

# Android 的发展历程及其与其他 UNIX 系统的关系

张菁芸 2014210785

## 一. Android 的发展历程

Android 是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由 Google 公司和开放手机联盟领导及开发。尚未有统一中文名称，中国大陆地区较多人使用“安卓”或“安致”。

- ✓ 2003 年 10 月，Andy Rubin 等人创建 Android 公司，并组建 Android 团队。
- ✓ 2005 年 8 月 17 日，Google 低调收购了成立仅 22 个月的高科技企业 Android 及其团队。安迪鲁宾成为 Google 公司工程部副总裁，继续负责 Android 项目。
- ✓ 2007 年 11 月 5 日，谷歌公司正式向外界展示了这款名为 Android 的操作系统，并且在这天谷歌宣布建立一个全球性的联盟组织，该组织由 34 家手机制造商、软件开发商、电信运营商以及芯片制造商共同组成，并与 84 家硬件制造商、软件开发商及电信营运商组成开放手持设备联盟（Open Handset Alliance）来共同研发改良 Android 系统，这一联盟将支持谷歌发布的手机操作系统以及应用软件，Google 以 Apache 免费开源许可证的授权方式，发布了 Android 的源代码。
- ✓ 2008 年，在 GoogleI/O 大会上，谷歌提出了 AndroidHAL 架构图，在同年 8 月 18 号，Android 获得了美国联邦通信委员会（FCC）的批准，在 2008 年 9 月，谷歌正式发布了 Android 1.0 系统，这也是 Android 系统最早的版本。
- ✓ 2009 年 4 月，谷歌正式推出了 Android 1.5 这款手机，从 Android 1.5 版本开始，谷歌开始将 Android 的版本以甜品的名字命名，Android 1.5 命名为 Cupcake（纸杯蛋糕）。该系统与 Android 1.0 相比有了很大的改进。
- ✓ 2009 年 9 月份，谷歌发布了 Android 1.6 的正式版，并且推出了搭载 Android 1.6 正式版的手机 HTC Hero（G3），凭借着出色的外观设计以及全新的 Android 1.6 操作系统，HTC Hero（G3）成为当时全球最受欢迎的手机。Android 1.6 也有一个有趣的甜品名称，它被称为 Donut（甜甜圈）。
- ✓ 2010 年 2 月份，Linux 内核开发者 Greg Kroah-Hartman 将 Android 的驱动程序从 Linux 内核“状态树”（“staging tree”）上除去，从此，Android 与 Linux 开发主流将分道扬镳。在同年 5 月份，谷歌正式发布了 Android 2.2 操作系统。谷歌将 Android 2.2 操作系统命名为 Froyo，翻译完名为冻酸奶。
- ✓ 2010 年 10 月份，谷歌宣布 Android 系统达到了第一个里程碑，即电子市场上获得官方数字认证的 Android 应用数量已经达到了 10 万个，Android 系统的应用增长非常迅速。在 2010 年 12 月，谷歌正式发布了 Android 2.3 操作系统 Gingerbread（姜饼）。
- ✓ 2011 年 1 月，谷歌称每日的 Android 设备新用户数量达到了 30 万部，到 2011 年 7 月，这个数字增长到 55 万部，而 Android 系统设备的用户总数达到了 1.35 亿，Android 系统已经成为智能手机领域占有量最高的系统。

- ✓ 2011 年 8 月 2 日，Android 手机已占据全球智能机市场 48% 的份额，并在亚太地区市场占据统治地位，终结了 Symbian（塞班系统）的霸主地位，跃居全球第一。
- ✓ 2011 年 9 月份，Android 系统的应用数目已经达到了 48 万，而在智能手机市场，Android 系统的占有率已经达到了 43%。继续在排在移动操作系统首位。谷歌将会发布全新的 Android 4.0 操作系统，这款系统被谷歌命名为 Ice Cream Sandwich（冰激凌三明治）。
- ✓ 2012 年 1 月 6 日，谷歌 Android Market 已有 10 万开发者推出超过 40 万活跃的应用，大多数的应用程序为免费。Android Market 应用程序商店目录在新年首周末突破 40 万基准，距离突破 30 万应用仅 4 个月。在 2011 年早些时候，Android Market 从 20 万增加到 30 万应用也花了四个月。[6]
- ✓ 2013 年 11 月 1 日，Android4.4 正式发布，从具体功能上讲，Android4.4 提供了各种实用小功能，新的 Android 系统更智能，添加更多的 Emoji 表情图案，UI 的改进也更现代，如全新的 HelloiOS7 半透明效果。

## 二. Android 与其他 UNIX 系统的关系

虽然 Android 基于 Linux 内核，但是它与 Linux 之间还是有很大的差别。成熟的操作系统有很多，比如 Unix 系统。但是 Android 为什么选择采用 Linux 内核呢？这就与 Linux 的一些特性有关了，比如：

- (1) 强大的内存管理和进程管理方案
- (2) 基于权限的安全模式
- (3) 支持共享库
- (4) 经过认证的驱动模型
- (5) Linux 本身就是开源项目

但是 Android 不是 Linux。理由如下：

### 1、它没有本地窗口系统

什么是本地窗口系统呢？本地窗口系统是指 GNU/Linux 上的 X 窗口系统，或者 Mac OS 的 Quartz 等。不同的操作系统的窗口系统可能不一样，Android 并没有使用（也不需要）Linux 的 X 窗口系统，这是 Android 不是 Linux 的一个基本原因。

