

2014 面试题目及其答案

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 dalvik 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 160，160 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 Contro

9、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。

简单地说一下为什么是这样: 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, 是吧?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, 一般我都是采用 **Android** 的消息机制 [**Handler,Message**]来处理多线程和界面交互的问题。这个我后面会讲一些, 最近因为这些东西头已经很大了, 等我理清思绪再跟大家分享。 **ller** 改变了 **View**, **View** 会 从潜在的 **Model** 中获取数据来刷新自己。

10、让 **Activity** 变成一个窗口: **Activity** 属性设定

讲点轻松的吧,可能有人希望做出来的应用程序是一个漂浮在手机主界面的东西, 那么

很简单你只需要设置一下 Activity 的主题就可以了在 AndroidManifest.xml 中定义 Activity 的地方一句话:

Xml 代码

```
1. 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

```
1. 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。

11、 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

```
1. public  
2. void onSaveInstanceState(Bundle outState) {  
3. super.onSaveInstanceState(outState);  
4. outState.putLong("id", 1234567890);  
5. }
```

public

```
void onSaveInstanceState(Bundle outState) {  
  
super.onSaveInstanceState(outState);
```

```
outState.putLong("id", 1234567890);  
  
}
```

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

```
1. if(savedInstanceState != null){  
  
2. long id = savedInstanceState.getLong("id");  
  
3. }  
  
if(savedInstanceState != null){  
  
long id = savedInstanceState.getLong("id");  
  
}
```

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, 那就不需要特殊记住什么, 哦, 没准你需要记住滚动条的位置...

12、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

```
1. Intent intent = new Intent();  
  
2. intent.setAction(Intent.ACTION_CALL);
```

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

```
4. startActivity(intent);
```

```
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

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

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

```
3. tartActivity(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>
```

这里面配置用到了 **action**, **data**, **category** 这些东西, 那么聪明的你一定想到 **intent** 里也会有这些东西, 然后一匹配不就找到接收者了吗?

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。

不知道原理说明白没, 总结一句, 就是你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, 这样就多么松耦合啊, 而且符合了生命周期被系统管理的原则。

想知道 category 都有啥，Android 为你预先定制好的 action 都有啥等等，请亲自访问
官方链接 Intent

ps:想知道怎么调用系统应用程序的同学，可以仔细看一下你的 logcat，每次运行一个程序的时候是不是有一些信息比如：

```
Starting activity: Intent
{ action=android.intent.action.MAINcategories={android.intent.category.LAUNCHER}
flags=0x10200000comp={com.android.camera/com.android.camera.GalleryPicker} }
```

再对照一下 Intent 的一些 set 方法，就知道怎么调用咯，希望你喜欢：)

13、什么是 ANR 如何避免它？

答：ANR：Application Not Responding，五秒

在 Android 中，活动管理器和窗口管理器这两个系统服务负责监视应用程序的响应。当出现下列情况时，Android 就会显示 ANR 对话框了：

对输入事件(如按键、触摸屏事件)的响应超过 5 秒

意向接受器(intentReceiver)超过 10 秒钟仍未执行完毕

Android 应用程序完全运行在一个独立的线程中(例如 main)。这就意味着，任何在主线程中运行的，需要消耗大量时间的操作都会引发 ANR。因为此时，你的应用程序已经没有机会去响应输入事件和意向广播(Intent broadcast)。

因此，任何运行在主线程中的方法，都要尽可能的只做少量的工作。特别是活动生命周期中的重要方法如 onCreate()和 onResume()等更应如此。潜在的比较耗时的操作，如访问网络和数据库；或者是开销很大的计算，比如改变位图的大小，需要在一个单独的子线程中完成(或者是使用异步请求，如数据库操作)。但这并不意味着你的主线程需要进入阻塞状态已等待子线程结束 -- 也不需要调用 Thread.wait()或者 Thread.sleep()方法。取而代之的是，主线程为子线程提供一个句柄(Handler)，让子线程在即将结束的时候调用它(xing:可以参看 Snake 的例子，这种方法与以前我们所接触的有所不同)。使用这种方法涉及你的应用程序，

能够保证你的程序对输入保持良好的响应,从而避免因输入事件超过 5 秒钟不被处理而产生的 ANR。这种实践需要应用到所有显示用户界面的线程,因为他们都面临着同样的超时问题。

14、什么情况会导致 Force Close ?如何避免?能否捕获导致其的异常?

答:一般像空指针啊,可以看起 logcat, 然后对应到程序中 来解决错误

15、Android 本身的 api 并未声明会抛出异常, 则其在运行时有无可能抛出 runtime 异常, 你遇到过吗? 诺有的话会导致什么问题? 如何解决?

答: 会, 如 nullPointerException, textView 没有初始化时, 却调用了其方法。打开控制台, 查看 logcat 找出异常并修改。

16、简要解释一下 activity、intent 、intent filter、service、Broadcast、BroadcastReceiver

答: 一个 activity 呈现了一个用户可以操作的可视化用户界面

一个 service 不包含可见的用户界面, 而是在后台无限地运行

可以连接到一个正在运行的服务中, 连接后, 可以通过服务中暴露出来的借口与其进行通信

一个 broadcast receiver 是一个接收广播消息并作出回应的 component, broadcast receiver 没有界面

intent:content provider 在接收到 ContentResolver 的请求时被激活。

activity, service 和 broadcast receiver 是被称为 intents 的异步消息激活的。

一个 intent 是一个 Intent 对象, 它保存了消息的内容。对于 activity 和 service 来说, 它指定了请求的操作名称和待操作数据的 URI

Intent 对象可以显式的指定一个目标 component。如果这样的话, android 会找到这个 component(基于 manifest 文件中的声明)并激活它。但如果一个目标不是显式指定的, android 必须找到响应 intent 的最佳 component。

它是通过将 Intent 对象和目标的 intent filter 相比较来完成这一工作的。一个 component 的 intent filter 告诉 android 该 component 能处理的 intent。intent filter 也是在 manifest 文件中声明的。

17、IntentService 有何优点?

答: IntentService 的好处

* Activity 的进程，当处理 Intent 的时候，会产生一个对应的 Service

* Android 的进程处理器现在会尽可能的不 kill 掉你

* 非常容易使用

18. 横竖屏切换时候 activity 的生命周期?

1、不设置 Activity 的 android:configChanges 时，切屏会重新调用各个生命周期，切横屏时会执行一次，切竖屏时会执行两次

2、设置 Activity 的 android:configChanges="orientation"时，切屏还是会重新调用各个生命周期，切横、竖屏时只会执行一次

3、设置 Activity 的 android:configChanges="orientation|keyboardHidden"时，切屏不会重新调用各个生命周期，只会执行 onConfigurationChanged 方法

19. 如何将 SQLite 数据库(dictionary.db 文件)与 apk 文件一起发布?

解答：可以将 dictionary.db 文件复制到 Eclipse Android 工程中的 res aw 目录中。所有在 res aw 目录中的文件不会被压缩，这样可以直接提取该目录中的文件。可以将 dictionary.db 文件复制到 res aw 目录中

20. 如何将打开 res aw 目录中的数据库文件?

解答：在 Android 中不能直接打开 res aw 目录中的数据库文件，而需要在程序第一次启动时将该文件复制到手机内存或 SD 卡的某个目录中，然后再打开该数据库文件。复制的基本方法是使用 getResources().openRawResource 方法获得 res aw 目录中资源的 InputStream 对象，然后将该 InputStream 对象中的数据写入其他的目录中相应文件中。在 Android SDK 中可以使用 SQLiteDatabase.openOrCreateDatabase 方法来打开任意目录中的 SQLite 数据库文件。

21. Android 引入广播机制的用意?

答：a:从 MVC 的角度考虑(应用程序内)

其实回答这个问题的时候还可以这样问，android 为什么要有那 4 大组件，现在的移动开发模型基本上也是照搬的 web 那一套 MVC 架构，只不过是改了点嫁妆而已。android 的四大组件本质上就是为了实现移动或者说嵌入式设备上的 MVC 架构，它们之间有时候是一种相互依存的关系，有时候又是一种补充关系，引入广播机制可以方便几大组件的信息和数据

b: 程序间互通消息(例如在自己的应用程序内监听系统来电)

c: 效率上(参考 UDP 的广播协议在局域网的方便性)

d: 设计模式上(反转控制的一种应用, 类似监听者模式)

1. 请描述下 Activity 的声明周期。

onCreate->onStart->onResume->onPause->onStop->onRestart->onDestroy

2. 如果后台的 Activity 由于某种原因被系统回收, 如何在回收之前保存当前状态。

onSaveInstanceState().

程序中的某一个 Activity A 在运行时, 主动或被动的运行另一个新的 Activity B, 这个时候 A 会执行 onSaveInstanceState()。B 完成以后又回来找 A, 这个时候有两种情况: 一是 A 被回收, 二是 A 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()里是带上了参数 savedInstanceState; 而没被回收的就直接执行 onResume(), 跳过 onCreate()。

3. 如何将一个 Activity 设置成窗口样式。

在 AndroidManifest.xml 中 Activity 定义处添加
android:theme="@android:style/Theme.Dialog" 或
android:theme="@android:style/Theme.Translucent"。

4. 如何退出 Activity? 如何安全退出已调用多个 Activity 的 Application?

1.Activity.finish();

也可以用 killProcess()和 System.exit()这样的方法

2.用 ActivityManager 的 restartPackage 方法

5. 请介绍下 android 大众常用的五种布局。

FrameLayout (框架布局)

只可以有一个控件, 并且不能设计这个控件的位置, 控件会放在左上角

LinearLayout (线性布局)

一行只能控制一个控件的线性布局, 所以当有很多控件需要在一个界面中列出时, 可以用

LinearLayout 布局

AbsoluteLayout (绝对布局)

可以放置多个控件，并且可以自己定义控件的 x,y 的位置

RelativeLayout（相对布局）

比如要在一行上显示多个控件，这时就要用到相对布局

TableLayout（表格布局）

将子元素的位置分配到行或列中，一个 TableLayout 由许多的 TableRow 组成

6. 请介绍下 android 的数据存储方式。

SharedPreferences 方式

用来存储” key-value paires” 格式的数据，它是一个轻量级的键值存储机制，只可以存储基本数据类型,可以通过它保存一些上次用户所做的修改操作

文件存储方式

它通过 FileInputStream 和 FileOutputStream 对文件进行操作。但是在 Android 中，文件是一个应用程序私有的，一个应用程序无法读写其他应用程序的文件，如果要共享数据就用

Content Providers

SQLite 数据库方式

Android 提供的一个标准的数据库，支持 SQL 语句

内容提供器（Content provider）方式

是所有应用程序之间数据存储和检索的桥梁，它的作用是使得各个应用程序之间实现数据共享

网络存储方式

通过网络来存储和获取数据

7. 请介绍下 ContentProvider 是如何实现数据共享的。

Android 提供了 ContentProvider，一个程序可以通过实现一个 ContentProvider 的抽象接口将自己的数据完全暴露出去，而且 ContentProviders 是以类似数据库中表的方式将数据暴露，也就是说 ContentProvider 就像一个“数据库”。那么外界获取其提供的数据，也就应该与从数据库中获取数据的操作基本一样，只不过是采用 URI 来表示外界需要访问的“数据库”。外部访问通过 ContentResolver 去访问并操作这些被暴露的数据。

8. 如何启用 Service，如何停用 Service。

