Android 高级面试题及答案

一 性能优化

1.如何对 Android 应用进行性能分析

android 性能主要之响应速度 和 UI 刷新速度。

可以参考博客：Android 系统性能调优工具介绍

首先从函数的耗时来说，有一个工具 TraceView 这是 androidsdk 自带的工作，用于测量函数耗时的。

UI 布局的分析，可以有 2 块，一块就是 Hierarchy Viewer 可以看到 View 的布局层次，以及每个 View 刷新加载的时间。

这样可以很快定位到那块 layout & View 耗时最长。

还有就是通过自定义 View 来减少 view 的层次。

2.什么情况下会导致内存泄露

内存泄露是个折腾的问题。

什么时候会发生内存泄露？内存泄露的根本原因：长生命周期的对象持有短生命周期的对象。短周期对象就无法及时释放。

I. 静态集合类引起内存泄露

主要是 hashmap，Vector 等，如果是静态集合 这些集合没有及时 setnull 的话，就会一直持有这些对象。

II.remove 方法无法删除 set 集 Objects.hash(firstName, lastName);

经过测试，hashcode 修改后，就没有办法 remove 了。

III. observer 我们在使用监听器的时候，往往是 addxxxlistener，但是当我们不需要的时候，忘记 removexxxlistener，就容易内存 leak。

广播没有 unregisterrecevier

IV.各种数据链接没有关闭，数据库 contentprovider，io，sokect 等。cursor

V.内部类：

java 中的内部类（匿名内部类），会持有宿主类的强引用 this。

所以如果是 new Thread 这种，后台线程的操作，当线程没有执行结束时，activity 不会被回收。

Context 的引用，当 TextView 等等都会持有上下文的引用。如果有 static drawable，就会导致该内存无法释放。

VI.单例

单例 是一个全局的静态对象，当持有某个复制的类 A 是，A 无法被释放，内存 leak。

3.如何避免 OOM 异常

首先 OOM 是什么？

当程序需要申请一段"大"内存，但是虚拟机没有办法及时的给到，即使做了 GC 操作以后

这就会抛出 OutOfMemoryException 也就是 OOM

Android 的 OOM 怎么样？

为了减少单个 APP 对整个系统的影响，android 为每个 app 设置了一个内存上限。

```
public void getMemoryLimited(Activity context)
    {
        ActivityManager activityManager
=(ActivityManager)context.getSystemService(Context.ACTIVITY_SERVICE);
        System.out.println(activityManager.getMemoryClass());
        System.out.println(activityManager.getLargeMemoryClass());
        System.out.println(Runtime.getRuntime().maxMemory()/(1024*1024));
    }
09-10 10:20:00.477 4153-4153/com.joyfulmath.samples I/System.out: 192
```

```
09-10 10:20:00.477 4153-4153/com.joyfulmath.samples I/System.out: 512
```

09-10 10:20:00.477 4153-4153/com.joyfulmath.samples I/System.out: 192

HTC M7 实测，192M 上限。512M 一般情况下，192M 就是上限，但是由于某些特殊情况，android 允许使用一个更大的 RAM。

如何避免 OOM

减少内存对象的占用

I.ArrayMap/SparseArray 代替 hashmap

II.避免在 android 里面使用 Enum

III.减少 bitmap 的内存占用

- inSampleSize：缩放比例，在把图片载入内存之前，我们需要先计算出一个合适的缩放比例，避免不必要的大图载入。
- decode format：解码格式，选择 ARGB_8888/RBG_565/ARGB_4444/ALPHA_8，存在很大差异。

