写上，本书具有以下特色。

（1）结构合理

从读者的实际需要出发，科学安排知识结构，内容由浅入深，叙述清楚，具有很强的知识性和实用性，反映了当前 Android 技术的发展和应用水平。同时全书精心筛选的最具代表性、读者最关心典型知识点，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；每个部分既相互连贯又自成体系，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针对性的学习。

（3）实用性强

本书彻底摒弃枯燥的理论和简单的操作，注重实用性和可操作性，详细讲解了各个部分的源代码知识，使用户在掌握相关操作技能的同时，还能学到相应的基础知识。

本书作者

参加本书编写的人员有：怀志和、王振丽、林慧晶、程娟、王文忠、陈强、何子夜、李天祥、万春朝、扶松柏、邓才兵、周锐、王东华、管西京、薛小龙。本团队在编写过程中由于时间和水平所限，书中难免有不足之处。如有纰漏和不尽如人意之处，诚请读者提出意见或建议，以便修订并使之更臻完善。另外，为了更好地为读者服务，本书专门提供了技术支持网站www.topchuban.com，欢迎读者光临论坛，无论是书中的质疑，还是学习过程中的疑惑，本团队将尽力为大家解答。

编者

2014 年 11 月

目　录

第 1 章　Android 底层开发基础

Android 是一种手机开发平台，它是建立在 Linux 基础之上的、能够迅速建立手机软件的解决方案。Android 外形比较简单，但功能十分强大，已经成为当前一个热点，并且是软件行业的一股新兴力量。本章将简单介绍 Android 的发展历程和背景，讲解在进行驱动开发之前所要做的工作，为读者学习后面的高级知识打下基础。

1.1　什么是驱动

生活中总会遇到这样的场景：买了一个新的 USB 鼠标，插在计算机上后会提示安装新的驱动；买了一台新的打印机，也提示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）。本节将简要讲解计算机领域中驱动的基本知识。

1.1.1　驱动程序的魅力

人们在安装新硬件时，总会被要求放入“这种硬件的驱动程序”。这是很令初学者头痛的事情，他们往往不是找不到驱动程序的盘片，就是找不到文件的位置，或是根本不知道什么是驱动程序。比如安装打印机这种普通的硬件设备，并不是把连接线接上就能完成的。如果将数据线连接到计算机后就开始使用，系统会提示找不到驱动程序，这时应该怎么办呢？对于菜鸟来说，即使参考说明书也未必能顺利安装。在不能成功使用硬件的原因中，70%以上是因为没有安装驱动程序。

其实在 Windows 系统中，安装主板、光驱、显卡、声卡这些硬件产品都对应一套完整的驱动程序。如果需要外接其他硬件设备，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 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、外部存储卡和摄像头等。要想使用这些外部辅助设备，需要安装对应的驱动程序。驱动程序是添加在操作系统中的一段代码，虽然这段代码比较简短，但是其中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，从而可以使用这些硬件。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，当安装新的硬件时，必须安装对应的驱动程序。凡是安装一个原本不属于计算机或手机中的硬件设备时，系统就会要求安装驱动程序，将新的硬件与计算机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计算机系统，并且将系统的指令传达给硬件，让它开始工作。

1.1.2 手机中的驱动程序

有的手机和计算机不能直接连接，必须用手机自带的磁盘驱动一下，其实就是安装了一个读取手机内存信息的程序。我们可以去网络中寻找安装驱动，只需在网上搜索机型和驱动就可以。或者把手机用 USB 线连到计算机上，会弹出一个对话框，选择让计算机自己在互联网上搜索驱动程序即可。

如果在手机中使用数据线、蓝牙、红外等连接方式连接电脑，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计算机，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，也可以在手机网站或论坛下载。

如果是通过串口连接计算机的，一般不需要驱动程序，但是此时会需要用软件来实现和手机的连接。在手机附赠光盘中通常会有这样的软件，或者可以去网站、论坛下载。

 1.2 开源还是不开源的问题

本节将简单谈一谈开源和不开源的问题。我们都知道 Android 是基于 Linux 的，因为 Linux 是开源的，Android 也号称开源，所以一经推出就受到广大程序员和手机厂商的青睐。但是本节的题目为什么是“开源还是不开源的问题”呢？这需要从 Android 的发展历史谈起。

