[**Android 高级面试题及答案**](http://www.cnblogs.com/deman/p/5860976.html)

**阅读目录**

* [1.如何对 Android 应用进行性能分析](http://www.cnblogs.com/deman/p/5860976.html#_label0)
* [2.什么情况下会导致内存泄露](http://www.cnblogs.com/deman/p/5860976.html#_label1)
* [3.如何避免 OOM 异常](http://www.cnblogs.com/deman/p/5860976.html#_label2)
* [4.Android 中如何捕获未捕获的异常](http://www.cnblogs.com/deman/p/5860976.html#_label3)
* [5.ANR 是什么？怎样避免和解决 ANR（重要）](http://www.cnblogs.com/deman/p/5860976.html#_label4)
* [6.Android 线程间通信有哪几种方式](http://www.cnblogs.com/deman/p/5860976.html#_label5)
* [7.Devik 进程，linux 进程，线程的区别](http://www.cnblogs.com/deman/p/5860976.html#_label6)
* [8.描述一下 android 的系统架构](http://www.cnblogs.com/deman/p/5860976.html#_label7)
* [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？](http://www.cnblogs.com/deman/p/5860976.html#_label8)
* [10. 简述 android 应用程序结构是哪些](http://www.cnblogs.com/deman/p/5860976.html#_label9)
* [11.请解释下 Android 程序运行时权限与文件系统权限的区别](http://www.cnblogs.com/deman/p/5860976.html#_label10)
* [12.Framework 工作方式及原理，Activity 是如何生成一个 view 的，机制是什么](http://www.cnblogs.com/deman/p/5860976.html#_label11)
* [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](http://www.cnblogs.com/deman/p/5860976.html#_label12)
* [14.Android 屏幕适配](http://www.cnblogs.com/deman/p/5860976.html#_label13)
* [15.什么是 AIDL 以及如何使用](http://www.cnblogs.com/deman/p/5860976.html#_label14)
* [16.Handler 机制](http://www.cnblogs.com/deman/p/5860976.html#_label15)
* [17.事件分发机制](http://www.cnblogs.com/deman/p/5860976.html#_label16)
* [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](http://www.cnblogs.com/deman/p/5860976.html#_label17)
* [19.子线程中能不能 new handler？为什么](http://www.cnblogs.com/deman/p/5860976.html#_label18)
* [20.Android 中的动画有哪几类，它们的特点和区别是什么](http://www.cnblogs.com/deman/p/5860976.html#_label19)
* [21.如何修改 Activity 进入和退出动画](http://www.cnblogs.com/deman/p/5860976.html#_label20)
* [22.SurfaceView & View 的区别](http://www.cnblogs.com/deman/p/5860976.html#_label21)
* [23.开发中都使用过哪些框架、平台](http://www.cnblogs.com/deman/p/5860976.html#_label22)
* [24.使用过那些自定义View](http://www.cnblogs.com/deman/p/5860976.html#_label23)
* [25.自定义控件：绘制圆环的实现过程](http://www.cnblogs.com/deman/p/5860976.html#_label24)
* [26.自定义控件：摩天轮的实现过程](http://www.cnblogs.com/deman/p/5860976.html#_label25)
* [27.GridLayout的使用](http://www.cnblogs.com/deman/p/5860976.html#_label26)
* [28.流式布局的实现过程](http://www.cnblogs.com/deman/p/5860976.html#_label27)
* [29.第三方登陆](http://www.cnblogs.com/deman/p/5860976.html#_label28)
* [30.第三方支付](http://www.cnblogs.com/deman/p/5860976.html#_label29)

**一 性能优化**

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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](http://www.cnblogs.com/wanqieddy/p/4494896.html)

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。

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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。

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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。

[复制代码](javascript:void(0);)

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));

}

[复制代码](javascript:void(0);)

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+

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**4.Android 中如何捕获未捕获的异常**

http://images.cnblogs.com/OutliningIndicators/ContractedBlock.gif CrashHandler

关键是实现Thread.UncaughtExceptionHandler

然后是在application的oncreate里面注册。

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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.activity.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。

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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)

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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。

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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。

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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。

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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。

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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。

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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。

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**16.Handler 机制**

参考：[android 进程/线程管理（一）----消息机制的框架](http://www.cnblogs.com/deman/p/4688054.html) 这个系类。

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**17.事件分发机制**

[android 事件分发机制](http://www.cnblogs.com/deman/p/5812570.html)

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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。

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**21.如何修改 Activity 进入和退出动画**

 overridePendingTransition

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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**

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**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

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**24.使用过那些自定义View**

pull2RefreshListView

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**25.自定义控件：绘制圆环的实现过程**

http://images.cnblogs.com/OutliningIndicators/ContractedBlock.gif CycleView

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

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**26.自定义控件：摩天轮的实现过程**

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**27.GridLayout的使用**

可以不需要adapter

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**28.流式布局的实现过程**

TBD.

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**29.第三方登陆**

QQ & 微信都有第三方登陆的sdk，要去注册app

[回到顶部](http://www.cnblogs.com/deman/p/5860976.html#_labelTop)

**30.第三方支付**

需要看支付宝的API文档