

3.一个有N个元素的一维数组（A[0].A[1], ..., A[n-1]），设计一个算法求解该数组最大子数组。（要求时间复杂度是O(n)）

4.用Java代码实现判断一个字符串中是否包含另一个字符串，不要用jdk中String的indexOf方法。

5.用Java实现一个Singleton.

6.简述Observer设计模式，并画出Observer模式的典型结构类图

7.& 和&&的区别

8.什么是dpi,ps,sp?适配是如何做的，为什么？

9.用最有效率的方法算出2乘以8等于几？

10.静态成员类、非静态成员类有什么区别？什么是匿名内部类？

11.Handler机制的原理，内部是如何实现的，消息队列的实现机制？

12.什么是ANR，如何定位和避免？

13.APK之间互相传递数据

14.简述Android的启动过程

15.如何加载ndk库？如何在jni中注册native函数，有几种注册方法？

16.如何收集软件崩溃信息？

17.简述软件编译流程和发布流程

18.是否编译过android源码和linux内核源码

19.写一个归并排序数组

20.自定义控件相关

21.缓存实现

22.线程同步(生产者 消费者)

23.符号匹配(栈)

24.还问了下拉刷新和分页加载的实现机制

盘古搜索面试总结

2.求一个数组的最大连续子序列和，比如：-10,3,5,2，-3,6,7

最笨的方法就是穷举法，用三层for循环

3.求一个数组中出现次数最多且值最大的数

android的基础题

1.service的生命周期

2.Intent用途，了解多少？IntentService知道么？

3.自定义控件

4.xmpp协议的理解

5.广播创建方式有几种?有什么区别？

6.多线程有几种实现方式？

继承Thread, 实现Runnable接口， AsyncTask

一、listview如何优化（这个很重要，多次被问到）

1.分页加载

2.convertView缓存

3.viewHolder

4.scrollListener，记住滑动的状态

二、常用排序算法，各种算法的复杂度，用Java实现一种排序算法。

四、单链表的反转、合并、排序

五、二分查找法

六、Intent Flag常用的记住几个

七、AIDL怎么用？全称是什么？

八、taskAffinity 亲和性

这个一般人应该都不熟悉，了解一下，面试中可能被问到，几率不大。

百度各种面试总结

1.Listview item缓存如何实现

convertView

2.很多view如何缓存

3.图片上传

4.图片过大如何处理？压缩后图片有多大？
分辨率压缩、质量压缩

5.OOM怎么办？

6.图片的缓存机制，如何做到高效和稳定？

7.自定义view怎么做能提高效率? 自定义view如何做图片缓存？自定义view如何判断单击和双击？自定义view 的 onDraw方法主要做什么？

避免重复调用measure 和 layout

8.是否看过view的源码

9.动画的机制及其底层实现？

10.二叉树、链表

11.平时看什么书？去哪些论坛看？

12.问项目底层实现

13.Linux文件系统实现，如何实现文件的隐藏？

面试算法题总结

1.有一个整数数组，有负数和整数，用一个方法把正负数分开，要求空间复杂度是O(1),时间复杂度是：O(N)（百度移动平台部）

2.有一个数列，奇数个，其中有3个是单个的， 其它都是成对的，怎么找出其中任意一个单个的数（小米一面）

3.在一个数组中，找出出现次数最多且数值最大的一个数并输出

4.用最快的方式将两个数组合并到一个数组中？时间复杂度

=====

2013-4-10补充：

问题：怎样的类可以作为HashMap的键？

答：这个问题是搜狗2012年的笔试题，我开始还真是没想太明白。刚才看到HashMap的时候才发现原来是这么回事。

当需要往ArrayList，HashMap中放东西时，像int，double这种内建类型是放不进去的，因为容器都是装object的，这就需要这些内建类型的外覆类了。比如：int对应的Integer，double对应的Double。