1.2.1 雾里看花的开源

在 Android 刚被推出的时候，只能用 Java 语言开发应用程序，这就需要所有的应用程序都运行在一个巨大的虚拟机上。2009 年 6 月，Android 发布了 NDK 工具包，这样就可以支持 C/C++ 语言编程，但是性能不如 SKD 工具包中的 Java 语言。

2010 年 2 月，在开源界发生了一件大事。Linux Kernel 的维护者 Greg Kroah-Hartman 宣布，将 Android 代码从 Linux Kernel 代码库中删除。此事对于普通用户可能并没有什么影响，但对于开发者来说，尤其是对于开源社区的开发者来说，算是一颗重磅炸弹。消息公布以后，外界普遍觉得惊讶和可惜。好不容易才有了一个如此受欢迎的开源手机系统，应该齐心协力、共同开发才对，为什么要“窝里斗”呢？到底是什么矛盾，使得 Linux Kernel 小组剔除 Android 代码呢？

从 Linux 2.6.33 版本开始，Google 智能手机操作系统 Android 核心代码被全部删除。这是因为提倡开源的 Android 在 Linux 面前使用了雾里看花的把戏，它修改了 Kernel 内核，但是又不提供修改的细节，这相当于自己搞了一个封闭的系统。尽管 Android 取得了空前的成功，但是 Google 也放弃了构建一个真正开源的手机系统的机会，从而不能获得由全世界程序员提供智

慧、分享代码和推动创新的好处。由此可见，是因为 Android 不真正开源，所以才被从 Linux 体系中删除。

Android 与 Ubuntu、Debian、Red Hat 等传统的 Linux 发行版相比，只有系统的底层结构是一样的，其他在 Android 中都不一样，尤其是程序员的编写接口是完全不同的。所以必须重新写 Android 应用程序后才能使用，现存的 Linux 程序无法移植过去。由此可见，Android 是一种全新的系统，它与 Linux 距离很远。

1.2.2　从为什么选择 Java 谈为什么不开源驱动程序

1. Java 的好处

Android 很好地解决了长期令手机制造商头痛不已的问题：在业界缺乏一个开源的 Java 虚拟机和统一的应用程序接口。使用 Android 后，程序员只要编写一次程序就可以用在各种手机硬件平台之上。这就是 Android 应用程序使用 Java 语言开发的原因，因为如果不这样做，就无法让程序实现和硬件无关。

可能很多熟知 Linux 的读者会反问：传统的 Linux 系统也不依赖特定的硬件，只要把源代码根据不同的平台分别编译，同一个程序就可以在不同的硬件架构、不同的 Linux 发行版中使用。那么 Android 只采用 Kernel、只允许用 Java 编程的真正原因到底是什么呢？

2. 为什么驱动不开源

Linux Kernel 的版权是 GPL。在此版本下，硬件厂商都希望自己的硬件能在 Linux Kernel 下运行，此时就必须使用驱动程序。但是如果把驱动程序的源代码公开，就等于公开硬件规格，这是广大硬件厂商所不能接受的。所以硬件厂商只提供编好的驱动程序，而不提供原始代码。

Android 的重点是商业应用，为了解决上述驱动开源的问题，Google 采用了自己的方法来绕过这个问题。Google 把驱动程序移到“userspace”中，即让驱动程序在 Linux Kernel 上面运行，而不是一起运行，这样就可以避开 GPL 规则。然后在 Kernel 上开一个小门，让本来不能直接控制硬件的“userspace”程序也可以碰得到，此时只需公布这个开的“小门”程序源代码即可。由此可见，Google 在 Kernel 和应用程序之间设计了一个中间层，这样既不违反 GPL 许可，又能不让外界看到厂商的硬件驱动和应用程序的源代码。

3. 带来的问题

但是 Google 的上述做法随之带来了一个问题，Kernel 和 Android 采取不同的许可证，Kernel 采用 GPL 许可证，而 Android 采用 Apache Software License（简称 ASL）许可证。在 GPL 许可证中规定，对源代码的任何修改都必须开源，所以 Android 需要开源，因为它修改了 Kernel。而在 ASL 许可证中规定，用户可以随意使用源代码而不必开源，所以建立在 Android 之上的硬件驱动和应用程序都可以保持封闭。这种封闭得到了更多硬件厂商的支持，Google 特意修改了 Kernel，使得原本应该包括在 Kernel 中的某些功能被转移到“userspace”中，所以就避开了开源。

4. 影响

Google 的上述行为有利于推广 Android，并且可以吸引更多厂商和软件开发商的加入，但是同时也宣布放弃了构建一个真正开源的手机系统的机会。所有为 Android 写的硬件驱动都不能合并到 Kernel 中，因为它们只在 Google 的代码里才有效，而在 Kernel 里根本没法用。

