# Android面试必备笔试题

**1：Android中五种数据存储方式分别是什么？他们的特点？**

答：Android提供了五种存取数据的方式

(1)SharedPreference，存放较少的五种类型的数据，只能在同一个包内使用，生成XML的格式存放在设备中

(2) SQLite数据库，存放各种数据，是一个轻量级的嵌入式数据库

(3) File文件，通过读取写入方式生成文件存放数据

(4) ContentProvider，主要用于让其他应用程序使用保存的数据

(5) 通过网络获取数据和写入数据到网络存储空间

**2:说说 android 中 mvc 的具体体现**

mvc是model,view,controller的缩写，mvc包含三个部分：

模型（model）对象：是应用程序的主体部分，所有的业务逻辑都应该写在该层。

视图（view）对象：是应用程序中负责生成用户界面的部分。也是在整个mvc架构中用户唯一可以看到的一层，接收用户的输入，显示处理结果。

控制器（control）对象：是根据用户的输入，控制用户界面数据显示及更新model对象状态的部分，控制器更重要的一种导航功能，响应用户出发的相关事件，交给m层处理。

android鼓励弱耦合和组件的重用，在android中mvc的具体体现如下：

(1)视图（view）：一般采用xml文件进行界面的描述，使用的时候可以非常方便的引入。(2)控制层（controller）：android的控制层的重任通常落在了众多的acitvity的肩上，这句话也就暗含了不要在acitivity中写过多的代码，要通过activity交割model业务逻辑层处理，这样做的另外一个原因是android中的acitivity的响应时间是5s，如果耗时的操作放在这里，程序就很容易被回收掉。

(3)模型层（model）：对数据库的操作、对网络等的操作都应该在model里面处理，当然对业务计算等操作也是必须放在的该层的。

**3:简述SharedPreferences存储方式以及SharedPreferences与SQLite数据库的区别**SharedPreferences也是一种轻型的数据存储方式，它的本质是基于XML文件存储key-value键值对数据，通常用来存储一些简单的配置信息。其存储位置在/data/data/<包名>/shared\_prefs目录下。SharedPreferences对象本身只能读取数据而不支持写入数据，存储修改是通过Editor对象实现。SharedPreferences对象与SQLite数据库相比，免去了创建数据库，创建表，写SQL语句等诸多操作，相对而言更加方便，简洁。但SharedPreferences也有其自身缺陷，比如其职能存储boolean，int，float，long和String五种简单的数据类型，比如其无法进行条件查询等。所以不论SharedPreferences的数据存储操作是如何简单，它也只能是存储方式的一种补充，而无法完全替代如SQLite数据库这样的其他数据存储方式。

**4：描述handler 机制的原理**

andriod提供了 Handler 和 Looper 来满足线程间的通信。

Handler 先进先出原则。

Looper类用来管理特定线程内对象之间的消息交换(Message Exchange)。

(1)Looper: 一个线程可以产生一个Looper对象，由它来管理此线程里的Message Queue(消息队列)。

(2)Handler: 你可以构造Handler对象来与Looper沟通，以便push新消息到Message Queue里;或者接收Looper从Message Queue取出)所送来的消息。

(3) Message Queue(消息队列):用来存放线程放入的消息。

(4)线程：UI thread 通常就是main thread，而Android启动程序时会替它建立一个Message Queue。

**5:显式intent和隐式intent的区别是什么（android）**

答：Intent定义：Intent是一种在不同组件之间传递的请求消息，是应用程序发出的请求和意图。作为一个完整的消息传递机制，Intent不仅需要发送端，还需要接收端。

显式Intent定义：对于明确指出了目标组件名称的Intent，我们称之为显式Intent。

隐式Intent定义：对于没有明确指出目标组件名称的Intent，则称之为隐式Intent。

说明：Android系统使用IntentFilter 来寻找与隐式Intent相关的对象。

**6：如何优化ListView**

1、如果自定义适配器，那么在getView方法中要考虑方法传进来的参数contentView是否为null，如果为null就创建contentView并返回，如果不为null则直接使用。在这个方法中尽可能少创建view。

2、给contentView设置tag（setTag（）），传入一个viewHolder对象，用于缓存要显示的数据，可以达到图像数据异步加载的效果。

3、如果listview需要显示的item很多，就要考虑分页加载。比如一共要显示100条或者更多的时候，我们可以考虑先加载20条，等用户拉到列表底部的时候再去加载接下来的20条。