```
Intent intent = new Intent();  
intent.setClass(this,cba.class);  
startService(intent);  
stopService(intent);
```

9. 注册广播有几种方式，这些方式有何优缺点？ Android 引入广播机制的用意。

2 种方式

1. 在 androidmanifest.xml 中注册
2. <receiver>
3. <intent-filter>

4. `<action android:name = "android.intent.action.PICK"/>`
5. `</intent-filter>`
6. `</receiver>`
7. 缺点：常驻型，占资源比较大
- 8.
9. `registerReceiver(receiver,filter);` BroadcastReceiver 更新 UI 一般用这种方法

10. 请解释下在单线程模型中 Message,Handler,Message Queue,Looper 之间的关系。

Message

Message 消息，理解为线程间交流的信息，处理数据后台线程需要更新 UI，则发送 Message 内含一些数据给 UI 线程。

2. Handler

Handler 处理者，是 Message 的主要处理者，负责 Message 的发送，Message 内容的执行处理。后台线程就是通过传进来的 Handler 对象引用来 `sendMessage(Message)`。而使用 Handler，需要 implement 该类的 `handleMessage(Message)` 方法，它是处理这些 Message 的操作内容，例如 Update UI。通常需要子类化 Handler 来实现 `handleMessage` 方法。

Message Queue

Message Queue 消息队列，用来存放通过 Handler 发布的消息，按照先进先出执行。

每个 message queue 都会有一个对应的 Handler。Handler 会向 message queue 通过两种方法发送消息：`sendMessage` 或 `post`。这两种消息都会插在 message queue 队尾并按先进先出执行。但通过这两种方法发送的消息执行的方式略有不同：通过 `sendMessage` 发送的是一个 message 对象，会被 Handler 的 `handleMessage()` 函数处理；而通过 `post` 方法发送的是一个 runnable 对象，则会自己执行。

4. Looper

Looper 是每条线程里的 Message Queue 的管家。Android 没有 Global 的 Message Queue，而 Android 会自动替主线程(UI 线程)建立 Message Queue，但在子线程里并没有建立 Message Queue。所以调用 `Looper.getMainLooper()` 得到的主线程的 Looper 不为 NULL，但调用 `Looper.myLooper()` 得到当前线程的 Looper 就有可能为 NULL。

对于子线程使用 Looper，API Doc 提供了正确的使用方法：

11. AIDL 的全程是什么？如何工作？能处理哪些类型的数据？

AIDL 的英文全称是 Android Interface Define Language

当 A 进程要去调用 B 进程中的 service 时，并实现通信，我们通常都是通过 AIDL 来操作的 A 工程：

首先我们在 `net.blogjava.mobile.aidlservice` 包中创建一个 `RemoteService.aidl` 文件，在里面我们自定义一个接口，含有方法 `get`。ADT 插件会在 `gen` 目录下自动生成一个 `RemoteService.java` 文件，该类中含有一个名为 `RemoteService.stub` 的内部类，该内部类中含有 `aidl` 文件接口的 `get` 方法。

说明一：`aidl` 文件的位置不固定，可以任意

然后定义自己的 `MyService` 类，在 `MyService` 类中自定义一个内部类去继承 `RemoteService.stub` 这个内部类，实现 `get` 方法。在 `onBind` 方法中返回这个内部类的对象，系统会自动将这个对象封装成 `IBinder` 对象，传递给他的调用者。

其次需要在 AndroidManifest.xml 文件中配置 MyService 类，代码如下：

```
<!-- 注册服务 -->
<service android:name=".MyService">
    <intent-filter>
        <!-- 指定调用 AIDL 服务的 ID -->
        <action android:name="net.blogjava.mobile.aidlservice.RemoteService" />
    </intent-filter>
</service>
```

为什么要指定调用 AIDL 服务的 ID,就是要告诉外界 MyService 这个类能够被别的进程访问,只要别的进程知道这个 ID,正是有了这个 ID,B 工程才能找到 A 工程实现通信。

说明: AIDL 并不需要权限

B 工程:

首先我们要将 A 工程中生成的 RemoteService.java 文件拷贝到 B 工程中,在 bindService 方法中绑定 aidl 服务

绑定 AIDL 服务就是将 RemoteService 的 ID 作为 intent 的 action 参数。

说明: 如果我们单独将 RemoteService.aidl 文件放在一个包里,那个在我们将 gen 目录下的该包拷贝到 B 工程中。如果我们将 RemoteService.aidl 文件和我们的其他类存放在一起,那么我们在 B 工程中就要建立相应的包,以保证 RemoteService.java 文件的报名正确,我们不能修改 RemoteService.java 文件

```
bindService(new Intent("net.blogjava.mobile.aidlservice.RemoteService"),
serviceConnection, Context.BIND_AUTO_CREATE);
```

ServiceConnection 的 onServiceConnected(ComponentName name, IBinder service)方法中的 service 参数就是 A 工程中 MyService 类中继承了 RemoteService.stub 类的内部类的对象。

Android 采用 RPC 的方式来实现 (remote procedure call) 远程通信,并且 Android 通过接口定义语言 AIDL 来生成两个进程之间互相访问的代码。例如,你在 Activity 里的代码需要访问 Service 中的一个方法,那么就可以通过这种方法来实现了。

- ? 创建一个 AIDL 文件 (参见代码)
- ? 实现 AIDL 文件生成的 Java 接口 (参见代码)
- ? 将你的接口暴露给客户端
- ? 客户端调用

12. 请解释下 Android 程序运行时权限与文件系统权限的区别。

Android 程序运行 是虚拟机 Dalvik(android 授权)

文件系统 是 linux 内核 授权

13. 系统上安装了多种浏览器,能否指定某浏览器访问指定页面?