同样是搜狗的笔试题：适合频繁插入和查找的数据结构

答：频繁插入，链表效率高，频繁查找数组效率高，那么综合一下，我个人觉得是哈希表，或者说HashMap等使用hash算法的数据结构。

- 1、对于简历上所写的每个问题，面试官感兴趣的会对我提问，并分析里边遇到的问题以及得到的经验。面试过程描述以前做过的项目。项目中你认为的难点是什么。
- 2、对于网络交互这块儿需要注意什么问题
- 3、android是一个什么样的系统。Jni调用机制是什么。用uml画一个在你的项目中使用的设计模式。

面试中常问的非技术类问题：

- 1.简单介绍一下你自己。
- 2.对自己做个评价，优点、缺点。
- 3.自己做过的项目，详细介绍一下。
- 4.实习经历，在实习过程中遇到哪些问题，如何解决的？
- 5.你对公司有什么问题要问吗（对公司有什么要了解的）？

技术类的问题主要就看对技术的掌握程度和自己的综合能力了。

百度移动终端研发工程师笔试题，题量不大，2个小时，有点难度。

- 1.写一个字符串逆序的程序，时间复杂度和空间复杂度最低，效率越高越好。
- 2.实质：二叉树的层序遍历，每层的结点用单链表输出。
- 3.1-N（N最大32000，且未知），内存只有4K，找出其中的重复数
- 4.编程题，比较复杂，是个系统设计题，与《编程之美》中3.2，电话号码对应英语单词类似。

百度软件开发工程师一面问题：

- 1.有101个数，为[1，100]之间的数，其中一个数是重复的，如何寻找这个重复的数，其时间复杂度和空间复杂度是多少？
- 2.Java中抽象类与接口的区别。
- 3.进程与线程之间的联系与区别。（多家公司都在问，好好研究一下）
- 4.谈谈对设计模式的认识与理解，简单介绍一下你所知道的设计模式。（多家公司都问，Android方向面试必考的）
- 5.线程、多线程相关（必问）
- 6.Linux常用的命令，shell编程，grep命令的使用。
- 7.海量数据查找或者排序，有资源限制要求。（常考的）

建议：简历中对自己的专业技能要实事求是的写，突出自己的重点，不宜托大，面试官面试时提问的依据就是简历上的内容。百度的工作环境很好，做技术的员工给人的感觉就是虽然人家的技术水平很高，但是都比较谦逊。百度确实是一个不错的互联网公司。

Sina无线客户端开发工程师面试

笔试的题目和7月22日实习生的题目差不多。面试主要问了：

- 1.二叉树的一些基础知识。
- 2.Java多线程，对stop()方法的认识，如何安全的终止一个线程。
- 3.设计模式相关的问题
- 4.谈谈你对面向对象编程中继承的认识
- 5.排序算法

网易2012校招无线客户端开发笔试，题目不太难，比较基础。

- 1.面向对象编程中多态的概念，用伪代码展示出来。
- 2.斐波那契数列编程，效率越高越好（详见《编程之美》2.9）
- 3.60W长度为32的字符串一次性读入内存中，大约占用多少内存空间？
- 4.一个X向量n，如何求其内部的最大距离。
- 5.在浏览器中输入<http://www.163.com>，从输入到页面在浏览器中显示出来，期间发生了哪些过程，越详细越好。
- 6.简单介绍一下Android系统中的Activity,Service, Intent, ContentProvider组件。
- 7.编程解析一段json格式内容，并在界面上显示出来。
- 8.英文阅读，考察英语能力。
- 9.网络应用分析，发挥自己的主观能动性和积累。

搜狗2012校招笔试

搜狗的题目出的很全，所有职位的题目都在一份卷子上，涉及到数据结构、网络、操作系统、算法、软件工程等多方面，根据自己所投递的职位，有针对性的选择要答的题目即可。

金山2012校园招聘笔试

为了保险，做了运维的题目，难度不大，主要是网络和Linux操作系统方面的，稍微准备一下即可。答题的时候自我感觉良好，结果没收到面试通知，被鄙视了，o(∩_∩)o

[Android开发工程师面试中经常会问到的问题](#)

- 1.Activity的生命周期（最基本的）。
- 2.Android基本框架结构。
- 3.Android终止一个进程的方法，如何彻底终止一个程序。
- 4.如何改写程序，改变android手机中返回键的功能（西安TCL研究院技术面试）。