**7：简述通过contentResolver获取ContentProvider内容的基本步骤**

第一：得到ContentResolver类对象：ContentResolver cr = getContentResolver（）；

第二：定义要查询的字段String数组。

第三：使用cr.query();返回一个Cursor对象。

第四：使用while循环得到Cursor里面的内容。

**8：描述4 种 activity 的启动模式**

(1)standard ：系统的默认模式，一次跳转即会生成一个新的实例。假设有一个activity命名为MainActivity，执行语句：

startActivity(new Intent(MainActivity.this, MainActivity.class))后，MainActivity将跳转到另外一个MainActivity，也就是现在的Task栈里面有MainActivity的两个实例。

按返回键后你会发现仍然是在MainActivity（第一个）里面。

(2)singleTop：singleTop 跟standard 模式比较类似。如果已经有一个实例位于Activity栈的顶部时，就不产生新的实例，而只是调用Activity中的newInstance()方法。如果不位

于栈顶，会产生一个新的实例。例：当MainActivity为 singleTop 模式时，执行跳转后栈里面依旧只有一个实例，如果现在按返回键程序将直接退出。

(3)singleTask： singleTask模式和后面的singleInstance模式都是只创建一个实例的。在这种模式下，无论跳转的对象是不是位于栈顶的activity，程序都不会生成一个新的实例

（当然前提是栈里面已经有这个实例）。这种模式相当有用，在以后的多activity开发中，经常会因为跳转的关系导致同个页面生成多个实例，这个在用户体验上始终有点不好，而

如果你将对应的activity声明为 singleTask 模式，这种问题将不复存在。

(4)singleInstance: 设置为 singleInstance 模式的 activity 将独占一个task（感觉task可以理解为进程），独占一个task的activity与其说是activity，倒不如说是一个应用，

这个应用与其他activity是独立的，它有自己的上下文activity。

**9：Android中如何访问自定义ContentProvider?**

通过ContentProvider的Uri访问开放的数据。

(1)ContenResolver对象通过Context提供的方法getContenResolver()来获得。(2)ContenResolver提供了以下方法来操作：insert  delete  update  query这些方法分别会调用ContenProvider中与之对应的方法并得到返回的结果。

**10：android中的动画有哪几类，它们的特点和区别是什么？**

答：Android中动画可以分为两大类：帧动画、补间动画

(1）补间动画：你定义一个开始和结束，中间的部分由程序运算得到。主要有：旋转、平移、放缩和渐变；AlphaAnimation（渐变型动画）、scaleAnimation（缩放型动画）、 TranslateAnimation（平移型动画）、 RotateAnimation（旋转型动画）

(2）逐帧动画：Frame（把一连串的图片进行系列化连续播放，如同放电影的效果），它是通过播放一张一张图片来达到动画的效果；

## 二

**1、 Android dvm的进程和Linux的进程, 应用程序的进程是否为同一个概念**

DVM指dalivk的虚拟机。每一个Android应用程序都在它自己的进程中运行，都拥有一个独立的Dalvik虚拟机实例。而每一个DVM都是在Linux 中的一个进程，所以说可以认为是同一个概念。

**2、sim卡的EF 文件有何作用**

sim卡的文件系统有自己规范，主要是为了和手机通讯，sim本身可以有自己的操作系统，EF就是作存储并和手机通讯用的

**3、嵌入式操作系统内存管理有哪几种，各有何特性 ?**

页式，段式，段页，用到了MMU,虚拟空间等技术

**4、什么是嵌入式实时操作系统, Android 操作系统属于实时操作系统吗?**

嵌入式实时操作系统是指当外界事件或数据产生时，能够接受并以足够快的速度予以处理，其处理的结果又能在规定的时间之内来控制生产过程或对处理系统作出快速响应，并控制所有实时任务协调一致运行的嵌入式操作系统。主要用于工业控制、军事设备、航空航天等领域对系统的响应时间有苛刻的要求，这就需要使用实时系统。又可分为软实时和硬实时两种，而android是基于linux内核的，因此属于软实时。

**5、一条最长的短信息约占多少byte?**

中文70(包括标点)，英文160个字节

**6、 android中的动画有哪几类，它们的特点和区别是什么？**

两种，一种是Tween动画、还有一种是Frame动画。Tween动画，这种实现方式可以使视图组件移动、放大、缩小以及产生透明度的变化；另一种Frame动画，传统的动画方法，通过顺序的播放排列好的图片来实现，类似电影。