### 2、它没有 glibc 支持

由于 Android 最初用于一些便携的移动设备上，所以，可能出于效率等方面的考虑，Android 并没有采用 glibc 作为 C 库，而是 Google 自己开发了一套 Bionic Libc 来代替 glibc。

### 3、它并不包括一整套标准的 Linux 使用程序

Android 并没有完全照搬 Linux 系统的内核，除了修正部分 Linux 的 Bug 之外，还增加了不少内容，比如：它基于 ARM 构架增加的 Gold-Fish 平台，以及 yaffs2 FLASH 文件系统等。

#### 4、Android 专有的驱动程序

除了上面这些不同点之外，Android 还对 Linux 设备驱动进行了增强，主要如下所示。

(1) Android Binder 基于 OpenBinder 框架的一个驱动，用于提供 Android 平台的进程间通信( InterProcess Communication, IPC)功能。源代码位于

drivers/staging/android/binder.c。

(2) Android 电源管理(PM) 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。源代码位于：

I kernel/power/earlysuspend.c

II kernel/power/consoleearlysuspend.c

III kernel/power/fbearlysuspend.c

IV kernel/power/wakelock.c

V kernel/power/userwakelock.c

(3)低内存管理器(Low Memory Killer) 比 Linux 的标准的 OOM (Out Of Memory) 机制更加灵活，它可以根据需要杀死进程以释放需要的内存。源代码位于

drivers/staging/android/lowmemorykiller.c。

(4)匿名共享内存(Ashmem) 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。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。源代码位于 drivers/misc/pmem.c。

(6) Android Logger -个轻量级的日志设备，用于抓取 Android 系统的各种日志。源代码位于 drivers/staging/android/logger.c。

(7) 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。源代码位于 drivers rtc/alarm.c。

(8) USB Gadget 驱动一个基于标准 Linux USB gadget 驱动框架的设备驱动，Android 的 USB 驱动是基于 gadget 框架的。源代码位于 drivers/usb/gadget/。

(9) Android Ram Console 为了提供调试功能，Android 允许将调试日志信息写入一个被称为 RAM Console 的设备里，它是一个基于 RAM 的 Buffer。源代码位于 drivers/staging/android / ramL console.c。

(10) Android timed device 提供了对设备进行定时控制的功能，目前支持 vibrator 和 LED 设备。源代码位于 drivers/staging/android /timed\_output.c(timed\_gpio.c)。

(11) Yaffs2 文件系统 Android 采用 Yaffs2 作为 MTD nand flash 文件系统，源代码位于 fs/yaffs2/目录下。Yaffs2 是一个快速稳定的应用于 NAND 和 NOR Flash 的跨平台的嵌入式设备文件系统，同其他 Flash 文件系统相比，Yaffs2 能使用更小的内存

来保存其运行状态，因此它占用内存小。Yaffs2 的垃圾回收非常简单而且快速，因此能表现出更好的性能。Yaffs2 在大容量的 NAND Flash 上的性能表现尤为突出，非常适合大容量的 Flash 存储。上面这些要点足以说明 Android 不是 Linux。

1)Android Binder 基于 OpenBinder 框架的一个驱动，用于提供 Android 平台的进程间通信(InterProcess Communication, IPC)功能。源代码位于 drivers/staging/android/binder.c。

2)Android 电源管理(PM) 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。源代码位于：

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

如果给内核添加驱动也可以称之为不同的话？

3)低内存管理器(Low Memory Killer) 比 Linux 的标准的 OOM(Out Of Memory) 机制更加灵活，它可以根据需要杀死进程以释放需要的内存。源代码位于 drivers/staging/ android/lowmemorykiller.c。

4)匿名共享内存(Ashmem) 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。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。源代码位于 drivers/misc/pmem.c。

6)Android Logger 一个轻量级的日志设备，用于抓取 Android 系统的各种日志。源代码位于 drivers/staging/android/logger.c。

7)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。源代码位于 drivers rtc/alarm.c。

8)USB Gadget 驱动 一个基于标准 Linux USB gadget 驱动框架的设备驱动，Android 的 USB 驱动是基于 gadget 框架的。源代码位于 drivers/usb/gadget/。

9)Android Ram Console 为了提供调试功能，Android 允许将调试日志信息写入一个被称为 RAM Console 的设备里，它是一个基于 RAM 的 Buffer。源代码位于 drivers/staging/android / ram\_console.c。

10)Android timed device 提供了对设备进行定时控制的功能，目前支持 vibrator 和 LED 设备。源代码位于 drivers/staging/android /timed\_output.c(timed\_gpio.c)。

11)Yaffs2 文件系统 Android 采用 Yaffs2 作为 MTD nand flash 文件系统，源代码位于 fs/yaffs2/目录下。Yaffs2 是一个快速稳定的应用于 NAND 和 NOR Flash 的跨平台的嵌入式设备文件系统，同其他 Flash 文件系统相比，Yaffs2 能使用更小的内存来保存其运行状态，因此它占用内存小。Yaffs2 的垃圾回收非常简单而且快速，因此能表现出更好的性能。Yaffs2 在大容量的 NAND Flash 上的性能表现尤为突出，非常适合大容量的 Flash 存储。