一般来说，大公司的校园招聘中首先看的还是基础，有工作或者实习经验的会重点考虑，多问一些在实习中的项目细节，同时会问一些算法，建议认真研读《编程之美》《编程珠玑》之类的书籍。首先要做好充足的准备，打好基础，其次还要看一些算法、设计的内容，一面以后的技术面问的题目都有一定的挑战性。如果有实习 或者项目经历，可以大大增加校招中简历筛选通过的机率。对于互联网公司，特别喜欢海量数据的排序、搜索，网上类似的资源很多，可以看看。技术更新永无止 境，要时刻保持清醒的头脑，不断学习，充实自己，才能在竞争

中不被淘汰，祝大家好运，找到心仪的工作。

=====

=====

1.程序之间亲和性的理解
eoe的一个朋友的回答，要不我还真想不到是什么。
是问affinity吧
sdk文档中有详细说明
看这篇文章：http://blog.csdn.net/song_shi_chao/article/details/8754300

2.Task任务栈的分配
<http://blog.csdn.net/guomeijuan916/article/details/8121468>

3.Activity如何启动一个service
这个问题开始真是小看了，个人觉得有两种：一种就是start一个service，另外一种就是bind一个service。

百度android
笔试题
java如何同步
wait 和 sleep的区别
java常用的数据结构 如何实现
重载与重写区别
android 如何提高稳定性

android 有哪些安全机制

1) http://www.360doc.com/content/11/12/19/23/3700464_173515364.shtml
2) <http://www.cnblogs.com/GnagWang/archive/2011/03/21/1990507.html>

基 于上面两篇文章，我也总结下：1.Android基于Linux，Linux的安全机制适用于Android，通过用户id，组id等实 现；2.Android的权限机制：在manifest.xml中会配置相应的权限 3.Android的独立进程机制：每个应用都运行在独立的进程中，互相不干扰，当然他们共享资源和内存等；还有注意shareId如果相同会运行在一个 进程中 4.签名机制：Android打包的时候有自己独立的keystore文件，所以这个文件是唯一的，保证应用的安全性

android 应用之间有几种共享数据的方法
android 哪些操作会导致内存泄露 如何排查
.9.png 和 png 的区别
sqlite 使用注意事项
Listview 中 getview中 convertview 的工作原理
service 有什么特征，在哪种情况下会用到service

面试
面试的时候会问你关于简历上的项目，会问得比较细，问你几个人做的，做了多长时间，画出原理图
intent 如何开启activity的？
manifest 的作用
内存过低时，是用什么策略杀死进程@？

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更深刻些。

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。

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？
在Google的虚拟机中Application实例中已经实现了main方法，有关具体的Application介绍可以参考 Android程序的入口点是什么,不是Main()
吗：<http://www.android123.com.cn/kaifafaq/829.html>

5. Java基础问题，说说HashSet和HashTable的区别？
这个相信Java程序员都知道HashSet相对于HashMap就是不能存放重复的数据，对于HashTable来说，存放的数据不能出现key或value为null这样的情况。

6. Java基础问题，用Java的集合类写下广度优先搜索。
这个属于基础数据结构问题，通过队列处理一层一层，相对于深度优先的栈处理方式来说更实用一些。

7. 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。

8. 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。

9. 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。

百度Android开发面试题

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重绘和内存泄露的好像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invalidate或者postinvalidate.

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.

这篇文章会涉及到以下内容

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

```
1
public class MyActivity extends Activity
2
{
3
protected void onCreate(Bundle savedInstanceState);
4
protected void onStart();
5
protected void onResume();
6
protected void onPause();
7
protected void onStop();
8
protected void onDestroy();
```

9

}

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。

二 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。

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

```
1
public void onSaveInstanceState(Bundle outState) {
2
    super.onSaveInstanceState(outState);
3
    outState.putLong("id", 1234567890);
4
}
5
public void onSaveInstanceState(Bundle outState) {
6
}
}
```

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
}
4
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，那就不需要特殊记住什么，哦，没准你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：

```
1
Intent intent = new Intent();
2
intent.setAction(Intent.ACTION_CALL);
3
intent.setData(Uri.parse("tel:"+ number));
4
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
startActivity(intent);
4
Intent intent = new Intent(this,MyActivity.class);intent.getExtras().putString("id","1");
5
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AndroidMenifest中的intentfilter的配置了

Xml代码