**7、handler机制的原理**

andriod提供了 Handler 和 Looper 来满足线程间的通信。Handler 先进先出原则。Looper类用来管理特定线程内对象之间的消息交换(Message Exchange)。

1）Looper: 一个线程可以产生一个Looper对象，由它来管理此线程里的Message Queue(消息队列)。

2）Handler: 你可以构造Handler对象来与Looper沟通，以便push新消息到Message Queue里；或者接收Looper从Message Queue取出)所送来的消息。

3） Message Queue(消息队列):用来存放线程放入的消息。

4）线程：UI thread 通常就是main thread，而Android启动程序时会替它建立一个Message Queue。

**8、说说mvc模式的原理，它在android中的运用**

MVC(Model\_view\_contraller)” 模型\_视图\_控制器”。 MVC应用程序总是由这三个部分组成。Event(事件)导致Controller改变Model或View，或者同时改变两者。只要 Controller改变了Models的数据或者属性，所有依赖的View都会自动更新。类似的，只要Controller改变了View，View会从潜在的Model中获取数据来刷新自己。

# View重绘和内存泄露是面试经常问的问题

**1. View的刷新:**

在需要刷新的地方,使用handle.sendmessage发送信息,然后在handle的getmessage里面执行invaliate或者postinvaliate.

**2. GC内存泄露**

出现情况:

1.数据库的cursor没有关闭

2.构造adapter时,没有使用缓存contentview衍生listview的优化问题-----减少创建view的对象,充分使用contentview,可以使用一静态类来优化处理getview的过程

3.Bitmap对象不使用时采用recycle()释放内存

4.activity中的对象的生命周期大于activity调试方法: DDMS==> HEAPSZIE==>dataobject==>[Total Size]还有其他问题,大家欢迎提出阿,可以是整体架构的,还有这个Hal层.

**涉及到以下几个内容**

一 Activity的生命周期

二 让Activity变成一个窗口：Activity属性设定

三 你后台的Activity被系统回收怎么办：onSaveInstanceState

四 调用与被调用：我们的通信使者 - Intent

**一 Activity的生命周期**

和其他手机 平台 的应用程序一样，Android的应用程序的生命周期是被统一掌控的，也就是说我们写的应用程序命运掌握在别人（系统）的手里，我们不能改变它，只能学习并适应它。

**简单地了解一下为什么是这样：**

我们手机在运行一个应用程序的时候，有可能打进来电话发进来短信，或者没有电了，这时候程序都会被中断，优先去服务电话的基本功能，另外系统也不允许你占用太多资源，至少要保证电话功能吧,所以资源不足的时候也就有可能被干掉。言归正传，Activity的基本生命周期如下代码所示：

**Java 代码**

public class MyActivity extends Activity

{

protected void onCreate(Bundle savedInstanceState);

protected void onStart();

protected void onResume();

protected void onPause();

protected void onStop();

protected void onDestroy();

}

你自己写的Activity会按需要重载这些方法，onCreate是免不了的，在一个Activity正常启动的过程中，他们被调用的顺序是 onCreate -> onStart ->onResume, 在Activity被干掉的时候顺序是onPause -> onStop -> onDestroy ，这样就是一个完整的生命周期，但是有人问了，程序正运行着呢来电话了，这个程序咋办？中止了呗，如果中止的时候新出的一个Activity是全屏的那么：onPause->onStop ，恢复的时候onStart->onResume ，如果打断这个应用程序的是一个Theme为Translucent 或者Dialog 的Activity那么只是onPause ,恢复的时候onResume 。

**详细介绍一下这几个方法中系统在做什么以及我们应该做什么：**

**onCreate:**

在这里创建界面，做一些数据的初始化工作

**onStart:**

到这一步变成用户可见不可交互的

**onResume:**

变成和用户可交互的，（在activity 栈系统通过栈的方式管理这些个Activity的最上面，运行完弹出栈，则回到上一个Activity)

**onPause:**

到 这一步是可见但不可交互的，系统会停止动画等消耗CPU 的事情从上文的描述已经知道，应该在这里保存你的一些数据,因为这个时候你的程序的优先级降低，有可能被系统收回。在这里保存的数据，应该在 onResume里读出来，注意：这个方法里做的事情时间要短，因为下一个activity不会等到这个方法完成才启动

