

Android Study Plan

deepwaterooo

2021 年 12 月 17 日

目录

1 Daily updates	1
2 职场弱弱的奔溃瞬间 (+1)	1
3 系统服务篇	2
4 通信框架篇	3
5 应用组件篇	3
6 珠玑拾遗	3
7 Android 操作系统概述	3
8 Android 开发环境搭建	3
9 Android 项目结构分析	3
10 Android UI 设计	3
11 Intent Receive	3
12 Service	3
13 Content Provider	4
14 Android 高级应用开发	4
15 Android 程序发布部署建	4
16 Android 底层架构分析	4
17 Android 移植	4
18 阶段项目实战与测试	4
19 自定义 view Android 11 api level android M	4
19.1 gradle.properties	4

1 Daily updates

- 因为需要准备一个面试，这几天不更新算法系列，只准备安卓面试
- win 10 wsl 1 ubuntu 18 adb configuration done, not complicated, once search for solutions, and config, it works.
- api level 30 androidx basic project environment can be configured, but dynamic fragment add not done yet, switched back to api 28 temporarily
- wsl 1 ubuntu 18 adb logcat is so slow. after 9pm yesterday, 150K lines of log, I could only adb logcat previous project samplefragments, can not grab any FirstFragment nor Second Fragment Log.i info. may switch laptop back to Mac, but it has only 8GB memory, and internet too slow for that laptop
- today will review major components, as well small details, notes summaries to move on
- 恶鬼们昨天晚上今天早上又疯狂作案一晚，一个故意晚回来，用洗手间，用厨房，半夜一两点钟还在故意厨房里开厨导柜吵别人；一个今天早上五点钟便受恶房东暗示指使，故意开车库噪音极大的声响，早上五点钟便故意吵醒别人休息，实在是无恶不作（它有昨天晚上下班后到休息前一晚上的时间他不提前装车，非要等到早上五点钟故意把别人吵醒，另一个贱畜牲的做法是在别人晚上入睡前故意吵别人，贱畜牲们的做法还不够猖狂吗？）
- BroadcastReceiver sample project runs though easily. Will look into ContentProvider projects too today, all of them are not complicated.
- ContentProvider Interprocess communication projects are connected well too. One functions as server, the other as client, not complicated at all.
- The coding interview is done (due to the interface reasons, the 120 minutes test was submitted in about 60 minutes by mistake, the coding test interface does not false-positive submit proof), expecting the final round of interview, while at the same time, I am returning back to algorithms, and tomorrow there is a weekly contest too.

2 职场弱弱的奔溃瞬间（+1）

- 联系我安排面试相关的按照名字来看，应该是一个美国人吧，关于文化、语言等，这样的 hr 写邮件与 candidate 沟通应该会相对比较顺利一点

Robin Williams | Adidev Technologies Inc
to me ▾

Mon, Dec 13, 11:41 AM (4 days ago)

Not a problem buddy! Thanks for sharing your contact information, I'll update it in my details.

Could you please give me a call at 404-631-6245?

Thanks & Regards,

Robin Williams

Adidev Technologies Inc

- 偶这个职场弱弱，瞬间被这五个字母吓傻了，怎么也不敢跟他们联系，直到下午约定的时候中介打电话进来。

- 记得上次与另一个公司的第一轮面试快结束时间谈及第二轮的 **coding test** 是怎么样，会是像 **leetcode** 之类的刷题界面一样吗？我是一个 **emacs** 依赖症重症患者，离开 **command line based editor emacs**，我不会写代码。记得当时与面试官沟通这个问题的时候，他说到会 **sharing screen**，这个我们面试者就可以在自己最熟悉与舒适的工作环境中尽可能高效地答题、写出理想的代码。
- 今天早上，当我第一时间读及收件箱邮件里的 **reminder**,

Test details

1. Test Duration: 1 hour and 20 mins
2. Can only enter exam one time
3. Test Link: will be sent from Abhishek Sarkar and it must be completed between **10am EST and 4pm EST today**.

Important

1. Once the test starts, you cannot pause it. The test will run continuously for 1 hour and 20 mins.
2. All popups/tabs/window switches will log you out of the test automatically. We recommend that you close all programs including your email before you start.
3. You can only copy and paste code within the code editor. However, nothing can be copied and pasted from any external sources including code from the code editor of another question, external websites, etc.
4. This test is being monitored remotely. Your webcam must be turned on at all times.
5. This test can be taken in full-screen mode only.