```
01
<intent-filter>
02
<action android:name="android.intent.action.VIEW" />
03
<action android:value="android.intent.action.EDIT" />
04
<action android:value="android.intent.action.PICK" />
05
<category android:name="android.intent.category.DEFAULT"/>
06
<data android:mimeType="vnd.android.cursor.dir/vnd.google.note" />
07
</intent-filter>
08
<intent-filter>
09
<action android:name="android.intent.action.VIEW"/>
10
<action android:value="android.intent.action.EDIT" />
11
<action android:value="android.intent.action.PICK" />
12
<category android:name="android.intent.category.DEFAULT" />
13
<data android:mimeType="vnd.android.cursor.dir/vnd.google.note" />
14
</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：)

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【Technicolor的面试官问的这个】）

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？

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

以下面试题都是在网上找的总结出来的，谢谢大家的分享！希望，我们共同进步，找到自己梦想的公司：

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：

答：dvm是dalivk虚拟机。每一个android应用程序都在自己的进程中运行，都拥有一个dalivk虚拟机实例。而每一个dvm都是在linux的一个进程。所以说可以认为是同一个概念。

2.android的动画有哪几种？他们的特点和区别是什么？

答：两种，一种是tween动画，一种是frame动画。tween动画，这种实现方式可以使视图组件移动，放大或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。

3.handler进制的原理：

答：android提供了handler和looper来满足线程间的通信。Handler先进先出原则。looper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:你可以构造一个handler对象来与looper沟通，以便push新消息到messagequeue里；或者接收looper（从messagequeue里取出）所送来的消息。

3)messagequeue:用来存放线程放入的消息。

4)线程：UI thread 通常就是main thread,而android启动程序时会为它建立一个message queue.

4.android view的刷新：

答：Android中对View的更新有很多多种方式，使用时要区分不同的应用场合。我感觉最要紧的是分清：多线程和双缓冲的使用情况。

1).不使用多线程和双缓冲

这种情况最简单了，一般只是希望在View发生改变时对UI进行重绘。你只需在Activity中显式地调用View对象中的invalidate()方法即可。系统会自动调用 View的onDraw()方法。

2).使用多线程和不使用双缓冲

这种情况需要开启新的线程，新开的线程就不好访问View对象了。强行访问的话会 报：

android.view.ViewRoot\$CalledFromWrongThreadException：Only the originalthread that created a view hierarchy can touch its views.

这时候你需要创建一个继承了android.os.Handler的子类，并重写handleMessage(Messagemsg)方法。 android.os.Handler是能发送和处理消息的，你需要在Activity中发出更新UI的消息，然后再你的Handler（可以使用匿名内 部类）中处理消息（因为匿名内部类可以访问父类变量，你可以直接调用View对象中的invalidate()方法 ）。也就是说：在新线程创建并发送一个Message，然后再主线程中捕获、处理该消息。

3).使用多线程和双缓冲

Android中SurfaceView是View的子类，她同时也实现了双缓冲。你可以定义一个她的子类并实现 SurfaceHolder.Callback接口。由于实现SurfaceHolder.Callback接口，新线程就不需要 android.os.Handler帮忙了。SurfaceHolder中lockCanvas()方法可以锁定画布，绘制玩新的图像后调用 unlockCanvasAndPost(canvas)解锁（显示），还是比较方便得。

5.说说mvc模式的原理，它在android中的运用:

答：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。

2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。

3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acitvity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acitvity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。

6.Activity的生命周期:

答：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
- 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

7.让Activity变成一个窗口：

答：Activity属性设定：有时候会做个应用程序是漂浮在手机主界面的。这个只需要在设置下Activity的主题theme,即在Manifest.xml定义

Activity的地方加一句：

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

如果是作半透明的效果：

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

8.Android中常用的五种布局:

答：LinearLayout线性布局；AbsoluteLayout绝对布局；TableLayout表格布局；RelativeLayout相对布局；FrameLayout帧布局；