**onstop:**

变得不可见，被下一个activity覆盖了

**onDestroy:**

这是activity被干掉前最后一个被调用方法了，可能是外面类调用finish方法或者是系统为了节省空间将它暂时性的干掉，可以用 isFinishing()来判断它，如果你有一个Progress Dialog在线程中转动，请在onDestroy里把他cancel掉，不然等线程结束的时候，调用Dialog的cancel方法会抛异常的。

onPause，onstop， onDestroy，三种状态下 activity都有可能被系统干掉为了保证程序的正确性，你要在onPause()里写上持久层操作的代码，将用户编辑的内容都保存到存储介质上（一般 都是数据库）。实际工作中因为生命周期的变化而带来的问题也很多，比如你的应用程序起了新的线程在跑，这时候中断了，你还要去维护那个线程，是暂停还是杀掉还是数据回滚，因为Activity可能被杀掉，所以线程中使用的变量和一些界面元素就千万要注意了，一般都是采用Android的消息机制 [Handler,Message]来处理多线程和界面交互的问题。

**二 让Activity变成一个窗口：Activity属性设定**

应用程序是一个漂浮在手机主界面的东西，只需要设置一下Activity的主题就可以了在AndroidManifest.xml 中定义 Activity的地方一句话：

Xml代码

android

:theme="@android:style/Theme.Dialog"

android:theme="@android:style/Theme.Dialog"

这就使你的应用程序变成对话框的形式弹出来了，或者 Xml代码

android:theme="@android:style/Theme.Translucent"

android:theme="@android:style/Theme.Translucent"

就 变成半透明的，[友情提示-.-]类似的这种activity的属性可以在android.R.styleable 类的AndroidManifestActivity 方法中看到，AndroidManifest.xml中所有元素的属性的介绍都可以参考这个类android.R.styleable

上面说的是属性名称，具体有什么值是在android.R.style中

可以看到，比如这个"@android:style/Theme.Dialog"就对应于android.R.style.Theme\_Dialog ,（'\_'换成'.' <--注意：这个是文章内容不是笑脸）就可以用在描述文件中了,找找类定义和描述文件中的对应关系就都明白了。

**三 你后台的Activity被系统回收怎么办：onSaveInstanceState**

当你的程序中某一个Activity A 在运行时中，主动或被动地运行另一个新的Activity B 这个时候A会执行

Java代码

Public void onSaveInstanceState(Bundle outState) {

super.onSaveInstanceState(outState);

outState.putLong("id", 1234567890);

}

public void onSaveInstanceState(Bundle outState); B 完成以后又会来找A, 这个时候就有两种情况，一种是A被回收，一种是没有被回收，被回 收的A就要重新调用onCreate()方法，不同于直接启动的是这回 onCreate()里是带上参数 savedInstanceState，没被收回的就还是onResume就好了。

savedInstanceState是一个Bundle对象，你基本上可以把他理解为系统帮你维护的一个Map对象。在onCreate()里你可能会用到它，如果正常启动onCreate就不会有它，所以用的时候要判断一下是否为空。

Java代码

if(savedInstanceState !=null){

long id =savedInstanceState.getLong("id");

}

if(savedInstanceState !=null); 就像官方的Notepad教程里的情况，你正在编辑某一个note，突然被中断，那么就把这个note的id记住，再起来的时候就可以根据这个id去把那个note取出来，程序就完整一些。这也是看你的应用需不需要保存什么，比如你的界面就是读取一个列表，那就只需要记住滚动条的位置...

**四 调用与被调用：我们的通信使者Intent**

要 说Intent了，Intent就是这个这个意图，应用程序间Intent进行交流，打个电话啦，来个电话啦都会发Intent, 这个是Android架构的松耦合的精髓部分，大大提高了组件的复用性，比如你要在你的应用程序中点击按钮，给某人打电话，很简单啊，看下代码先：

Java代码 :

Intent intent = new Intent();

intent.setAction(Intent.ACTION\_CALL);

intent.setData(Uri.parse("tel:"+ number));

startActivity(intent);

扔出这样一个意图，系统看到了你的意图就唤醒了电话拨号程序，打出来电话。什么读联系人，发短信啊，邮件啊，统统只需要扔出intent就好了，这个部分设计地确实很好啊。那Intent通过什么来告诉系统需要谁来接受他呢？