```
Intent intent =newIntent();
intent.setAction("android.intent.action.VIEW");
Uri content_url =Uri.parse("http://www.163.com");
intent.setData(content_url);
```

```
intent.setClassName("com.android.browser", "com.android.  
browser.BrowserActivity");  
startActivity(intent);
```

14. 有一个一维整型数组 `int[] data` 保存的是一张宽为 `width`, 高为 `height` 的图片像素值信息。请写一个算法, 将该图片所有的白色不透明(0xffffffff)像素点的透明度调整为 50%。

答: 像素数组 `int[] data` 里就是一个个像素的颜色值, 你循环和 0xffffffff 比较, 如果相等, 那么就把这个值换成 0x88ffffff 就行了, 这样像素数组里所有的白色不透明(0xffffffff)都变成的半透明

15. 你如何评价 Android 系统? 优缺点。

优点: 开源特性, 得到众多厂商支持。

软件发展很快。

界面 UI, 系统优化不错。

缺点: 版本过多, 升级过快。

用户体验不一致。

16. android 中的动画有哪几类, 他们的特点和区别是什么?

两种,

一种是 Tween 动画(补间动画),

一种是 Frame 动画(逐帧动画)。

Tween 动画: 使试图组件移动, 放大, 缩小以及产生透明度的变化。

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。

17. 横竖屏切换时 activity 的生命周期

不设置 Activity 的 `android:configChanges` 时, 切屏会重新调用各个生命周期, 切横屏时会执行一次, 切竖屏时会执行两次。

设置 Activity 的 `android:configChanges="orientation"` 时, 切横, 竖屏时生命周期只会执行一次。

设置 Activity 的 `android:configChanges="orientation|keyboardHidden"` 时, 切屏不会重新调用声明周期, 只会执行 `onConfigurationChanged` 方法。

进程与线程

当一个程序开始运行时, 它就是一个进程, 进程包含运行中的程序和程序所使用的内存和系统资源。而一个进程又是由多个线程组成的。引入线程优点是易于调度, 提供开发效率, 通过线程可以方便有效的实现并发, 进程可创建多个线程来执行同一个程序的不同部分, 开销小, 创建线程比创建进程要快, 所需开销很少。

宏就是把一系列常用的操作作为一个整体, 保存起来, 以后用的时候直接通过一定方式用就是了

先说宏和函数的区别:

1. 宏做的是简单的字符串替换(注意是字符串的替换, 不是其他类型参数的替换), 而函数的参

数的传递,参数是有数据类型的,可以是各种各样的类型.

2. 宏的参数替换是不经计算而直接处理的,而函数调用是将实参的值传递给形参,既然是值,自然是计算得来的.

3. 宏在编译之前进行,即先用宏体替换宏名,然后再编译的,而函数显然是编译之后,在执行时,才调用的.因此,宏占用的是编译的时间,而函数占用的是执行时的时间.

4. 宏的参数是不占内存空间的,因为只是做字符串的替换,而函数调用时的参数传递则是具体变量之间的信息传递,形参作为函数的局部变量,显然是占用内存的.

5. 函数的调用是需要付出一定的时空开销的,因为系统在调用函数时,要保留现场,然后转入被调用函数去执行,调用完,再返回主调函数,此时再恢复现场,这些操作,显然在宏中是没有的.定义一个宏替代两个输入数的乘积

怎么写?

```
#include<stdio.h>
#define MULTIPLY(x,y) ((x)*(y))
int main()
{
    int x,y,z;
    scanf("%d%d",&x,&y);
    z=MULTIPLY(x,y);
    printf("%d",z);
    return 0;
}
```

--

Android 面试题 经典

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 dalvik 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, 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

android 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

android 的官方建议应用程序的开发采用 mvc 模式。何谓 mvc?

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

1 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。

1 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。

1 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，想用用户出发的相关事件，交给 m 哦得了处理。

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

1)视图层 (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，当然，如何你对 android 了解的比较的多了话，就一定可以想到在 android 中也可以使用 javascript+html 等的方式作为 view 层，当然这里需要进行 java 和 javascript 之间的通信，幸运的是，android 提供了它们之间非常方便的通信实现。

2)控制层 (controller): android 的控制层的重任通常落在了众多的 activity 的肩上，这句话也就暗含了不要在 activity 中写代码，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 activity 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。

9、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。

简单地说一下为什么是这样: 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();
}

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，是吧?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。

10、让 Activity 变成一个窗口：Activity 属性设定

讲点轻松的吧,可能有人希望做出来的应用程序是一个漂浮在手机主界面的东西，那么很

简单你只需要设置一下 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。

11、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);
```

```
}
```

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");
}
```

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, 那就不需要特殊记住什么, 哦, 没准你需要记住滚动条的位置...

12、 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);
```

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 AndroidManifest 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>
```

这里面配置用到了 `action`, `data`, `category` 这些东西, 那么聪明的你一定想到 `intent` 里也会有这些东西, 然后一匹配不就找到接收者了吗?

`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。

不知道原理说明白没, 总结一句, 就是你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, 这样就多么松耦合啊, 而且符合了生命周期被系统管理的原则。

想知道 `category` 都有啥, `Android` 为你预先定制好的 `action` 都有啥等等, 请亲自访问官方链接 [Intent](#)

ps:想知道怎么调用系统应用程序的同学, 可以仔细看一下你的 `logcat`, 每次运行一个程序的时候是不是有一些信息比如:

```
Starting          activity:          Intent
{          action=android.intent.action.MAINcategories={ android.intent.category.LAUNCHER }
  flags=0x10200000comp={ com.android.camera/com.android.camera.GalleryPicker } }
```

再对照一下 `Intent` 的一些 `set` 方法, 就知道怎么调用咯, 希望你喜欢:)

13. 如何退出 `Activity`?如何安全退出已调用多个 `Activity` 的 `Application`?

对于单一 `Activity` 的应用来说, 退出很简单, 直接 `finish()` 即可。

当然, 也可以用 `killProcess()` 和 `System.exit()` 这样的方法。

但是, 对于多 `Activity` 的应用来说, 在打开多个 `Activity` 后, 如果想在最后打开的 `Activity` 直接退出, 上边的方法都是没有用的, 因为上边的方法都是结束一个 `Activity` 而已。

当然, 网上也有人说可以。

就好像有人问, 在应用里如何捕获 `Home` 键, 有人就会说用 `keyCode` 比较 `KEYCODE_HOME` 即可, 而事实上如果不修改 `framework`, 根本不可能做到这一点一样。

所以, 最好还是自己亲自试一下。

那么, 有没有办法直接退出整个应用呢?

在 2.1 之前, 可以使用 `ActivityManager` 的 `restartPackage` 方法。

它可以直接结束整个应用。在使用时需要权限 `android.permission.RESTART_PACKAGES`。

注意不要被它的名字迷惑。

可是，在 2.2，这个方法失效了。

在 2.2 添加了一个新的方法，`killBackgroundProcesses()`，需要权限 `android.permission.KILL_BACKGROUND_PROCESSES`。

可惜的是，它和 2.2 的 `restartPackage` 一样，根本起不到应有的效果。

另外还有一个方法，就是系统自带的应用程序管理里，强制结束程序的方法，`forceStopPackage()`。

它需要权限 `android.permission.FORCE_STOP_PACKAGES`。

并且需要添加 `android:sharedUserId="android.uid.system"` 属性

同样可惜的是，该方法是非公开的，他只能运行在系统进程，第三程序无法调用。

因为需要在 `Android.mk` 中添加 `LOCAL_CERTIFICATE := platform`。

而 `Android.mk` 是用于在 `Android` 源码下编译程序用的。

从以上可以看出，在 2.2，没有办法直接结束一个应用，而只能用自己的办法间接办到。

现提供几个方法，供参考：

1、抛异常强制退出：

该方法通过抛异常，使程序 Force Close。

验证可以，但是，需要解决的问题是，如何使程序结束掉，而不弹出 Force Close 的窗口。

2、记录打开的 Activity：

每打开一个 Activity，就记录下来。在需要退出时，关闭每一个 Activity 即可。

3、发送特定广播：

在需要结束应用时，发送一个特定的广播，每个 Activity 收到广播后，关闭即可。

4、递归退出

在打开新的 Activity 时使用 `startActivityForResult`，然后自己加标志，在 `onActivityResult` 中处理，递归关闭。

除了第一个，都是想办法把每一个 Activity 都结束掉，间接达到目的。

但是这样做同样不完美。

你会发现，如果自己的应用程序对每一个 Activity 都设置了 `nosensor`，在两个 Activity 结束的间隙，`sensor` 可能有效了。

但至少，我们的目的达到了，而且没有影响用户使用。

为了编程方便，最好定义一个 Activity 基类，处理这些共通问题。

摘自：<http://blog.csdn.net/debug2/archive/2011/02/18/6193644.aspx>

14. 请介绍下 Android 中常用的五种布局。

1、LinearLayout - 线性布局。

`orientation` - 容器内元素的排列方式。`vertical`: 子元素们垂直排列；`horizontal`: 子元素们水平排列

`gravity` - 内容的排列形式。常用的有 `top`, `bottom`, `left`, `right`, `center` 等

2、 AbsoluteLayout - 绝对布局。

layout_x - x 坐标。以左上角为顶点

layout_y - y 坐标。以左上角为顶点

3、 TableLayout - 表格式布局

表格布局主要以行列的形式来管理子控件，其中每一行即一个 TableRow 对象，每个 TableRow 对象可以添加子控件，并且每加入一个空间即相当于添加了一列

4、 RelativeLayout - 相对布局。

layout_centerInParent - 将当前元素放置到其容器内的水平方向和垂直方向的中央位置（类似的属性有：layout_centerHorizontal, layout_alignParentLeft 等）

layout_marginLeft - 设置当前元素相对于其容器的左侧边缘的距离

layout_below - 放置当前元素到指定的元素的下面

layout_alignRight - 当前元素与指定的元素右对齐

5、 FrameLayout - 层叠布局。以左上角为起点，将 FrameLayout 内的元素一层覆盖一层地显示，在帧布局中，先添加的图片会被后添加的图片覆盖。

摘自：<http://javalover00000.javaeye.com/blog/851266>

15. 请介绍下 Android 的数据存储方式。

Android 提供了 5 种方式存储数据：

- 1、使用 SharedPreferences 存储数据；
- 2、文件存储数据；
- 3、SQLite 数据库存储数据；
- 4、使用 ContentProvider 存储数据；
- 5、网络存储数据；

Android 中的数据存储都是私有的，其他应用程序都是无法访问的，除非通过 ContentResolver 获取其他程序共享的数据。

摘自：<http://www.moandroid.com/?p=319>

16. 请介绍下 ContentProvider 是如何实现数据共享的。

一个程序可以通过实现一个 Content provider 的抽象接口将自己的数据完全暴露出去，而且 Content providers 是以类似数据库中表的方式将数据暴露。Content providers 存储和检索数据，通过它可以所有的应用程序访问到，这也是应用程序之间唯一共享数据的方法。要想使应用程序的数据公开化，可通过 2 种方法：创建一个属于你自己的 Content provider 或者将你的数据添加到一个已经存在的 Content provider 中，前提是有相同数据类型并且有写入 Content provider 的权限。

如何通过一套标准及统一的接口获取其他应用程序暴露的数据？Android 提供了 ContentResolver，外界的程序可以通过 ContentResolver 接口访问 ContentProvider 提供的数据。

参考：<http://www.moandroid.com/?p=319>

17. 如何启用 Service，如何停用 Service。

1. 第一种是通过调用 Context.startService()启动，调用 Context.stopService()结束，startService()可以传递参数给 Service

2. 第二种方式是通过调用 Context.bindService()启动, 调用 Context.unbindService()结束, 还可以通过 ServiceConnection 访问 Service。

在 Service 每一次的开启关闭过程中, 只有 onStart 可被多次调用(通过多次 startService 调用), 其他 onCreate, onBind, onUnbind, onDestroy 在一个生命周期中只能被调用一次。

参考: <http://www.cnblogs.com/feisky/archive/2010/06/14/1758336.html>

18. 注册广播有几种方式, 这些方式有何优缺点?请谈谈 Android 引入广播机制的用意。android 中, 不同进程之间传递信息要用到广播, 可以有两种方式来实现。

第一种方式: 在 Manifest.xml 中注册广播, 是一种比较推荐的方法, 因为它不需要手动注销广播 (如果广播未注销, 程序退出时可能会出错)。

具体实现在 Manifest 的 application 中添加:

```
<receiver android:name=".mEvtReceiver">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
    </intent-filter>
</receiver>
```

上面两个 android:name 分别是广播名和广播的动作 (这里的动作是表示系统启动完成), 如果要自己发送一个广播, 在代码中为:

```
Intent i = new Intent("android.intent.action.BOOT_COMPLETED");
sendBroadcast(i);
```

这样, 广播就发出去了, 然后是接收。

接收可以新建一个类, 继承至 BroadcastReceiver, 也可以建一个 BroadcastReceiver 的实例, 然后得写 onReceive 方法, 实现如下:

```
protected BroadcastReceiver mEvtReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if (action.equals("android.intent.action.BOOT_COMPLETED")) {
            //Do something
        }
    }
};
```

第二种方式, 直接在代码中实现, 但需要手动注册注销, 实现如下:

```
IntentFilter filter = new IntentFilter();
filter.addAction("android.intent.action.BOOT_COMPLETED");
registerReceiver(mEvtReceiver, filter); //这时注册了一个 receiver ,名为 mEvtReceiver, 然后同样用上面的方法以重写 onReceive,
```

最后在程序的 onDestroy 中要注销广播, 实现如下:

```
@Override
public void onDestroy() {
    super.onDestroy();
}
```



```
unregisterReceiver(mPlayerEvtReceiver);  
}
```

Android 系统中的广播是广泛用于应用程序之间通信的一种手段，它类似于事件处理机制，不同的地方就是广播的处理是系统级别的事件处理过程（一般事件处理是控件级别的）。在此过程中仍然是离不开 Intent 对象，理解广播事件的处理过程，灵活运用广播处理机制，在关键之处往往能实现特别的效果，

在 Android 中如果要发送一个广播必须使用 sendBroadcast 向系统发送对其感兴趣的广播接收器中。

使用广播必须要有一个 intent 对象必设置其 action 动作对象

使用广播必须在配置文件中显式的指明该广播对象

每次接收广播都会重新生成一个接收广播的对象

在 Broadcast 中尽量不要处理太多逻辑问题，建议复杂的逻辑交给 Activity 或者 Service 去处理

转自：<http://www.dengdeng.name/u/deng/archives/2010/92.html>

<http://www.cnblogs.com/TerryBlog/archive/2010/08/16/1801016.html>

19. 请解释下在单线程模型中 Message、Handler、Message Queue、Looper 之间的关系。简单的说，Handler 获取当前线程中的 looper 对象，looper 用来从存放 Message 的 MessageQueue 中取出 Message，再有 Handler 进行 Message 的分发和处理

参见：<http://hi.baidu.com/???????/blog/item/4e576b7c58d742360cd7dac9.html>

<http://blog.csdn.net/xuxinyi/archive/2010/12/25/6097560.aspx>

20. AIDL 的全称是什么?如何工作?能处理哪些类型的数据?

AIDL 全称 Android Interface Definition Language（Android 接口描述语言）是一种借口描述语言；编译器可以通过 aidl 文件生成一段代码，通过预先定义的接口达到两个进程内部通信进程跨界对象访问的目的。AIDL 的 IPC 的机制和 COM 或 CORBA 类似，是基于接口的，但它是轻量级的。它使用代理类在客户端和实现层间传递值。如果要使用 AIDL，需要完成 2 件事情：1. 引入 AIDL 的相关类；2. 调用 aidl 产生的 class。理论上，参数可以传递基本数据类型和 String，还有就是 Bundle 的派生类，不过在 Eclipse 中，目前的 ADT 不支持 Bundle 做为参数，

具体实现步骤如下：

1、创建 AIDL 文件，在这个文件里面定义接口，该接口定义了可供客户端访问的方法和属性。

2、编译 AIDL 文件，用 Ant 的话，可能需要手动，使用 Eclipse plugin 的话，可以根据 aidl 文件自动生产 java 文件并编译，不需要人为介入。

3、在 Java 文件中，实现 AIDL 中定义的接口。编译器会根据 AIDL 接口，产生一个 JAVA 接口。这个接口有一个名为 Stub 的内部抽象类，它继承扩展了接口并实现了远程调用需要的几个方法。接下来就需要自己去实现自定义的几个接口了。

4、向客户端提供接口 ITaskBinder，如果写的是 service，扩展该 Service 并重载 onBind ()方

方法来返回一个实现上述接口的类的实例。

5、在服务器端回调客户端的函数。前提是当客户端获取的 **IBinder** 接口的时候,要去注册回调函数,只有这样,服务器端才知道该调用那些函数

AIDL 语法很简单,可以用来声明一个带一个或多个方法的接口,也可以传递参数和返回值。由于远程调用的需要,这些参数和返回值并不是任何类型.下面是些 AIDL 支持的数据类型:

1. 不需要 import 声明的简单 Java 编程语言类型(int,boolean 等)
2. String, CharSequence 不需要特殊声明
3. List, Map 和 Parcelables 类型, 这些类型内所包含的数据成员也只能是简单数据类型, String 等其他比支持的类型.

(另外: 我没尝试 Parcelables, 在 Eclipse+ADT 下编译不过, 或许以后会有所支持).

实现接口时有几个原则:

.抛出的异常不要返回给调用者. 跨进程抛异常处理是不可取的.

.IPC 调用是同步的。如果你知道一个 IPC 服务需要超过几毫秒的时间才能完成地话, 你应该避免在 Activity 的主线程中调用。也就是 IPC 调用会挂起应用程序导致界面失去响应. 这种情况应该考虑单起一个线程来处理.

.不能在 AIDL 接口中声明静态属性。

IPC 的调用步骤:

1. 声明一个接口类型的变量, 该接口类型在.aidl 文件中定义。
2. 实现 ServiceConnection。
3. 调用 ApplicationContext.bindService(),并在 ServiceConnection 实现中进行传递.
4. 在 ServiceConnection.onServiceConnected()实现中, 你会接收一个 IBinder 实例(被调用的 Service). 调用

YourInterfaceName.Stub.asInterface((IBinder)service)将参数转换为 YourInterface 类型。

5. 调用接口中定义的方法。你总要检测到 DeadObjectException 异常, 该异常在连接断开时被抛出。它只会被远程方法抛出。

6. 断开连接, 调用接口实例中的 ApplicationContext.unbindService()

参考: <http://buaadallas.blog.51cto.com/399160/372090>

21. 请解释下 Android 程序运行时权限与文件系统权限的区别。

apk 程序是运行在虚拟机上的,对应的是 Android 独特的权限机制,只有体现到文件系统上时才使用 linux 的权限设置。

android 系统有的权限是基于签名的。

具体参见: <http://blog.csdn.net/Zengyangtech/archive/2010/07/20/5749999.aspx>

22. 系统上安装了多种浏览器,能否指定某浏览器访问指定页面?请说明原由。

通过直接发送 Uri 把参数带过去,或者通过 manifest 里的 intentfilter 里的 data 属性

23. 有一个一维整型数组 int[]data 保存的是一张宽为 width, 高为 height 的图片像素值信息。请写一个算法, 将该图片所有的白色不透明(0xffffffff)像素点的透明度调整为 50%。

24. 什么是 ANR 如何避免它?

答: ANR: Application Not Responding, 五秒在 Android 中, 活动管理器和窗口管理器这两个系统服务负责监视应用程序的响应。当出现下列情况时, Android 就会显示 ANR 对话框了:

对输入事件(如按键、触摸屏事件)的响应超过 5 秒

意向接受器(intentReceiver)超过 10 秒钟仍未执行完毕

Android 应用程序完全运行在一个独立的线程中(例如 main)。这就意味着, 任何在主线程中运行的, 需要消耗大量时间的操作都会引发 ANR。因为此时, 你的应用程序已经没有机会去响应输入事件和意向广播(Intent broadcast)。

因此, 任何运行在主线程中的方法, 都要尽可能的只做少量的工作。特别是活动生命周期中的重要方法如 onCreate()和 onResume()等更应如此。潜在的比较耗时的操作, 如访问网络和数据库;或者是开销很大的计算, 比如改变位图的大小, 需要在一个单独的子线程中完成 (或者是使用异步请求, 如数据库操作)。但这并不意味着你的主线程需要进入阻塞状态已等待子线程结束 — 也不需要调用 Thread.wait()或者 Thread.sleep()方法。取而代之的是, 主线程为子线程提供一个句柄(Handler), 让子线程在即将结束的时候调用它(xing:可以参看 Snake 的例子, 这种方法与以前我们所接触的有所不同)。使用这种方法涉及你的应用程序, 能够保证你的程序对输入保持良好的响应, 从而避免因输入事件超过 5 秒钟不被处理而产生的 ANR。这种实践需要应用到所有显示用户界面的线程, 因为他们都面临着同样的超时问题。

25. 什么情况会导致 Force Close ?如何避免?能否捕获导致其的异常?

答: 一般像空指针啊, 可以看看 logcat, 然后对应到程序中 来解决错误

26. Android 本身的 api 并未声明会抛出异常, 则其在运行时有无可能抛出 runtime 异常, 你遇到过吗?诺有的话会导致什么问题?如何解决?