9.Android的五种数据存储方式：

答：sharedPreferences；文件；SQLite；contentProvider；网络

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

答：Handler获取当前线程中的looper对象，looper用来从存有Message的Message Queue里取出message，再由Handler进行message的分发和处理。

11.AIDL的全称是什么?如何工作?能处理哪些类型的数据？

答：AIDL(AndroidInterface Definition Language)android接口描述语言

12.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：

答：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。代码如下：

```
Intent intent = new Intent();
Intent.setAction(“android.intent.action.View”);
Uri uriBrowsers = Uri.parse(“http://www.sina.com.cn”);
Intent.setData(uriBrowsers);
//包名、要打开的activity
intent.setClassName(“com.android.browser”,“com.android.browser.BrowserActivity”);
startActivity(intent);
```

13.什么是ANR,如何避免？

答：ANR的定义：

在android上，如果你的应用程序有一段时间响应不移灵敏，系统会向用户提示“应用程序无响应”（ANR：application Not Responding）对话框。因此，在程序里对响应性能的设计很重要，这样，系统不会显示ANR给用户。

如何避免：

首先来研究下为什么它会在android的应用程序里发生和如何最佳构建应用程序来避免ANR.

android应用程序通常是运行在一个单独的线程（例如：main）里，这就意味你的应用程序所做的事情如果在主线程里占用了大长时间的话，就会引发ANR对话框，因为你的应用程序并没有给自己机会来处理输入事件或者Intent广播。

因此，运行在主线程里的任何访求都尽可能少做事情。特别是，activity应该在它的关键生命周期方法（onCreate()和onResume()）里尽可能少的去作创建操作。潜在的耗时操作，例如网络或数据库操作，或者高耗时的计算如改变位图尺寸，应该在子线程里（或者以数据库操作为例，通过异步请求的方式）来完成。然而，不是说你的主线程阻塞在那里等待子线程的完成---也不是调用Thread.wait()或者Thread.sleep()。替代的方法是：主线程应该为子线程提供一个Handler,以便完成时能够提交给主线程。以这种方式设计你的应用程序，将能保证你的主线程保持对输入的响应性并能避免由5秒输入事件的超时引发的ANR对话框。这种做法应该在其它显示UI的线程里效仿，因为它们都受相同的超时影响。

IntentReceiver执行时间的特殊限制意味着它应该做：在后台里做小的、琐碎的工作，如保存设定或注册一个Notification。和在主线 程里调用的其它方法一样，应用程序应该避免在BroadcastReceiver里做耗时的操作或计算，但也不是在子线程里做这些任务（因为BroadcastReceiver的生命周期短），替代的是，如果响应Intent广播需要执行一个耗时的动作的话，应用程序应该启动一个 Service。顺便提及一句，你也应该避免在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。如果你的应用程序在响应Intent广播时需要向用户展示什么，你应该使用Notification Manager来实现。

一般来说，在应用程序里，100到200ms是用户能感知阻滞的时间阈值，下面总结了一些技巧来避免ANR,并有助于让你的应用程序看起来有响应性。

如果你的应用程序为响应用户输入正在后台工作的话，可以显示工作的进度（ProgressBar和ProgressDialog对这种情况来说很有用）。特别是游戏，在子线程里做移动的计算。如果你的程序有一个耗时的初始化过程的话，考虑可以显示一个Splash Screen或者快速显示主画面并异步来填充这些信息。在这两种情况下，你都应该显示正在进行的进度，以免用户认为程序被冻结了。

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：如空指针等可以导致ForceClose;可以看Logcat，然后找到对应的程序代码来解决错误。

15.横竖屏切换时候的activity的生命周期：

答：

- 1) 新建一个activity,并把各个生命周期打印出来
- 2) 运行activity，得到如下信息：

onCreate()à

onStart()à

onResume()à

- 3) 按ctrl+F12切换成横屏时
- onSaveInstanceState()à
- onPause()à
- onStop()à

```
onDestroy()à
onCreate()à
onStart()à
onRestoreInstanceState()à
onResume()à
4) 再按ctrl+f12切换成竖屏时，发现打印了两次相同的Log
onSaveInstanceState()à
onPause()à
onStop()à
onDestroyà
onCreate()à
onStart()à
onRestoreInstanceState()à
onResume()à
```

```
onSaveInstanceState()à
onPause()à
onStop()à
onDestroyà
onCreate()à
onStart()à
onRestoreInstanceState()à
onResume()à
```

