

Android操作系统为什么成功

3180103772 张溢弛

摘要

本文从Android操作系统的基本概念出发，深入研究了Android操作系统的发展历程、系统架构和设计的独到之处，并探究了Android操作系统大获成功的原因，同时也为国产操作系统的开发之路提供一定的借鉴和思考。

一、Android操作系统概述

1.1 基本介绍

安卓(Android)操作系统是一种由美国的谷歌公司和开放手机联盟领导并开发的，基于Linux内核并且不包含GNU组件的开源操作系统，主要用于手机、平板电脑等移动设备中。安卓操作系统继承了Linux内核的宏内核结构，支持Apache License、GPL等软件许可，是一种底层基于C/C++，应用层基于Java和Kotlin的移动端操作系统，包含了操作系统、中间件、用户界面和应用软件等多个架构层级。

Android一词的本意是指“机器人”，最早出现于法国作家利尔亚当的科幻小说作品《未来夏娃》中，因此安卓操作系统的logo也是一个绿色的机器人，这款操作系统意境面世就引起了巨大的反响，在经历了数次版本迭代之后取得了巨大的成功。

1.2 发展历程

安卓操作系统最早是由Andy Rubin开发的手机操作系统，后来被谷歌收购。2007年11月，谷歌领导并建立了开放手机联盟并对安卓操作系统进行了改进，随后就以Apache作为开源许可证开源了安卓操作系统的源代码。

随后安卓操作系统进行了若干次的版本迭代，发行版本逐渐从V1.0升级到了V11.0，而在操作系统不断迭代升级的过程中，安卓操作系统也从最初单纯的手机操作系统扩展到了平板电脑、电视机、数码相机、游戏机和智能手表等移动终端上。安卓操作系统早在2011年就在市场份额上打败了塞班操作系统，并逐渐追赶上苹果公司的iOS操作系统

二、Android的系统总体架构

安卓操作系统的体系架构和其他大部分操作系统一样，采用了分层的架构，从架构图上来看，安卓操作系统主要分为四个层级架构，自顶向下分别是：应用程序层、应用程序框架层、系统运行时库和Linux内核层。

2.1 应用程序层(APP)

- 安卓操作系统会打包一系列重要的应用软件一起发布，比如通话、短信、日历、浏览器等等，这些应用程序(也叫做安卓APP)都是使用Java或者Kotlin语言进行编写的。
- 与此同时用户可以从应用商店/浏览器中下载其他的应用软件以扩展手机的功能，用户可以删除自己下载的应用程序，但不能删除一些非常重要的应用程序比如通话和短信，这是由用户的权限决定的，类似于Linux，安卓操作系统中也把用户分为root用户和普通用户等层级。

2.2 应用程序框架层

- 安卓应用程序的开发人员可以使用一系列的API框架来简化开发的过程，增加组件的重用和代码的可靠性，开发人员可以在完成开发之后发布其程序功能块以供其他开发者再次使用，但是在开发过程中开发人员必须遵循框架的安全性。

- 每个应用程序都包含了一系列的服务和系统，主要包括：
 - 丰富可扩展的视图：包含表、网格、文本框、按钮等基本组件，可以用来构建应用程序
 - 内容提供者：用于不同应用程序之间的数据交互和共享
 - 资源管理器：提供各类静态资源的访问方式，包括本地存储的文本、图片和各类布局文件
 - 通知管理器：使得应用程序可以在状态栏中显示各类提示信息
 - 活动管理器：用来管理应用程序的生命周期并且提供常用的导航回退功能

2.3 系统运行库和运行时环境

- 安卓操作系统的运行时库和运行时环境提供了一些C/C++库文件，可以被安卓系统的各类组件使用，通过在安卓应用程序框架中进行封装来实现各类业务逻辑
- 系统核心的C/C++库包括
 - 系统C库：从BSD继承来的标准C系统函数库libc
 - 媒体库：基于PacketVideo和OpenCORE，支持多种音视频的回放和录制，并且支持静态媒体文件的存储和编码，支持的编码格式包括JPG,PNG,MP3等等
 - Surface Manager：用于管理各个已经存在的子系统，并提供了2D和3D涂层的无缝融合
 - LibWebCore：是一个Web浏览器引擎，支持安卓浏览器和可嵌入式的web视图
- 安卓的运行时环境：
 - 安卓的应用程序都运行在安卓的运行时环境中，运行时环境主要由一系列中间件构成
 - 中间件是介于操作系统和应用程序之间的软件，安卓操作系统中的中间件主要分为两层：函数层和虚拟机。安卓操作系统改良了libc库开发出了Bionic，并且封装了Webkit浏览器引擎、OpenCORE多媒体框架、skia图形渲染引擎和SQLite数据库等一系列功能性的库和组件。
 - 安卓操作系统的中间件大多数采用Java实现，并且运行在特殊的Dalvik虚拟机上面，这是一种寄存器形态的Java虚拟机，变量存放在寄存器中，和普通的JVM相比，Dalvik并不运行Java的字节码，而是运行之中dex后缀的文件。

