

前言

看到一篇文章中提到“最近几年国内的初级Android程序员已经很多了，但是中高级的Android技术人才仍然稀缺”，这的确不假，从我在百度所进行的一些面试来看，找一个适合的高级Android工程师的确不容易，一般需要进行大量的面试才能挑选出一个比较满意的。为什么中高级Android程序员不多呢？这是一个问题，我不好回答，但是我想写一篇文章来描述下Android的学习路线，期望可以帮助更多的Android程序员提升自己。由于我也是从一个菜鸟过来的，所以我会结合我的个人经历以及我对Android学习过程的认识来写这篇文章，这会让这篇文章更加真实，而并非纸上谈兵。

我的工作经历

前面说到，我也是从一个Android菜鸟过来的。其实这句话放在任何人的身上都是适用的，即大家都是一步步过来的，因此作为初学者也不必因为技术差而郁闷，同理，高手也不要看不起一些所谓的菜鸟，因为这不公平，技术只有在同等的工作年限上才具有一定的可比性，也许你眼中的菜鸟只是个工作半年的新手，而你已经工作5年，可想而知，这根本不具有可比性，搞不好别人5年后可以达到比你更高的技术高度。

我有若干个技术QQ群，群里（481798332）的小伙伴对我的工作经历比较感兴趣，刚好在这里我就做下介绍。我是硕士研究生毕业，我已经工作3年零3个月了，职位上从最开始的腾讯初级工程师变成了现在的百度Android资深工程师。最开始我并不是做Android的，先是做了半年的C++，接着做了3个月的Web前端，然后公司内部转岗做Android到至今，纯Android工作年限的话其实是2.5年。但是我认为我的Android工作经验（注：工作年限不等于工作经验）不止2.5年，我投入大量的业余时间Android开发上，并且我习惯去思考问题、总结问题并将其融会贯通，因此我的Android技术在短时间内有了很大的提升。

在Android学习过程中，初学者踩过的坑我也都踩过，我也深深地知道大家在学习过程中最棘手的问题是什么。举个例子，在3年前，我想在SlidingMenu中放入一个ListView，发现二者总是不能很好地一起工作，这肯定是由于滑动冲突的问题，我也知道，但是不知道怎么解决。我就去学校图书馆翻遍了所有的Android书籍，无果。大家肯定都知道原因，为什么我无法从书中查找到问题的答案，因为入门书不讲滑动冲突，所谓的高级编程书也不讲。还有一个问题，我曾经以为view.post(runnable)可以让runnable的run方法在线程中执行，所以我就在run方法里面去做耗时的操作，现在想想我当时是多菜啊，因此我曾经也是菜鸟。

直到若干年后的某一天，我终于琢磨透了滑动冲突的事，然后我就把解决问题的思想写在了CSDN博客上，但是好像看得人并不多，很多人并不能体会我的用心，后来我博客的访问量越来越大，这才慢慢地得到了一些人的关注。后来有一天我有了写书的契机，我想到了我最开始学习Android时所踩过的坑，想到滑动冲突对我的困扰，为了更好地传播我的技术经验，让更多的人少踩一些坑，让更多的人成为Android高级工程师，我毅然决定将Android开发中最重要的、最疑难的、最容易困扰大家的、成为高级工程师所必备的知识点和盘托出，这就是《**Android开发艺术探索**》存在的原因以及意义。书的反响怎么样呢？从目前读者的评价来看，内容基本无差评，我收到了很多读者的肯定以及感谢，这说明很多人能够理解我的用心。

说了那么多，言归正传，下面说下Android学习路线的话题，本文打算从4个阶段来对Android的学习过程做一个全面的分析，分别为Android初级、中级、高级以及资深工程师，具体请看下面的分析。同理，本篇学习路线仍然只针对Android应用开发，不针对Rom开发和逆向工程等。这里虚拟一位“小明”的角色，在这里小明就是Android初学者的代表。

初级工程师

小明之前完全没接触过Android开发，是个应届生，他的待遇是13k，然后小明以校招的身份进入了百度，然后小明需要怎么学习才能成为初级工程师呢？这个时候，小明对编程基础、数据结构、C语言都有一定基础，Java语法什么的也都掌握的比较好，Android才有java语言，无奈的是小明并不会搞Android。

小明首先需要购买一本Android入门的书籍，为了更快地学习Android，小明业余时间也都用来一边看书一边照着书中的例子敲代码，结果2周时间小明就把这本书学了一遍。看完这本书后，小明对Android的历史、结构、代码规范等都有了一个大概的了解，并且，小明已经可以写出一些简单的Activity了。这个时候在小明眼里，Android开发很简单很好玩，通过在xml中摆放一些按钮文本框什么的就可以做一些界面了。

小明开始跟着他的技术导师做需求，一些简单的小需求小明自然是不在话下了。突然有一天来了一个需求，该需求要求小明在Activity中为一个button加一个动画效果，小明慌了：“完全没接触过，书上也没有讲，怎么办呢？”小明冷静了下，打开了百度搜索，输入“Android 动画”，打开前几个链接，小明恍然大悟，照着网上的例子把需求给实现了。后来导师告诉他：“学好Android，官方文档是必须看的，既全面又权威”。然后小明如获至宝，花了一年时间把上面的guide和training都看了一遍，并且他还动手抄了几个小例子。

