



对于任何一个对Android开发感兴趣的人而言，对于android系统的学习必不可少。而学习系统最佳的方法就如linus所言:"RTFSC"(Read The Fucking Source Code)。

下面从知乎整理了一些优质回答，以飨读者。

巨人的肩膀

- AOSP项目官方: <https://source.android.com/source/index.html>

这个一定要先读: 项目介绍, 代码下载, 环境搭建, 刷机方法, Eclipse配置都在这里. 这是一切的基础.

- Android官方手册: <https://developer.android.com/training/index.html>

这个其实是给App开发者看的. 但是里面也有不少关于系统机制的介绍, 值得细读.

- 老罗的Android之旅: <http://blog.csdn.net/luoshengyang>

此老罗非彼老罗. 罗升阳老师的博客非常有营养, 基本可以作为**指引你开始阅读AOSP源码的教程**. 你可以按照博客的时间顺序一篇篇挑需要的看. 但这个系列的博客有些问题:

早期的博客是基于旧版本的Android;

大量的代码流程追踪. 读文章时你一定要清楚你在看的东西在整个系统处于什么样的位置.

- Innost的专栏: <http://blog.csdn.net/innost>

邓凡平老师也是为Android大牛, 博客同样很有营养. 但是不像罗升阳老师的那么系统. 更多的是一些技术点的深入探讨.

- Android Issues: <http://code.google.com/p/android/issues/list>

Android官方Issue列表. 我在开发过程中发现过一些奇怪的bug, 最后发现这里基本都有记录. 当然你可以提一些新的, 有没有人改就是另外一回事了.

- Google: <https://www.google.com>

一定要能流畅的使用这个工具. 大量的相关知识是没有人系统的总结的, 你需要自己搞定.

阅读方法

假设我想研究Android的UI系统, 首先要找什么和UI有亲戚关系吧!

View大神跳出来了, 沿着它往下找找看, 发现它在贴图在画各种形状, 但是它在哪里画呢, 马良也要纸吧?

开发Android的同学逃不掉Activity吧! 它有个setContentView()的方法, 从这个名字看好像它是把view和activity结合的地方. 赶紧看它的实现和被调用, 然后我们就发现了Window, ViewRoot和WindowManager的身影, 沿着WM和WMS我们就惊喜会发现了Surface, 以及draw的函数, 它居然在一个DecorView上画东西哈. 借助Source Insight, UIJava层的横向静态图呼之欲出了.

完成这个静态UML, 我觉得我可以开始功能实现上追踪了, 这部分主要是C++的代码 (这也是我坚定劝阻的放弃Eclipse的原因), 我沿着draw函数, 看到了各个层级的关系, SurfaceSession的控制和事务处理, SharedBuffer读写控制, 彪悍的SurfaceFlinger主宰一切, OpenGL ES的神笔马良.

Framebuffer和FramebufferDevice的图像输出, LCD设备打开后, 开始接收FBD发过来的一帧帧图像, 神奇吧.

好吧, 就这样, 再往底层我爱莫能助了!

软件

当我决定要阅读源码, 要具备一款好用的阅读器、下载源码等

- Windows阅读器: [Source Insight](#)

在这个工具帮助下, 你才可以驾驭巨大数量的Android 源码, 你可以从容在Java, C++,C代码间遨游, 你可以很快找到你需要的继承和调用关系.

- Mac OS阅读器: [Understand](#)

参考: [OS X 下真怀念 Source Insight](#)

- 源码下载:

如果你有梯子:

[官方下载](#)

[git路径](#)

如果没有:

[github路径](#) (可以直接download zip或者使用git clone)

欢迎读者提供优质下载路径(镜像等)来共享

相关知识

- Java

Java是AOSP的主要语言之一. 没得说, 必需熟练掌握.

熟练的Android App开发

- Linux

Android基于Linux的, 并且AOSP的推荐编译环境是Ubuntu 12.04. 所以熟练的使用并了解Linux这个系统是必不可少的. 如果你了解偏底层的代码, 那么必需了解基本的Linux环境下的程序开发. 如果再深入到驱动层, 那么Kernel相关的知识也要具备.

- 设计模式

去学习一下, android系统里的代码很多地方都闪烁着设计模式的光芒, 这也是你成为大牛的必经之路.当然你只要先了解一下, 在阅读中慢慢感受就行.

- Make

AOSP使用Make系统进行编译. 了解基本的Makefile编写会让你更清晰了解AOSP这个庞大的项目是如何构建起来的.

- Git

AOSP使用git+repo进行源码管理. 这应该是程序员必备技能吧.

- C++

Android系统的一些性能敏感模块及第三方库是用C++实现的, 比如: Input系统, Chromium项目(WebView的底层实现).

感谢知乎及知乎er

[大牛们是怎么阅读 Android 系统源码的?](#)

彩蛋

最后，很多优秀资源来源于国外，如果android学习者连android官网都打不开的话，那就有点。。。。

关于vpn要**委婉**吐槽一句：

与其像本人以前一样花一大堆时间去搜索各种不稳定的**免费vpn**，还真心不如花个十几块钱省一个月心来的实在。学习技术总要支付学费，但购买优质vpn这点个人觉得是性价比极高也极机智的做法。

关于作者

欢迎关注本人的Github<https://github.com/wingjay>, 及阅读本人相关文章

[《如何在一天之内完成一款具备cool属性的Android产品<简诗>》](#)