27. 简要解释一下 activity、 intent 、 intent filter、 service、 Broadcast、 BroadcastReceiver

答: 一个 activity 呈现了一个用户可以操作的可视化用户界面

一个 service 不包含可见的用户界面, 而是在后台无限地运行

可以连接到一个正在运行的服务中, 连接后, 可以通过服务中暴露出来的借口与其进行通信

一个 broadcast receiver 是一个接收广播消息并作出回应的 component, broadcast receiver 没有界面

intent:content provider 在接收到 ContentResolver 的请求时被激活。

activity, service 和 broadcast receiver 是被称为 intents 的异步消息激活的。

一个 intent 是一个 Intent 对象, 它保存了消息的内容。对于 activity 和 service 来说, 它指定了请求的操作名称和待操作数据的 URI

Intent 对象可以显式的指定一个目标 component。如果这样的话, android 会找到这个

component(基于 manifest 文件中的声明)并激活它。但如果一个目标不是显式指定的, android 必须找到响应 intent 的最佳 component。

它是通过将 Intent 对象和目标的 intent filter 相比较来完成这一工作的。一个 component 的 intent filter 告诉 android 该 component 能处理的 intent。intent filter 也是在 manifest 文件中声明的。

28、IntentService 有何优点?

答: IntentService 的好处

* Activity 的进程, 当处理 Intent 的时候, 会产生一个对应的 Service

* Android 的进程处理器现在会尽可能的不 kill 掉你

* 非常容易使用

29、横竖屏切换时候 activity 的生命周期?

1、不设置 Activity 的 android:configChanges 时, 切屏会重新调用各个生命周期, 切横屏时会执行一次, 切竖屏时会执行两次

2、设置 Activity 的 android:configChanges=" orientation" 时, 切屏还是会重新调用各个生命周期, 切横、竖屏时只会执行一次

3、设置 Activity 的 android:configChanges=" orientation|keyboardHidden" 时, 切屏不会重新调用各个生命周期, 只会执行 onConfigurationChanged 方法

30. 如何将 SQLite 数据库(dictionary.db 文件)与 apk 文件一起发布?

解答: 可以将 dictionary.db 文件复制到 Eclipse Android 工程中的 res aw 目录中。所有在 res aw 目录中的文件不会被压缩, 这样可以直接提取该目录中的文件。可以将 dictionary.db 文件复制到 res aw 目录中

31. 如何将打开 res aw 目录中的数据库文件?

解答: 在 Android 中不能直接打开 res aw 目录中的数据库文件, 而需要在程序第一次启动时将该文件复制到手机内存或 SD 卡的某个目录中, 然后再打开该数据库文件。复制的基本方法是使用 getResources().openRawResource 方法获得 res aw 目录中资源的 InputStream 对象, 然后将该 InputStream 对象中的数据写入其他的目录中相应文件中。在 Android SDK 中可以使用 SQLiteDatabase.openOrCreateDatabase 方法来打开任意目录中的 SQLite 数据库文件。

32. Android 引入广播机制的用意?

答: a:从 MVC 的角度考虑(应用程序内)

其实回答这个问题的时候还可以这样问, android 为什么要有那 4 大组件, 现在的移动开发模型基本上也是照搬的 web 那一套 MVC 架构, 只不过是改了点嫁妆而已。android 的四大组件本质上就是为了实现移动或者说嵌入式设备上的 MVC 架构, 它们之间有时候是一种相互依存的关系, 有时候又是一种补充关系, 引入广播机制可以方便几大组件的信息和数据交互。

b: 程序间互通消息(例如在自己的应用程序内监听系统来电)

c: 效率上(参考 UDP 的广播协议在局域网的方便性)

d: 设计模式上(反转控制的一种应用, 类似监听者模式)

转自: <http://www.cnmsdn.com/html/201101/1295431222ID9251.html>

33、android 的优势与不足

Android 平台手机 5 大优势:

一、开放性

在优势方面, Android 平台首先就是其开发性, 开发的平台允许任何移动终端厂商加入到 Android 联盟中来。显著的开放性可以使其拥有更多的开发者, 随着用户和应用的日益丰富,

一个崭新的平台也将很快走向成熟

开发性对于 Android 的发展而言，有利于积累人气，这里的人气包括消费者和厂商，而对于消费者来讲，随大的受益正是丰富的软件资源。开放的平台也会带来更大竞争，如此一来，消费者将可以用更低的价位购得心仪的手机。

二、挣脱运营商的束缚

在过去很长的一段时间，特别是在欧美地区，手机应用往往受到运营商制约，使用什么功能接入什么网络，几乎都受到运营商的控制。从去年 iPhone 上市，用户可以更加方便地连接网络，运营商的制约减少。随着 EDGE、HSDPA 这些 2G 至 3G 移动网络的逐步过渡和提升，手机随意接入网络已不是运营商口中的笑谈，当你可以通过手机 IM 软件方便地进行即时聊天时，再回想不久前天价的彩信和图铃下载业务，是不是像噩梦一样？

互联网巨头 Google 推动的 Android 终端天生就有网络特色，将让用户离互联网更近。

三、丰富的硬件选择

这一点还是与 Android 平台的开放性相关，由于 Android 的开放性，众多的厂商会推出千奇百怪，功能特色各具的多种产品。功能上的差异和特色，却不会影响到数据同步、甚至软件的兼容，好比你从诺基亚 Symbian 风格手机 一下改用苹果 iPhone ，同时还可将 Symbian 中优秀的软件带到 iPhone 上使用、联系人等资料更是可以方便地转移，是不是非常方便呢？

四、不受任何限制的开发商

Android 平台提供给第三方开发商一个十分宽泛、自由的环境，不会受到各种条条框框的阻扰，可想而知，会有多少新颖别致的软件会诞生。但也有其两面性，血腥、暴力、情色方面的程序和游戏如可控制正是留给 Android 难题之一。

五、无缝结合的 Google 应用

如今叱咤互联网的 Google 已经走过 10 年度历史，从搜索巨人到全面的互联网渗透，Google 服务如地图、邮件、搜索等已经成为连接用户和互联网的重要纽带，而 Android 平台手机将无缝结合这些优秀的 Google 服务。

再说 Android 的 5 大不足：

一、安全和隐私

由于手机与互联网的紧密联系，个人隐私很难得到保守。除了上网过程中经意或不经意留下的个人足迹，Google 这个巨人也时时站在你的身后，洞穿一切，因此，互联网的深入将会带来新一轮的隐私危机。

二、首先开卖 Android 手机的不是最大运营商

众所周知，T-Mobile 在 23 日，于美国纽约发布了 Android 首款手机 G1。但是在北美市场，最大的两家运营商乃 AT&T 和 Verizon，而目前所知取得 Android 手机销售权的仅有 T-Mobile 和 Sprint，其中 T-Mobile 的 3G 网络相对于其他三家也要逊色不少，因此，用户可以买账购买 G1，能否体验到最佳的 3G 网络服务则要另当别论了！

三、运营商仍然能够影响到 Android 手机

在国内市场，不少用户对购得移动定制机不满，感觉所购的手机被人涂画了广告一般。这样的情况在国外市场同样出现。Android 手机的另一发售运营商 Sprint 就将在其机型中内置其手机商店程序。

四、同类机型用户减少

在不少手机论坛都会有针对某一型号的子论坛，对一款手机的使用心得交流，并分享软件资源。而对于 Android 平台手机，由于厂商丰富，产品类型多样，这样使用同一款机型的用户越来越少，缺少统一机型的程序强化。举个稍显不当的例子，现在山寨机泛滥，品种各异，就很少有专门针对某个型号山寨机的讨论和群组，除了哪些功能异常抢眼、颇受追捧的机型

以外。

五、过分依赖开发商缺少标准配置

在使用 PC 端的 Windows Xp 系统的时候，都会内置微软 Windows Media Player 这样一个浏览器程序，用户可以选择更多样的播放器，如 Realplay 或暴风影音等。但入手开始使用默认的程序同样可以应付多样的需要。在 Android 平台中，由于其开放性，软件更多依赖第三方厂商，比如 Android 系统的 SDK 中就没有内置音乐播放器，全部依赖第三方开发，缺少了产品的统一性。