- 作为一个久经沙场、久经面试的弱弱，内心瞬间奔溃、溃不成军。。。。。这究竟是要干什么???
 - 好在，作为一个久经沙场，久经面试的弱弱，死猪不怕开水烫，大不了就是不是 **command line based editor** 的界面会写得慢、会写得不顺，会无数次敲错字符，会写出来的代码不成型，**bug** 多，极度降低效率，但它好歹还是一个 **test** 不是？这种 **test** 的机会也不是很多呀
 - 于是，偶们还是硬着头皮明知山有虎地跑去把 **coding test** 给做了。。
- 回想 2012 年 9 月还是 10 月继自己装 **linux** 系统把电脑弄坏之后，被自己作业 **c++** 代码里的一个 **bug** 吓得魂飞魄散，急忙给代课教师写邮件求救。。。。。原来这么多年过去了，偶们还是这么一个弱弱。。。。。

3 系统服务篇

:clock1: DONE: Android 如何启动? [x] DONE: Android 应用进程启动流程 [] 什么是系统服务? [] ActivityManagerService [] SystemServer [x] DONE: Android 应用安装过程源码解析 [] WindowManagerService [] Zoyote 前世今生

4 通信框架篇

[x] Binder 完全解析 [x] DONE: Binder 完全解析（一）概述 [x] DONE: Binder 完全解析（二）设计详解 [x] DONE: Binder 完全解析（三）AIDL 实现原理分析 [x] Handler 通信框架 [x] DONE: Handler 消息机制源码解析

5 应用组件篇

[] Application 是什么? DONE: Context 分析 [] Activity 组件分析 [x] DONE: Activity 生命周期是如何实现的 [] Services 组件分析 [] ContentProvider 组件分析 [] Broadcast 组件分析

6 珠玑拾遗

[] Gradle 用法 [] 混淆一二事
Andriod 系统开发

7 Android 操作系统概述

Android 平台介绍；Android 平台特性；Android 平台架构；Android Navtive C/C++ 程序开发；Android NDK；Native 开发方式与 JAVA 开发方式比较。

8 Android 开发环境搭建

Android SDK 介绍；Eclipse ADT 插件；Android 模拟器开发。

9 Android 项目结构分析

资源管理 (Resources) 分析；drawable 分析；layout 分析；Activity 分析；Intent 分析；Service 分析；Content 分析。

10 Android UI 设计

标准控件的使用；设计开发自定义控件；Layout 布局的使用；触摸/按键（UI Events）事件处理方法；View，SurfaceView，Canvas,Paint 类分析使用；显示文本以及显示特殊效果文本；绘图及显示图片；实现动画效果。

11 Intent Receive

Intent 的作用和目的；属性讲解；Android 定义解析 Intent；AndroidManifest.xml 深入分析。

12 Service

什么是 Service，如何使用 Service，Service 的生命周期，BroadcastReceiver 的使用。

13 Content Provider

SQLite 介绍，创建 Content Providers，使用 Content Providers，使用 URI 语法进行增删改查。

14 Android 高级应用开发

访问本地通讯录；网络连接的相关知识；流媒体的处理；URLConnection 和 HttpURLConnection 的应用；HttpClient 的分析；本地文件浏览管理；音视频播放处理；Widget 应用开发。

15 Android 程序发布部署建

Android 调试桥；启用 logcat 日志调试；模拟器上安装删除软件；打包 * 签名和安装软件到设备。

16 Android 底层架构分析

移植 Android 到新的硬件平台；需要支持 Linux 操作系统的硬件平台架构分析；支持 Android 的 Linux 内核特性分析；为 Linux 内核增加 Android 特性；移植 Android Debug Bridge 调试接口；编写 / 移植 Android 内核驱动；硬件支持 double frame buffer/page flipping；bionic 库移植与优化；Dalvik Vm 移植；第三方应用程序移植；建立 Android 移植开发平台；新的嵌入式处理器引入的 Android 相关问题；获得高效的 Android 工具链。

17 Android 移植

支持 ARM11 的 Linux-2.6.28 内核新特性简介；移植 LCD double buffer 驱动；移植触摸屏驱动；移植 Android 键盘驱动；移植 Wifi 驱动支持 Android 上网功能；移植电源管理驱动，支持 Android 电池管理；部署 Android 系统到实际 ARM11 平台。

18 阶段项目实战与测试

通过对 ITelephony 接口和 ISms 接口以及 AIDL 在 Android 程序中的开发应用，开发一个打电话和发短信的程序。

19 自定义 view Android 11 api level android M

19.1 gradle.properties

- 什么是 Jetifier? 例如，要使用 androidx 打包的依赖项创建新项目，此新项目需要在 gradle.properties 文件中添加以下行：

```
java version 8 compileOptions { sourceCompatibility JavaVersion.VERSION_1_8 targetCompatibility JavaVersion.VERSION_1_8 }
```

```
import android.os.Bundle; import android.support.design.widget.FloatingActionButton; import android.support.design.widget.Snackbar; import android.view.View; import android.view.Menu; import android.view.MenuItem; import androidx.appcompat.app.AppCompatActivity; import androidx.appcompat.widget.Toolbar; import com.google.android.material.floatingactionbutton.FloatingActionButton; import com.google.android.material.snackbar.Snackbar;
```

```
<com.me.generalprac.CustomTitleView android:layout_width="match_parent" android:layout_height="wrap_content">
<include layout="@layout/custom_title"/>
```