IV.减少资源图片的大小，过大的图片可以考虑分段加载

内存对象的重复利用

大多数对象的复用，都是利用对象池的技术。

I.listview/gridview/recycleview contentview 的复用

II.inBitmap 属性对于内存对象的复用 ARGB_8888/RBG_565/ARGB_4444/ALPHA_8

这个方法在某些条件下非常有用，比如要加载上千张图片的时候。

III.避免在 ondraw 方法里面 new 对象

IV.StringBuilder 代替+

4.Android 中如何捕获未捕获的异常

```
public class CrashHandler implements Thread.UncaughtExceptionHandler {

    private static CrashHandler instance = null;


    public static synchronized CrashHandler getInstance()
    {
        if(instance == null)
        {
            instance = new CrashHandler();
        }
        return instance;
    }
```

```
    public void init(Context context)
    {
        Thread.setDefaultUncaughtExceptionHandler(this);
    }

    @Override
    public void uncaughtException(Thread thread, Throwable ex) {
        StringBuilder stringBuilder = new StringBuilder();
        stringBuilder.append("Thread:");
        stringBuilder.append(thread.toString());
        stringBuilder.append("\t");
        stringBuilder.append(ex);
        TraceLog.i(stringBuilder.toString());
        TraceLog.printCallStatck(ex);
    }
}
```

关键是实现 Thread.UncaughtExceptionHandler

然后是在 application 的 oncreate 里面注册。

5.ANR 是什么？怎样避免和解决 ANR（重要）

ANR->Application Not Responding

也就是在规定的时间内，没有响应。

三种类型：

1）. KeyDispatchTimeout(5 seconds) --主要类型按键或触摸事件在特定时间内无响应

2）. BroadcastTimeout(10 seconds) --BroadcastReceiver 在特定时间内无法处理完成

3）. ServiceTimeout(20 seconds) --小概率类型 Service 在特定的时间内无法处理完成

为什么会超时：事件没有机会处理 & 事件处理超时

怎么避免 ANR

ANR 的关键

是处理超时，所以应该避免在 UI 线程，BroadcastReceiver 还有 service 主线程中，处理复杂的逻辑和计算

而交给 work thread 操作。

1）避免在 activity 里面做耗时操作，oncreate & onresume

2）避免在 onReceiver 里面做过多操作

3）避免在 Intent Receiver 里启动一个 Activity，因为它会创建一个新的画面，并从当前用户正在运行的程序上抢夺焦点。

4）尽量使用 handler 来处理 UI thread & workthread 的交互。

如何解决 ANR

首先定位 ANR 发生的 log：

```
04-01 13:12:11.572 I/InputDispatcher( 220): Application is not
responding:Window{2b263310com.android.email/com.android.email.activ
ity.SplitScreenActivitypaused=false}.  5009.8ms since event,
5009.5ms since waitstarted
CPUusage from 4361ms to 699ms ago ----CPU 在 ANR 发生前的使用情况
04-0113:12:15.872 E/ActivityManager(  220): 100%TOTAL: 4.8% user + 7.6%
kernel + 87% iowait

04-0113:12:15.872 E/ActivityManager(  220): CPUusage from 3697ms to
4223ms later:-- ANR 后 CPU 的使用量
```

从 log 可以看出，cpu 在做大量的 io 操作。

所以可以查看 io 操作的地方。

当然，也有可能 cpu 占用不高，那就是 主线程被 block 住了。

6.Android 线程间通信有哪几种方式

1）共享变量（内存）

2）管道

3）handle 机制

runOnUiThread(Runnable)

view.post(Runnable)

7.Devik 进程，linux 进程，线程的区别

Dalvik 进程。

每一个 android app 都会独立占用一个 dvm 虚拟机，运行在 linux 系统中。

所以 dalvik 进程和 linux 进程是可以理解为一个概念。

8.描述一下 android 的系统架构

从小到上就是：

linux kernel,lib dalvik vm ,application framework, app

9.android 应用对内存是如何限制的?我们应该如何合理使用内存？

activitymanager.getMemoryClass() 获取内存限制。

关于合理使用内存，其实就是避免 OOM & 内存泄露中已经说明。

10. 简述 android 应用程序结构是哪些

1）main code

2) unit test

3）mianifest

4）res->drawable,drawable-xxhdpi,layout,value,mipmap

mipmap 是一种很早就有的技术了，翻译过来就是**纹理映射技术.**

google 建议只把启动图片放入。

5）lib

6）color

11.请解释下 Android 程序运行时权限与文件系统权限的区别

文件的系统权限是由 linux 系统规定的，只读，读写等。

运行时权限，是对于某个系统上的 app 的访问权限，允许，拒绝，询问。该功能可以防止非法的程序访问敏感的信息。

12.Framework 工作方式及原理，Activity 是如何生成一个 view 的，机制是什么

Framework 是 android 系统对 linux kernel，lib 库等封装，提供 WMS，AMS，bind 机制，handler-message 机制等方式，供 app 使用。