34、android 中有哪几种解析 xml 的类？官方推荐哪种？以及它们的原理和区别。

XML 解析主要有三种方式，SAX、DOM、PULL。常规在 PC 上开发我们使用 Dom 相对轻松些，但一些性能敏感的数据库或手机上还是主要采用 SAX 方式，SAX 读取是单向的，优点：不占内存空间、解析属性方便，但缺点就是对于套嵌多个分支来说处理不是很方便。而 DOM 方式会把整个 XML 文件加载到内存中去，这里 Android 开发网提醒大家该方法在查找方面可以和 XPath 很好的结合如果数据量不是很大推荐使用，而 PULL 常常用在 J2ME 对于节点处理比较好，类似 SAX 方式，同样很节省内存，在 J2ME 中我们经常使用的 KXML 库来解析。

详细情况请参考 http://blog.csdn.net/Android_Tutor/archive/2010/09/17/5890835.aspx

<http://www.linuxidc.com/Linux/2010-11/29768.htm>

<http://littlefermat.blog.163.com/blog/static/59771167200981853037951/>

35、DDMS 和 TraceView 的区别？

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

36、Activity 被回收了怎么办？

只有另启用了

37、java 中如何引用本地语言

可以用 JNI 接口

38、谈谈 Android 的 IPC 机制

IPC 是内部进程通信的简称，是共享”命名管道”的资源。Android 中的 IPC 机制是为了让 Activity 和 Service 之间可以随时的进行交互，故在 Android 中该机制，只适用于 Activity 和 Service 之间的通信，类似于远程方法调用，类似于 C/S 模式的访问。通过定义 AIDL 接口文件来定义 IPC 接口。Servier 端实现 IPC 接口，Client 端调用 IPC 接口本地代理。

39、NDK 是什么

NDK 是一些列工具的集合，

NDK 提供了一系列的工具，帮助开发者迅速的的开发 C/C++ 的动态库，并能自动将 so 和 java 应用打成 apk 包。

NDK 集成了交叉编译器，并提供了相应的 mk 文件和隔离 cpu、平台等的差异，开发人员只需简单的修改 mk 文件就可以创建出 so

40 描述一下 android 的系统架构

android 系统架构分从下往上为 linux 内核层、运行库、应用程序框架层、和应用程序层

linuxkernel: 负责硬件的驱动程序、网络、电源、系统安全以及内存管理等功能。

libraries 和 androidruntime: libraries: 即 c/c++ 函数库部分，大多数都是开放源代码的函

数据库，例如 webkit，该函数库负责 android 网页浏览器的运行，例如标准的 c 函数库 libc、openssl、sqlite 等，当然也包括支持游戏开发 2dsgl 和 3dopengles，在多媒体方面有 mediaframework 框架来支持各种影音和图形文件的播放与显示，例如 mpeg4、h.264、mp3、aac、amr、jpg 和 png 等众多多媒体文件格式。android 的 runtime 负责解释和执行生成的 dalvik 格式的字节码。

applicationframework（应用软件架构），java 应用程序开发人员主要是使用该层封装好的 api 进行快速开发。

applications:该层是 java 的应用程序层，android 内置的 googlemaps、e-mail、即时通信工具、浏览器、mp3 播放器等处于该层，java 开发人员开发的程序也处于该层，而且和内置的应用程序具有平等的位置，可以调用内置的应用程序，也可以替换内置的应用程序。

上面的四个层次，下层为上层服务，上层需要下层的支持，调用下层的服务，这种严格分层的方式带来的极大的稳定性、灵活性和可扩展性，使得不同层的开发人员可以按照规范专心特定层的开发。

android 应用程序使用框架的 api 并在框架下运行，这就带来了程序开发的高度一致性，另一方面也告诉我们，要想写出优质高效的程序就必须对整个 applicationframework 进行非常深入的理解。精通 applicationframework，你就可以真正的理解 android 的设计和运行机制，也就更能够驾驭整个应用层的开发。

一些常见的 Android 面试基础题做下总结，看看你能做出多少道？

1. Intent 的几种有关 Activity 启动的方式有哪些，你了解每个含义吗？

这里 Android123 提示大家，Intent 的一些标记有 FLAG_ACTIVITY_BROUGHT_TO_FRONT、FLAG_ACTIVITY_CLEAR_TOP、FLAG_ACTIVITY_CLEAR_WHEN_TASK_RESET、FLAG_ACTIVITY_EXCLUDE_FROM_RECENTS、FLAG_ACTIVITY_MULTIPLE_TASK 和 FLAG_ACTIVITY_NEW_TASK 等。每种含义大家看 SDK 文档和具体跑下这样你的记忆会

2. Activity 和 Task 的启动模式有哪些？每种含义是什么？

有关在 AndroidManifest.xml 中的 android:launchMode 定义，主要有 standard、singleTop、singleTask 和 singleInstance，同时对于 android:taskAffinity 这些问题大家也要了解，Android 开发网在以前的文章中讲过，不过很多开发者仍然不是很清楚，这些基础问题我们以后仍然会再次总结。

关于 Activity 的启动模式，参考：
<http://blog.csdn.net/feng88724/archive/2011/05/11/6412638.aspx>

3. 通过 Intent 传递一些二进制数据的方法有哪些？

1). 使用 Serializable 接口实现序列化，这是 Java 常用的方法。

2). 实现 Parcelable 接口，这里 Android 的部分类比如 Bitmap 类就已经实现了，同时 Parcelable 在 Android AIDL 中交换数据也很常见的。

4. 能说下 Android 应用的入口点吗？

真正的 Android 入口点是 application 的 main，你可以看下 androidmanifest.xml 的包含关系就清楚了。可以没有 Activity 但是必须有 Application

5. Android 都有哪些 XML 解析器，都熟练掌握吗？

这里 XmlPull、SAX 和 DOM 相信做过 Web 开发的都已经滚瓜烂熟了。

6. SQLite 支持事务吗？添加删除如何提高性能？

SQLite 作为轻量级的数据库，比 MySQL 还小，但支持 SQL 语句查询，提高性能可以考虑通过原始经过优化的 SQL 查询语句方式处理。

7. Android Service 和 Binder、AIDL 你都熟练吗?

作为 Android 重要的后台服务，这些每个 Android 开发者都应该掌握，这也算是和 Java SE 最大的不同了，具体的实例大家可以查看 Android 音乐播放器的源代码 Music.git 中的，这里不再赘述。

8. 你用过哪款 Android 手机，有哪些优点和不足，相对于 iPhone 或 Symbian 又有哪些优缺点?

把这个作为面试题也是考察下，可以大概了解到它对 Android 的了解程度，多移动开发的认识。

【2011-5-12 补充】

1. 请描述下 Activity 的生命周期。

创建 onCreate - 启动 onStart - 恢复 onResume - 暂停 onPause - 结束 onEnd - 销毁 onDestroy

2. 如果后台的 Activity 由于某原因被系统回收了，如何在被系统回收之前保存当前状态?

在” 暂停 onPause” 状态将数据保存。

3. 如何将一个 Activity 设置成窗口的样式。

设置 Theme。

4. 如何退出 Activity? 如何安全退出已调用多个 Activity 的 Application?

5. 请介绍下 Android 中常用的五种布局。

线性布局 LinearLayout, 相对布局 RelativeLayout

表单布局 TableLayout,
绝对布局 AbsLayout (已淘汰)

帧布局 FrameLayout

6. 请介绍下 Android 的数据存储方式。

Preference,文件,数据库 SQLite,网络存储

7. 请介绍下 ContentProvider 是如何实现数据共享的。

8. 如何启用 Service，如何停用 Service。

启动：