1.2.3 对驱动开发者来说是一把双刃剑

正因为所有为 Android 写的硬件驱动都不能合并到 Kernel 中，这些驱动程序只能在 Google 代码中有效，而在 Kernel 中根本没法用，所以 Google 从不向 Kernel 提交大量的硬件驱动程序和平台源代码。

硬件厂商都不开源驱动代码，为生存在 Android 底层的开发人员，特别是从事驱动开发的人员，带来了巨大的就业机会。开发人员可以为硬件厂商开发不开源的驱动程序从而获得报酬。随着 Android 的异常火爆，市面上有很多企业在招聘 Android 驱动开发人员。由此可见，驱动的不开源既为我们的学习带来了难题，也为就业机会增加了砝码，是一把双刃剑！

 1.3 Android 和 Linux

Linux 是类 UNIX 计算机操作系统的统称。Linux 操作系统的内核名字也是“Linux”。Linux 操作系统是自由软件和开放源代码发展中最著名的例子。2007 年，基于 Linux 的 Android 系统横空出世，Android 将是未来智能手机发展的趋势。自从 2008 年 9 月 22 日，美国运营商 T-Mobile USA 在纽约正式发布了第一款基于 Android 的手机后，更多的移动设备厂商看到了 Android 的光明前景，并纷纷加入其中。随着 Android 手机的普及，Android 应用的需求越来越大，这是一个潜力巨大的市场，吸引无数软件开发厂商和开发者投身其中。本节将简要讲解 Android 和 Linux 之间的关系。

1.3.1 Linux 简介

众所周知，在传统计算机操作系统领域，Windows、Netware 和 UNIX 一直占据主导地位。但是最近几年发生了一些变化，以自由标榜自己的 Linux 越来越显示出其咄咄逼人的气势，日益成为一个令人生畏的对手。Linux 正在发起一场产业革命，逐渐吞噬着传统操作系统的市场份额。请回头看一看我们平常使用的技术：嵌入式系统、移动计算、移动互联网工具、服务器、超级计算。在几乎每个技术领域，Linux 都展现出作为未来主导平台的势头。随着 SaaS、云计算、虚拟化、移动平台、Web 2.0 等新兴技术的发展，Linux 事业正面临着巨大发展机遇。主要体现在如下三个方面。

1. 向企业级核心应用迈进

Linux 的采用已由网络领域逐步转向了关键业务应用，企业关键任务成为 IBM 的增长领域，比如 ERP 软件。同时，随着 IT 决策逐步从 IT 主管下放给 IT 管理人员，这些管理者对 Linux 显示了强烈的支持，但同时对安全、可用性与服务提出了更高的需求。尽管很多用户仍将 UNIX 视为关键任务的平台，但随着 Linux 开发者逐步缩小两者的功能性差距，越来越多的用户开始将关键业务部署在 Linux 之上。

2. Linux 将主导移动平台

Linux 进入移动终端操作系统后，很快就以其开放源代码的优势吸引了越来越多的终端厂商和运营商，关注它的包括摩托罗拉等知名的厂商。与其他操作系统相比，Linux 是个后来者，

但 Linux 具有其他操作系统无法比拟的优势。其一，Linux 具有开放的源代码，能够大大降低成本。其二，既满足了手机制造商根据实际情况有针对性地开发自己的 Linux 手机操作系统的要求，又吸引了众多软件开发商对内容应用软件的开发，丰富了第三方应用。

3. 新技术为 Linux 加速

在目前的企业级计算领域，云计算、SaaS、虚拟化是热门技术话题。在这些领域，Linux 同样大有可为。云计算将全部使用 Linux，Linux 也是一款未来运行数据中心虚拟机的理想操作系统。

1.3.2　Android 和 Linux 的关系

在了解 Linux 和 Android 的关系之前，需要先明确如下三点。

- Android 采用 Linux 作为内核。
- Android 对 Linux 内核做了修改，目的是适应在移动设备上使用。
- Android 开始是作为 Linux 的一个分支，后来由于无法并入 Linux 的主开发树，已被 Linux Kernel 小组从开发树中删除。

1. Android 是继承于 Linux 的

Android 是在 Linux 内核基础之上运行的，提供的核心系统服务包括安全、内存管理、进程管理、网络组和驱动模型等内容。内核部分还相当于一个介于硬件层和系统中其他软件组之间的一个抽象层次。但是严格来说它不算是 Linux 操作系统。