5) 修改AndroidManifest.xml，把该Activity添加android:configChanges=“orientation”,执行步骤3

```
onSaveInstanceState()à
onPause()à
onStop()à
onDestroy()à
onCreate()à
onStart()à
onRestoreInstanceState()à
onResume()à
```

6) 修改AndroidManifest.xml，把该Activity添加android:configChanges=“orientation”,执行步骤4,发现不会再打印相同信息，但多打印了一

行onConfigChanged

```
onSaveInstanceState()à
onPause()à
onStop()à
onDestroy()à
onCreate()à
onStart()à
onRestoreInstanceState()à
onResume()à
onConfigurationChanged()à
```

7) 把步骤5的android:configChanges=“orientation”改成

android:configChanges=“orientation|keyboradHidden”,执行步骤3，就只打印onConfigChanged

```
onConfigurationChanged()à
```

8) 把步骤5的android:configChanges=“orientation”改成

android:configChanges=“orientation|keyboradHidden”,执行步骤4

```
onConfigurationChanged()à
onConfigurationChanged()à
```

总结：

- 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, 切屏会重新调用各个生命周期，切横屏、竖屏时都只会执行一次，但是竖屏最后多打印一条onConfigurationChanged()
- 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(),横屏一次，竖屏两次

再总结下整个activity的生命周期：

- 1) 当前activity产生事件弹出Toast和AlertDialog的时候Activity的生命周期不会有改变
- 2) Activity运行时按下HOME键（跟被完全覆盖一样的）

```
onSavaInstanceStateà
onPauseà
onStopà
```

```
onRestartà
onStartà
onResumeà
```

- 3) 未被完全覆盖，只是失去焦点：
- ```
onPauseà
onResumeà
```

16.如何将SQLite数据库(.db文件)与apk文件一起发布？

答：可以将.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.db文件复制到res aw目录中

17.如何将打开res aw目录中的数据库文件？

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。

18.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：

答：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。

19.DDMS和TraceView的区别？

答：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

20.谈谈Android的IPC机制：

答：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和服务之间可以随时的进行交互，故在Android中该机制，只适用于Activity和服务之间的通信，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。

21.NDK是什么：

答：NDK是一系列工具的集合

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

22.描述一下android的系统架构：

答：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。

Linux内核层：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。

运行库和androidruntion：运行库：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的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，如mpeg4、h.264、mp3、aac、amr、jpg和png等众多 的多媒体文件格式。Androidruntion负责解释和执行生成的dalvik格式的字节码

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进行快速开发的。

应用程序层：该层是java的应用程序层，android内置的googlemaps、email、IM、浏览器等，都处于该层，java开发人员工发的程序也处于该层，而且和内置的应用程序具有平等的地位，可以调用内置的应用程序，也可以替换内置的应用程序

23.Activity 与 Task的启动模式有哪些，它们含义具体是什么？

答：在一个activity中，有多次调用startActivity来启动另一个activity，要想只生成一个activity实例，可以设置启动模式。

一个activity有四种启动模式：standed,signleTop,singleTask,singleInstance

Standed:标准模式，一调用startActivity()方法就会产生一个新的实例。

SingleTop:如果已经有一个实例位于activity栈顶，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。

singleTask:会在一个新的task中产生这个实例，以后每次调用都会使用这个，不会去产生新的实例了。

SingleInstance:这个和singleTask基本一样，只有一个区别：在这个模式下的activity实例所处的task中，只能有这个activity实例，不能有其他实例

24.Application类的作用：

答：API里的第一句是：

Base class for those who need to maintain global application state

如果想在整个应用中使用全局变量，在java中一般是使用静态变量，public类型；而在android中如果使用这样的全局变量就不符合Android的框架架构，但是可以使用一种更优雅的方式就是使用Application context。