有一天，小明又需要做一个动画相关的需求，这可难不倒小明，它熟练地打开了www.baidu.com，输入“Android 动画”，突然他楞了一下：“总不能每次写动画都要百度一下吧！”，于是他在CSDN开了一个博客，把动画相关的知识点都写上去，为的是后面再写动画相关的代码就不用百度去搜了，事实如何呢？后面再写动画相关的代码，小明的确不用再去百度搜了，因为通过写一篇动画博客，他把动画相关的细节都已经记住了，这样他就可以不用再去参考任何文档了，后来小明还学会了把一些琐碎的不方便放在博客上的东西写到了印象笔记上面，什么时候忘了10秒钟以内都可以快速找回来，而不是花10分钟去再次搜索一遍。

这里总结一下，Android入门的时候，需要有一本入门书，好好学习书中的内容，同时花一年时间把Android官方文档中的training和guide看一遍，同时通过写博客和记笔记的方式来做总结，建议让自己的每篇博客都有价值些。通过一年时间的学习，相信每个人都可以达到中级工程师的水平。

技术要求：

- 基本知识点

比如四大组件如何使用、如何创建Service、如何进行布局、简单的自定义View、动画等常见技术

- 书籍推荐

《第一行代码 Android》、《疯狂Android》

中级工程师

小明经过一年的努力学习终于成为Android中级工程师了，月薪变成了17k。到了中级工程师，已经可以在公司里干很多体力活了，但是一些很重要的任务小明还不能一个人承担起来，这个时候小明需要学习的内容就很多了，如下所示：

- AIDL：熟悉AIDL，理解其工作原理，懂transact和onTransact的区别；
- Binder：从Java层大概理解Binder的工作原理，懂Parcel对象的使用；
- 多进程：熟练掌握多进程的运行机制，懂Messenger、Socket等；
- 事件分发：弹性滑动、滑动冲突等；
- 玩转View：View的绘制原理、各种自定义View；

- 动画系列：熟悉View动画和属性动画的不同点，懂属性动画的工作原理；
- 懂性能优化、熟悉mat等工具
- 懂点常见的设计模式

学习方法

阅读进阶书籍，阅读Android源码，阅读官方文档并尝试自己写相关的技术文章，需要有一定技术深度和自我思考。在这个阶段的学习过程中，有2个点是比较困扰大家的，一个是阅读源码，另一个是自定义View以及滑动冲突。

如何阅读源码呢？这是个头疼的问题，但是源码必须要读。阅读源码的时候不要深入代码细节不可自拔，要关注代码的流程并尽量挖掘出对应用层开发有用的结论。另外仔细阅读源码中对一个类或者方法的注释，在看不懂源码时，源码中的注释可以帮你更好地了解源码中的工作原理，这个过程虽然艰苦，但是别无他法。

如何玩转自定义View呢？我的建议是不要通过学习自定义view而学习自定义view。为什么这么说呢？因为自定义view的种类太多了，各式各样的绚丽的自定义效果，如何学的玩呢！我们要透过现象看本质，更多地去关注自定义view所需的知识点，这里做如下总结：

- 搞懂view的滑动原理
- 搞懂如何实现弹性滑动
- 搞懂view的滑动冲突
- 搞懂view的measure、layout和draw
- 然后再学习几个已有的自定义view的例子
- 最后就可以搞定自定义view了，所谓万变不离其宗

大概再需要1-2年时间，即可达到高级工程师的技术水平。我个人认为通过《Android开发艺术探索》和《Android群英传》可以缩短这个过程为0.5-1年。注意，达到高级工程师的技术水平不代表就可以立刻成为高级工程师（受机遇、是否跳槽的影响），但是技术达到了，成为高级工程师只是很简单的事。

技术要求：

- 稍微深入的知识点
- AIDL、Messenger、Binder、多进程、动画、滑动冲突、自定义View、消息队列等
- 书籍推荐
- 《Android开发艺术探索》、《Android群英传》

高级工程师

小明成为了梦寐以求的高级工程师，月薪达到了20k，还拿到了一丢丢股票。这个时候小明的Android水平已经不错了，但是小明的目标是资深工程师，小明听说资深工程师月薪可以达到30k+。

为了成为Android资深工程师，需要学习的东西就更多了，并且有些并不是那么具体了，如下所示：

- 继续加深理解” 稍微深入的知识点 “中所定义的内容
- 了解系统核心机制：
 1. 了解SystemServer的启动过程
 2. 了解主线程的消息循环模型
 3. 了解AMS和PMS的工作原理
 4. 能够回答问题” 一个应用存在多少个Window？ “
 5. 了解四大组件的大概工作流程
 6. ...

- 基本知识点的细节

1. Activity的启动模式以及异常情况下不同Activity的表现

2. Service的onBind和onReBind的关联

3. onServiceDisconnected(ComponentName className)和binderDied()的区别

4. AsyncTask在不同版本上的表现细节

5. 线程池的细节和参数配置

6. ...

- 熟悉设计模式，有架构意识

学习方法

这个时候已经没有太具体的学习方法了，无非就是看书、看源码和做项目，平时多种总结，尽量将知识融会贯通从而形成一种体系化的感觉。同时这个阶段对架构是有一定要求的，架构是抽象的，但是设计模式是具体的，所以一定要加强下设计模式的学习。关于设计模式的学习，最近一本新书推荐给大家

《Android 源码设计模式解析与实战》，既可以学习设计模式，又可能体会到Android源码中的设计思想，我最近也在阅读此书。

技术要求：

- 稍微深入的知识点

- 系统核心机制

- 基本知识点的细节

- 设计模式和架构

- 书籍推荐

《Android开发艺术探索》、《Android 源码设计模式解析与实战》、《Android内核剖析》

资深工程师

这个阶段的程序员也许并没有太具体的学习路线了。