```
Context.startService()  
and  
Context.bindService().
```

关闭： Context.stopService().

```
Service.stopSelf()  
or  
Service.stopSelfResult()
```

9. 注册广播有几种方式，这些方式有何优缺点？请谈谈 Android 引入广播机制的用意。

10. 请解释下在单线程模型中 Message、Handler、Message Queue、Looper 之间的关系。

11. AIDL 的全称是什么？如何工作？能处理哪些类型的数据？

12. 请解释下 Android 程序运行时权限与文件系统权限的区别。(Edited by Sodino)

13. 系统上安装了多种浏览器，能否指定某浏览器访问指定页面？请说明原由。

14. 有一个一维整型数组 `int[]data` 保存的是一张宽为 `width`，高为 `height` 的图片像素值信息。请写一个算法，将该图片所有的白色不透明(`0xffffffff`)像素点的透明度调整为 50%。

15. 你如何评价 Android 系统？优缺点。（这一题重复了）

优势：

一、开放性 三、丰富的硬件选择 五、无缝结合的 Google 应用支持

缺点： 一、安全和隐私

二，开发商自定义，不够统一 四、同类机型用户减少

【2011-5-12 补充】

1、什么是 ANR 如何避免它？

<http://blog.csdn.net/Zengyangtech/archive/2010/11/21/6025671.aspx>

2、什么情况会导致 Force Close ？如何避免？能否捕获导致其的异常？

3、Android 本身的 api 并未声明会抛出异常，则其在运行时有可能抛出 runtime 异常，你遇到过吗？诸有的话会导致什么问题？如何解决？

会有运行时异常，运行时异常无需捕捉

4、简要解释一下 activity、 intent 、 intent filter、 service、 Broadcast、 BroadcastReceiver

这个问题问的很普遍，自己多看看吧。在此不多说了！

5、IntentService 有何优点？

【2011-5-12 补充】

1. android:paddingLeft 与 android:layout_marginLeft 的区别

当按钮分别设置以上两个属性时，得到的效果是不一样的。

android:paddingLeft="30px"

按钮上设置的内容（例如图片）离按钮左边边界 30 个像素

android:layout_marginLeft="30px"

整个按钮离左边设置的内容 30 个像素

2. Android 动画有哪几种？描述一下

两种。 Tween 动画和 Frame 动画。 Tween 动画主要是透明度、尺寸伸缩、旋转、位移等效果。

Frame 动画可以理解成 gif， 一帧一帧的显示图片。比较常用的有滚动条效果。

3. 对 Intent、Activity、广播、Service 等的理解。

4. 哪些情况下会发生 ANR？怎么对应？（这个问题也重复了）

Application Not Response。

5. 隐式、显式 Intent 的区别

- 显式意图

通过名字指明目标组件（这个组件名字字段 component name field, 前面提到过,

有一个数值集）。既然组件名称通常不为其他应用程序的开发者所了解，显式意图典型的被用作应用程序的内部消息-例如一个活动启动一个附属服务或姊妹活动。

隐式意图

不命名目标组件（组件名称字段为空）。隐式意图经常用来激活其他应用程序的组件。

这一块参考 Intent 说明，可以同时看一下 Notepad 的例子，帮助理解，个人感觉很重要！

(上面几个问题感觉都不错，各位自己查漏补缺吧！有些还没有写答案的，各位有空可以回答一下~ 我会将你的答案填充上去)

2011-07-05 补充

返回键与 Home 键区别？

back 键默认行为是 finish 处于前台的 Activity 的即 Activity 的状态为 Destroy 状态为止，再次启动该 Activity 是从 onCreate 开始的(不会调用 onSaveInstanceState 方法)。Home 键默认是 stop 前台的 Activity 即状态为 onStop 为止而不是 Destroy,若再次启动它，会调用 onSaveInstanceState 方法，保持上次 Activity 的状态则是从 OnRestart 开始的---->onStart()--->onResume()。

1、Android 操作系统是 Google 在什么时间公布的

A、2007 年 11 月 5 日 B、2010 年 5 月 20 日晚 10 点 30 分 C、2009 年 11 月 10 日

正确答案：A

2、Android 基于什么平台的开源智能手机操作系统名称

A、WinCE B、Linux C、SHP

正确答案：B

3、Linux 平台由以下下哪些组成？

A、操作系统 B、中间件 C、用户界面 D、应用软件

正确答案：ABCD

4、Android 操作系统主要的特点是：（ ）

A、与 Google 无缝结合 B、实现个性化应用
C、良好的平台开放性 D、软件均要收费

正确答案：ABC

5、Android 操作系统向开发人员提供是

A、全部免费 B、部分收费 C、只有 10%左右是收费的

正确答案：A

6、Android 个性化界面设置的添加方式

A、点选屏幕下方添加键进行添加 B、长按屏幕空白处进行添加 C、直接拖拽至屏幕进行添加

正确答案：ABC

7、个性化界面设置不同内容包括：

A、快捷方式 B、小插件 C、窗口小部件 D、文件夹

正确答案：ABCD

8、Android 操作系统采用全新解锁方式描述正确的是：（ ）

A、自定义图型解锁； B、滑动解锁； C、规定图型解锁； D、只可侧键解锁；

正确答案：AB

9、Android 支持谷歌全方位移动服务，丰富最新网络体验包括：

A、电子市场； B、Google Talk； C、Google 搜索；
D、G-mail 邮箱； E、Google 地图

正确答案：ABCDE

10、截止 2010 年 12 月，中国上市的 Android 操作系统已经发布了以下哪几个版本

A、1.0 B、1.3 C、1.5 D、1.6 E、2.0 F、2.1 G、2.2

正确答案：ACDEFG

11、在中国，以下哪些属于 Android 软件主要下载门户？（ ）

A、 安卓版 B、 机锋网 C、 百度 D、 Google E、 优亿市场

正确答案: ABE

12、 Android 安装软件后缀是: ()

A. .sis B. cab C. .apk D. .jar

正确答案: C

13、 以下哪些不是 Android 操作系统的版本: ()

A、 1.0 B、 1.3 C、 2.0 D、 2.1 E、 2.2

正确答案: B

14、 Android 操作平台的开放性也针对了众多手机品版的厂家进行了完全的开放, 以下有哪几个品牌的手机从未使用过 Android 操作系统: ()

A. HTC B. I phone C. samsung D. 诺基亚 E. 黑莓

正确答案: BDE

15、 Android1.5 操作系统叫 ()

A. 纸杯蛋糕 B. 甜甜圈 C. 法式奶油夹心甜点 D. 果馅饼 E. 冻酸奶

正确答案: A

16、 Android1.6 操作系统叫 ()

A. 纸杯蛋糕 B. 甜甜圈 C. 法式奶油夹心甜点 D. 果馅饼 E. 冻酸奶

正确答案: B

17、 Android2.0 操作系统叫 ()

A. 纸杯蛋糕 B. 甜甜圈 C. 法式奶油夹心甜点 D. 果馅饼 E. 冻酸奶

正确答案: C

18、 Android2.1 操作系统叫 ()

A. 纸杯蛋糕 B. 甜甜圈 C. 法式奶油夹心甜点 D. 果馅饼 E. 冻酸奶

正确答案: D

19、 Android2.2 操作系统叫 ()

A. 纸杯蛋糕 B. 甜甜圈 C. 法式奶油夹心甜点 D. 果馅饼 E. 冻酸奶

正确答案: E

20、 Android2.2 操作系统与 Android2.1 操作系统对比有哪些方面的提升 ()

A. 整体性能大幅度的提升 B. 3G 网络共享功能 C. 全面的 Flash 支持
D. App 2 sd 功能(安装程序到 SD 卡) E. 全新的软件商店
F. 更好的网络体验

正确答案: ABCDEF

21、 Android 系统可以同时运行多个程序吗? ()

A. 可以; B. 不可以

正确答案: A

22、 以下对 Android 操作系统描述正确的是: ()

A. 系统资源占用少, 系统运行速度快
B. 完全开放性的操作系统, 对所有软件开发商、手机厂商、运营商平等享有
C. 有众多的软件提供给消费者下载
D. 个性化的操作界面, 操作简单

正确答案: ABCD

23、 Android 操作系统的手机可以有以下几个方法进行软件安装 ()

A. 通过手机直接登陆百度网站下载安装;
B. 通过手机直接登陆 Android 门户网站进行下载安装;

- C. 通过数据线与电脑连接直接下载安装;
- D. 通过 PC 终端上网下载至 SD 卡里再插入手机进行安装;

正确答案: ABCD

24、Android 操作系统具有个性化操作界面设置, 以下关于进行添加描述正确的是()

- A. 点选屏幕下方添加键进行添加
- B. 长按屏幕空白处进行添加
- C. 直接拖拽至屏幕进行添加
- D. 不可直接拖拽至屏幕进行添加

正确答案: ABC

26、Android 从哪个版本开始支持动态壁纸 ()

- A. 1.0
- B. 1.3
- C. 2.0
- D. 2.1
- E. 2.2

正确答案: D

27、Android 从哪个版本开始支持 3D 动态相册 ()

- A. 1.0
- B. 1.3
- C. 2.0
- D. 2.1
- E. 2.2

正确答案: D

28、Android 从哪个版本开始系统自带支持 mobileAP 热点 ()

- A. 1.0
- B. 1.3
- C. 2.0
- D. 2.1
- E. 2.2

正确答案: E

29、Android 从哪个版本开始支持 Flash10.1 ()

- A. 1.0
- B. 1.3
- C. 2.0
- D. 2.1
- E. 2.2

正确答案: E

30、Android 从哪个版本开始系统自带支持 AllShare 协议 ()

- A. 1.0
- B. 1.3
- C. 2.0
- D. 2.1
- E. 2.2

正确答案: E

31、Android 从哪个版本开始可以提供多个不同账户统一收件箱

- A. 1.0
- B. 1.3
- C. 2.0
- D. 2.1
- E. 2.2

正确答案: C

32、Android 从哪个版本开始浏览器支持双击缩放

- A. 1.0
- B. 1.3
- C. 2.0
- D. 2.1
- E. 2.2

正确答案: C

33、Android 从哪个版本开始强化了语音识别的搜索功能

- A. 1.0
- B. 1.3
- C. 2.0
- D. 2.1
- E. 2.2

正确答案: C

34、Android 从哪个版本开始加入微软 Exchange 服务支持

- A. 1.0
- B. 1.3
- C. 2.0
- D. 2.1
- E. 2.2

正确答案: C

35、Android 从哪个版本开始支持双击上传图片至 YouTube

- A. 1.0
- B. 1.3
- C. 2.0
- D. 2.1
- E. 2.2

正确答案: C

36、Android 从哪个版本开始支持谷歌地图服务更新

- A. 1.0
- B. 1.3
- C. 2.0
- D. 2.1
- E. 2.2

正确答案: C

37、Android 从哪个版本开始支持所有文本输入区应用语音识别

- A. 1.0
- B. 1.3
- C. 2.0
- D. 2.1
- E. 2.2

正确答案: D

38、Android 从哪个版本开始支持 3D 相片集

- A. 1.0 B. 1.3 C. 2.0 D. 2.1 E . 2.2

正确答案: D

39、Android 从哪个版本开始支持修复 FM 收音机

- A. 1.0 B. 1.3 C. 2.0 D. 2.1 E . 2.2

正确答案: D

40、 以下哪些是 Android 的优势 ()

- A、 开源的操作系统，代码完全开放，开放成本最低
- B、 便捷的用户体验界面，操作简单
- C、 平台技术先进，更好的支持互联网业务
- D、 系统资源占用少，系统速度快

正确答案: ABCD

41、Android 从哪个版本开始软件可安装到存储卡上

- A. 1.0 B. 1.3 C. 2.0 D. 2.1 E . 2.2

正确答案: E

42、Android 从哪个版本开始自动更新设备上已经安装的应用程序

- A. 1.0 B. 1.3 C. 2.0 D. 2.1 E . 2.2

正确答案: E

43、Android 能实现手机的主屏幕的自定义，并将常用的各种应用以图标的形式添加至主屏幕界面

- A、对
- B、错

正确答案: A

44、Android 能实现手机的主屏幕的自定义，但是添加的图标要按照功能属性排列

- A、对
- B、错

正确答案: B

45、Android 操作系统的主屏界面图标设置一次后，就不能更改

- A、对
- B、错

正确答案: B

46、Android 操作系统可以将哪些类型不同内容，随意添加至主屏幕

- A、快捷键
- B、窗口小部件
- C、文件夹
- D、壁纸

正确答案: ABCD