首先需要重写Application，主要重写里面的onCreate方法，就是创建的时候，初始化变量的值。然后在整个应用中的各个文件中就可以对该变量进行操作了。

启动Application时，系统会创建一个PID，即进程ID，所有的Activity就会在此进程上运行。那么我们在Application创建的时候初始化全局变量，同一个应用的所有Activity都可以取到这些全局变量的值，换句话说，我们在某一个Activity中改变了这些全局变量的 值，那么在同一个应用的其他Activity中值就会改变

25.说明onSaveInstanceState() 和 onRestoreInstanceState()在什么时候被调用：

答：Activity 的 onSaveInstanceState() 和 onRestoreInstanceState()并不是生命周期方法，它们不同于 onCreate()、 onPause()等生命周期方法，它们并不一定会被触发。当应用遇到意外情况（如：内存不足、用户直接按Home键）由系统销毁 一个Activity时，onSaveInstanceState()才会被调用。但是当用户主动去销毁一个Activity时，例如在应用中按返回 键，onSaveInstanceState()就不会被调用。因为在这种情况下，用户的行为决定了不需要保存Activity的状态。通常 onSaveInstanceState()只适合于保存一些临时性的状态，而 onPause()适合于数据的持久化保存。

另外，当屏幕的方向发生了改变， Activity会被摧毁并且被重新创建，如果你想在Activity被摧毁前缓存一些数据，并且在Activity被重新创建后恢复缓存的数据。可以重 写Activity的 onSaveInstanceState() 和 onRestoreInstanceState()方法。

26.android的服务的生命周期？哪个方法可以多次被调用：

答：1)与采用Context.startService()方法启动服务有关的生命周期方法

onCreate() -> onStart() -> onDestroy()

onCreate()该方法在服务被创建时调用，该方法只会被调用一次，无论调用多少次startService()或bindService()方法，服务也只被创建一次。

onStart() 只有采用Context.startService()方法启动服务时才会回调该方法。该方法在服务开始运行时被调用。多次调用startService()方法尽管不会多次创建服务，但onStart() 方法会被多次调用。

onDestroy()该方法在服务被终止时调用。

2)与采用Context.bindService()方法启动服务有关的生命周期方法

onCreate() -> onBind() -> onUnbind() -> onDestroy()

onBind()只有采用Context.bindService()方法启动服务时才会回调该方法。该方法在调用者与服务绑定时被调用，当调用者与服务已经绑定，多次调用Context.bindService()方法并不会导致该方法被多次调用。

onUnbind()只有采用Context.bindService()方法启动服务时才会回调该方法。该方法在调用者与服务解除绑定时被调用。

如果先采用startService()方法启动服务,然后调用bindService()方法绑定到服务，再调用unbindService()方法解除绑定，最后调用

bindService()方法再次绑定到服务，触发的生命周期方法如下：

onCreate() ->onStart() ->onBind() ->onUnbind()[重载后的方法需返回true] ->onRebind()

27.android的broadcast的生命周期：

答：1)Broadcast receiver生命周期中仅有一个回调方法：

void onReceive(Context curContext, Intent broadcastMsg)

当接收器接收到一条broadcast消息，Android就会调用onReceiver(),并传递给它一个Intent对象，这个对象携带着那条 broadcast消息。我们认为仅当执行这个方式时，Broadcast receiver是活动的；这个方法返回时，它就终止了。这就是Broadcast receiver的生命周期。

2)由于Broadcast receiver的生命周期很短，一个带有活动的Broadcast receiver的进程是受保护的，以避免被干掉；但是别忘了有一点，Android会在任意时刻干掉那些携带不再活动的组件的进程，所以很可能会造成这个问题。

3)解决上述问题的方案采用一个Service来完成这项工作，Android会认为那个进程中（Service所在的进程）仍然有在活动的组件。

28.android view，surfaceview，glsurfaceview的区别：

答：SurfaceView是从View基类中派生出来的显示类，直接子类有GLSurfaceView和VideoView，可以看出GL和视频播放以及Camera摄像头一般均使用SurfaceView