简单来说 framework 就是提供 app 生存的环境。

1）Activity 在 attch 方法的时候，会创建一个 phonewindow（window 的子类）

2）onCreate 中的 setContentView 方法，会创建 DecorView

3）DecorView 的 addview 方法，会把 layout 中的布局加载进来。

13.多线程间通信和多进程之间通信有什么不同，分别怎么实现

线程间的通信可以参考第 6 点。

进程间的通信：bind 机制（IPC->AIDL），linux 级共享内存，boradcast，

Activity 之间，activity & serview 之间的通信，无论他们是否在一个进程内。

14.Android 屏幕适配

屏幕适配的方式：xxxdpi， wrap_content,match_parent. 获取屏幕大小，做处理。

dp 来适配屏幕，sp 来确定字体大小

drawable-xxdpi, values-1280*1920 等 这些就是资源的适配。

wrap_content,match_parent, 这些是 view 的自适应

weight，这是权重的适配。

15.什么是 AIDL 以及如何使用

Android Interface Definition Language

AIDL 是使用 bind 机制来工作。

参数：

java 原生参数

String

parcelable

list & map 元素 需要支持 AIDL

16.Handler 机制

android 进程/线程管理（一）----消息机制的框架 这个系类。

17.事件分发机制

android 事件分发机制

18.子线程发消息到主线程进行更新 UI，除了 handler 和 AsyncTask，还有什么

EventBus，广播，view.post, runinUiThread

但是无论各种花样，本质上就 2 种：handler 机制 + 广播

19.子线程中能不能 new handler？为什么

必须可以。子线程 可以 new 一个 mainHandler，然后发送消息到 UI Thread。

20.Android 中的动画有哪几类，它们的特点和区别是什么

视图动画，或者说补间动画。只是视觉上的一个效果，实际 view 属性没有变化，性能好，但是支持方式少。

属性动画，通过变化属性来达到动画的效果，性能略差，支持点击等事件。android 3.0

帧动画，通过 drawable 一帧帧画出来。

Gif 动画，原理同上，canvas 画出来。

具体可参考：https://i.cnblogs.com/posts?categoryid=672052

21.如何修改 Activity 进入和退出动画

overridePendingTransition

22.SurfaceView & View 的区别

view 的更新必须在 UI thread 中进行

surfaceview 会单独有一个线程做 ui 的更新。

surfaceview 支持 open GL 绘制。

二项目框架的使用

23.开发中都使用过哪些框架、平台

I.EventBus 事件分发机制，由 handler 实现，线程间通信

II.xUtils->DbUtils,ViewUtils,HttpUtils,BitmapUtils

III.百度地图

IV.volley

V.fastjson

VI.picciso

VII.友盟

VIII.zxing

IX.Gson

24.使用过那些自定义 View

pull2RefreshListView

25.自定义控件：绘制圆环的实现过程

```
package com.joyfulmath.samples.Cycle;

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.util.AttributeSet;
import android.view.View;

/**
 * Created by Administrator on 2016/9/11 0011.
```

```
 */
public class CycleView extends View {
    Paint mPaint = new Paint();
    public CycleView(Context context) {
        this(context, null);
    }

    public CycleView(Context context, AttributeSet attrs) {
        super(context, attrs);
        initView();
    }

    private void initView() {
        mPaint.setAntiAlias(true);
        mPaint.setStyle(Paint.Style.STROKE);
        mPaint.setStrokeWidth(20);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        canvas.drawCircle(100,100,50,mPaint);
    }
}
```

关键是 canvas.drawCycle & paint.setsytle(stoken)

26.自定义控件：摩天轮的实现过程

27.GridLayout 的使用

可以不需要 adapter

28.流式布局的实现过程

TBD.

29.第三方登陆

QQ & 微信都有第三方登陆的 sdk，要去注册 app

30.第三方支付

需要看支付宝的 API 文档