因为 Android 内核是由标准的 Linux 内核修改而来的，所以继承了 Linux 内核的诸多优点，保留了 Linux 内核的主题架构。同时 Android 按照移动设备的需求，在文件系统、内存管理、进程间通信机制和电源管理方面进行了修改，添加了相关的驱动程序和必要的新功能。但是和其他精简的 Linux 系统相比（比如 uClinux），Android 在很大程度上保留了 Linux 的基本架构，因此 Android 的应用性和扩展性更强。

2. Android 和 Linux 内核的区别

Android 系统的系统层面的底层是 Linux，中间加上了一个叫做 Dalvik 的 Java 虚拟机，表面层上面是 Android 运行库。每个 Android 应用都运行在自己的进程上，享有 Dalvik 虚拟机为它分配的专有实例。为了支持多个虚拟机在同一个设备上高效运行，Dalvik 被改写过。

Dalvik 虚拟机执行的是 Dalvik 格式的可执行文件（.dex）。该格式经过优化，将内存耗用降到最低。Java 编译器将 Java 源文件转为.class 文件，.class 文件又被内置的 dx 工具转化为.dex 格式文件，这种文件在 Dalvik 虚拟机上注册并运行。

Android 系统的应用软件都是运行在 Dalvik 之上的 Java 软件，而 Dalvik 是运行在 Linux 中的，在一些底层功能——比如线程和低内存管理方面，Dalvik 虚拟机是依赖 Linux 内核的。由此可见，Android 是运行在 Linux 之上的操作系统，但是它本身不能算是 Linux 的某个版本。

Android 内核和 Linux 内核的差别主要体现在 11 个方面，接下来将一一简要介绍。

（1）Android Binder

Android Binder 是基于 OpenBinder 框架的一个驱动，用于提供 Android 平台的进程间通信（IPC，Inter-Process Communication）。原来的 Linux 系统上层应用的进程间通信主要是 D-Bus

（Desktop Bus），采用消息总线的方式进行 IPC。

其源代码位于 drivers/staging/android/binder.c。

（2）Android 电源管理（PM）

Android 电源管理是一个基于标准 Linux 电源管理系统的轻量级 Android 电源管理驱动，针对嵌入式设备做了很多优化。利用锁和定时器来切换系统状态，控制设备在不同状态下的功耗，以达到节能的目的。

其源代码分别位于如下文件中。

```
kernel/power/earlysuspend.c
kernel/power/consoleearlysuspend.c
kernel/power/fbearlysuspend.c
kernel/power/wakelock.c
kernel/power/userwakelock.c
```

（3）低内存管理器（Low Memory Killer）

Android 中的低内存管理器和 Linux 标准的 OOM（Out Of Memory）相比，其机制更加灵活，它可以根据需要杀死进程来释放需要的内存。Low Memory Killer 的代码非常简单，里面的关键是函数 lowmem_shrinker()，作为一个模块在初始化时调用 register_shrinke 注册一个 lowmem_shrinker，它会被 vm 在内存紧张的情况下调用。lowmem_shrinker 完成具体操作。简单说就是寻找一个最合适的进程杀死，从而释放它占用的内存。

其源代码位于 drivers/staging/android/lowmemorykiller.c。

（4）匿名共享内存（Ashmem）

匿名共享内存为进程间提供大块共享内存，同时为内核提供回收和管理这个内存的机制。如果一个程序尝试访问 Kernel 释放的一个共享内存块，它将会收到一个错误提示，然后重新分配内存并重载数据。

其源代码位于 mm/ashmem.c。

（5）Android PMEM（Physical）

PMEM 用于向用户空间提供连续的物理内存区域，DSP 和某些设备只能工作在连续的物理内存上。驱动中提供了 mmap、open、release 和 ioctl 等接口。

源代码位于 drivers/misc/pmem.c。

（6）Android Logger

Android Logger 是一个轻量级的日志设备，用于抓取 Android 系统的各种日志，是 Linux 所没有的。

其源代码位于 drivers/staging/android/logger.c。

（7）Android Alarm

Android Alarm 提供了一个定时器，用于把设备从睡眠状态唤醒，同时它还提供了一个即使在设备睡眠时也会运行的时钟基准。

其源代码位于如下文件。

```
drivers/rtc/alarm.c
drivers/rtc/alarm-dev.c
```