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160，160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  
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：  
  
l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。  
  
l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。  
  
l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acitvity的肩上，这句话也就暗含了不要在acitivity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。  
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");  
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  
  
< 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"  
/>  
  
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，这样就多么松藕合啊，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，第三方程序无法调用。  
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，通过它可以让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，onDestory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：  
  
  
  
  
  
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recevier ,名为mEvtReceiver，然后同样用上面的方法以重写onReceiver，  
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/%C1%F7%C0˲%BB%D0%DD/blog/item/4e576b7c58d742360cd7dac9.html  
  
http://blog.csdn.net/xuxinyl/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adil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Therad.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，从而避免因为输入事件超过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，可以看起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、Broadcase、BroadcaseReceiver  
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  
\* Acitivity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  
如今叱诧互联网的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等众多的多媒体文件格式。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。