2.4 Linux内核层

安卓的内核是Linux Kernel，但并不是GNU/Linux，因为安卓操作系统移除了GNU/Linux的大部分支持功能，包括Cairo、X11、Glibc等等。

这是因为安卓作为一款商业应用，必须移除被GNU GPL授权证书所约束的部分，因此安卓在此基础上将驱动程序移植到了用户空间，使得Linux的驱动和内核彻底分开。

安卓的Linux内核控制包括安全、存储器模型、程序管理、网络堆栈和驱动程序模型等等。

三、Android的优秀设计

3.1 硬件抽象层的设计

硬件抽象层(HAL)是安卓操作系统的一大特性，以封闭源代码的形式来提供硬件驱动模块，并且将安卓的程序开发框架和内核进行了隔离，不至于使得安卓系统过分依赖Linux内核，达到了内核独立的目的，也可以让安卓可以在不考虑驱动程序如何实现的前提下进行发展。

3.2 安全权限机制的设计

安卓操作系统本身就是一个权限分立的操作系统，在这类操作系统中，每个应用都以一个系统识别身份运行，系统的各部分也分别使用各自独立的识别方式，用类似于Linux操作系统的方式将应用和应用，应用和操作系统隔离开来。

安卓操作系统的安全架构的核心设计理念是，在默认设置下，所有应用都没有权限对其他的应用程序、操作系统和用户程序产生比较大的影响，包括个人隐私数据的读写，其他应用程序文件的读写，访问网络和阻止设备待机等等。并且在应用程序安装的时候就会提供权限清单并要求用户进行确认。

3.3 应用组件层级的设计

3.3.1 广播接收器设计

在安卓的应用程序中，广播是一种广泛运用于应用程序之间信息传输和交互的机制，而广播接收器就是对广播进行过滤接受并进行响应的一类组件。广播接收器不能生成UI组件，但是只要是在运行时的代码中使用 `Context.registerReceiver` 进行注册。之后，只要有广播发送到当前应用程序，即使没有启动，系统也会自动启动程序并接收广播。

3.3.2 内容提供者设计

内容提供者是安卓提供的第三方应用数据的访问方案。由于安卓手机对数据的保护是比较严密的，一个应用程序的数据库和文件等内容一般是不允许其他程序直接访问的，但是内容提供者(`content provider`)可以实现这一部分的功能，应用程序可以通过派生 `ContentProvider` 类的方法来实现第三方应用数据的访问，每个内容提供者都可以使用一个url来作为独立的标识。

四、关于Android操作系统的思考

4.1 为什么Android选择Linux内核？

成熟的操作系统内核有很多，但是安卓操作系统选用了Linux作为内核，这和Linux本身的一些特性有关，包括：

- 强大的进程管理和内存管理方案
- 基于权限的隐私安全保护
- 支持库的共享
- 经过认证的驱动模型
- Linux本身是开源的，和安卓不谋而合

4.2 为什么Android和Linux操作系统的区别在哪里？

- 安卓操作系统没有本地窗口系统，即Linux中的X窗口系统。
- 和常见的Linux桌面发行版(比如Ubuntu)相比，安卓操作系统的设备用户没有root权限可以进入操作系统，这是因为作为一筐商用的移动端操作系统，安卓需要照顾到大多数非开发者用户的使用体验，因此不向用户开放root权限，这使得用户的操作难度大大降低，也不用担心因为进入了root模式而对手机本身造成不必要的操作
- 安卓没有glibc库作为支持，这是出于效率的考虑，谷歌自己开发了一套BionicLibc来代替glibc，好处就是比起glibc，Bionic的运行效率更高，对硬件的支持要求也比较低，安卓对低频CPU进行了优化
- 安卓操作系统并不包括一整套标准的Linux实用程序，因为它并没有照搬Linux内核而实在修复bug的基础上增加了yaffs2 FLAH文件系统
- 安卓操作系统有专用的驱动程序，对Linux的驱动设备进行了增强，比如电源管理、低内存管理、匿名共享内存和轻量级日志设备等等

4.3 为什么Android能够成功？

- 开源：相比于iOS和已经被淘汰的塞班系统，安卓最大的优势就是其开源的特性，开放的平台允许任何终端厂商和开发者加入到安卓系统开发的阵营中，随着用户和应用的日益丰富，一个崭新的平台也将很快成熟。同时开放性的平台也会带来更大的竞争，使得安卓手机的价格竞争更加激烈，这样一来消费者就可以用更少的钱来购买心仪的移动设备，这也促进了安卓的市场份额不断扩大。
- 丰富的硬件选择：这一点还是和安卓操作系统的开放性有关，由于安卓的开放性，众多的厂商就会推出更多具有特色的产品以博取市场份额，但是这却不会影响数据的同步和软件的兼容，这也构成了基于安卓操作系统的移动设备的一大竞争力。
- 谷歌的大力支持：安卓操作系统是由互联网巨头谷歌公司牵头开发和维护的，得到了谷歌的大力推广和支持，此外，谷歌提供的各类服务，比如搜索引擎、地图、邮件等功能，都可以和安卓操作系统的设备无缝结合。
- 这也启示我们，要做出一款属于自己的操作系统，依然还有很长的路要走。