通常使用Intent有两种方法，第一种是直接说明需要哪一个类来接收代码如下:

Java代码

Intent intent = new Intent(this,MyActivity.class);

intent.getExtras().putString("id","1");

startActivity(intent);

Intent intent = new Intent(this,MyActivity.class);intent.getExtras().putString("id","1");

tartActivity(intent);

第一种方式很明显，直接指定了MyActivity为接受者,并且传了一些数据给MyActivity，在MyActivity里可以用getIntent()来的到这个intent和数据。

第二种就需要先看一下AndroidMenifest中的intentfilter的配置了

Xml代码

<intent-filter>

<action android:name="android.intent.action.VIEW" />

<action android:value="android.intent.action.EDIT" />

<action android:value="android.intent.action.PICK" />

<category android:name="android.intent.category.DEFAULT"/>

<data android:mimeType="vnd.android.cursor.dir/vnd.google.note" />

</intent-filter>

<intent-filter>

<action android:name="android.intent.action.VIEW"/>

<action android:value="android.intent.action.EDIT" />

<action android:value="android.intent.action.PICK" />

<category android:name="android.intent.category.DEFAULT" />

<data android:mimeType="vnd.android.cursor.dir/vnd.google.note" />

</intent-filter>

这里面配置用到了action, data, category这些东西， action其实就是一个意图的字符串名称。上面这段intent-filter的配置文件说明了这个Activity可以接受不同的 Action，当然相应的程序逻辑也不一样咯,提一下那个 mimeType,他是在ContentProvider里定义的，你要是自己实现一个ContentProvider就知道了，必须指定 mimeType才能让数据被别人使用。

总结：调用别的界面不是直接new那个界面，而是通过扔出一个intent，让系统帮你去调用那个界面，这样就多么松藕合啊，而且符合了生命周期被系统管理的原则。

## 面试官最直接的问题（高频率）

一，listview你是怎么优化的。

二，view的刷新，之前说过

三，IPC及原理

四，Android多线程

五，Android为什么要设计4大组件，他们之间的联系，不设计行不行（主要是为了实现MVC模式，然而java中最难的模式也是这个，很少有产品能将这个模式做得很好

六，service的周期，activity的周期，谈下你对Android内部应用的了解，比如他做电话，以及联系人等等应用。框架层有很多东西还是多看看，熟悉Android怎么做的，不管你做应用程开发还是应用框架层开发很有好处的。

在就是你项目经验，突出你遇到什么难点，然后是怎么解决的！尽量将每个技术点凸显出来，当然面试官有时候会为了体现你是否真正做过，他会问你，你在这个应用中做那个模块，用了多少个类之类的问题。

偶尔有的面试官会问你，你用过Android自带的单元测试了没，怎么用的？

当然我面试过很多家单位，有的是做平板，手机，数字电视，有的是做出个erp之类的客户端等等，出于前面的三个，基本上都是将Android的全部改掉，如果真正要做Android的话，大家要学的还很多。

总 之，一句话，什么样的面试官都有，去面试的时候要做好一切心理准备，不管是技术还是基础都得扎实。一个人的交谈能力也很重要，总之不是非常标准的普通话， 最起码你说的得让别人听得懂，而且得把面试官讲得非常彻底，这样你获得offer的机会更大，谈工资也有优势~~当然曾经一家公司的面试官跟我说过，技术 是不惜钱的，只要你有能力，多少钱他都请。

1.View如何刷新？

2.DDMS与TraceView的区别？

3.activity被回收了怎么办？

4.在Java中如何引入C语言？

参考答案 :

1.View 可以调用invalidate()和postInvalidate()这两个方法刷新

2.DDMS是一个程序执行查看器，在里面你可以看见线程和堆栈等信息，TraceView是程序性能分析器

3.activity回收了，那就只有另起了

4.java调用C语言程序，可以用JNI接口来实现

上面答案仅作参考，毕竟我个人能力也有限嘛，难免会回答错误，呵呵.....

解答：

1.View受系统刷新（系统内部有个循环，监控事件、做业务处理、绘UI），可以用postInvalidate()促使系统刷新。

2.不知道

3.Activity生命周期如果被系统destroy了，也就是说的回收，只有从新start了

4.通过JNI调用。