SurfaceView和View最本质的区别在于，surfaceView是在一个新起的单独线程中可以重新绘制画面而View必须在UI的主线程中更新画面。那么在UI的主线程中更新画面 可能会引发问题，比如你更新画面的时间过长，那么你的主UI线程会被你正在画的函数阻塞。那么将无法响应按键，触屏等消息。

当使用surfaceView 由于是在新的线程中更新画面所以不会阻塞你的UI主线程。但这也带来了另外一个问题，就是事件同步。比如你触屏了一下，你需要surfaceView中 thread处理，一般就需要有一个event queue的设计来保存touch event，这会稍稍复杂一点，因为涉及到线程同步。

所以基于以上，根据游戏特点，一般分成两类。

1)被动更新画面的。比如棋类，这种用view就好了。因为画面的更新是依赖于 onTouch 来更新，可以直接使用 invalidate。因为这种情况下，这一次Touch和下一次的Touch需要的时间比较长些，不会产生影响。

2)主动更新。比如一个人在一直跑动。这就需要有一个单独的thread不停的重绘人的状态，避免阻塞main UI thread。

5、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

6、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，竖屏的时候还会执行onConfigurationChanged方法

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

1. 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

2. 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。

3. 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)

41. 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。

43. 请介绍下 Android 的数据存储方式。

- 一.SharedPreferences 方式
- 二.文件存储方式
- 三.SQLite 数据库方式
- 四.内容提供器（Content provider）方式
- 五. 网络存储方式

43. 请介绍下 ContentProvider 是如何实现数据共享的。

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。

44. 如何启用 Service，如何停用 Service。Android 中的 service 类似于 windows 中的 service，service 一般没有用户操作界面，它运行 于系统中不容易被用户发觉， 可以使用它开发如监控之类的程序。

- 一。步骤 第一步：继承 Service 类 public class SMSService extends Service { }  
第二步：在 AndroidManifest.xml 文件中的<application>节点里对服务进行配置: <service android:name=".DemoService" />
- 二。Context.startService()和 Context.bindService 服务不能自己运行，需要通过调用 Context.startService()或 Context.bindService()方法启动服务。这两个方法都可 以启动 Service，但是它们的使用场合有所不同。
  - 1.使用 startService()方法启用服务，调用者与服务之间没有关连，即使调用者退出了，服务 仍然运行。使用 bindService()方法启用服务，调用者与服务绑定在了一起，调用者一旦退出，服务也就 终止。
  - 2.采用 Context.startService()方法启动服务，在服务未被创建时，系统会先调用服务的 onCreate()方法， 接着调用 onStart()方法。如果调用 startService()方法前服务已经被创建， 多次调用 startService() 方法并 不会导致多次创建服务，但会导致多次调用 onStart()方法。采用 startService()方法启动的服务，只能调用 Context.stopService()方法结束服务，服务结束 时会调用 onDestroy()方法。
  - 3.采用 Context.bindService()方法启动服务，在服务未被创建时，系统会先调用服务的 onCreate()方法， 接着调用 onBind()方法。这个时候调用者和服务绑定在一起，调用者退出了，系统就会先调 用服务的 onUnbind()方法， 。接着调用 onDestroy()方法。如果调用 bindService()方法前服务已经被绑定，多次调用 bindService()方法并不会 导致多次创建服务及绑定(也就是说 onCreate()和 onBind()方法并不会被多次调用)。如果调用 者希望与正在绑定的服务 解除绑定，可以调用 unbindService()方法，调用该方法也会导致系统调用服务的 onUnbind()-->onDestroy()方法。
- 三。 Service 的生命周期 1.Service 常用生命周期回调方法如下： onCreate() 该方法在服务被创建时调用，该方法只会被调用一次，无论调用多少次 startService()或 bindService()方法， 服务也 只被创建一次。 onDestroy()该方法在服务被终止时调用。 2. Context.startService()启动 Service 有关的生命周期方法 onStart() 只有采用 Context.startService()方法启动服务时才会回调该方法。该方法在服务开 始运行时被调用。 多次调用 startService()方法尽管不会多次创建服务，但 onStart() 方法会被多次调用。 3. Context.bindService()启动 Service 有关的生命周期方法 onBind()只有采用 Context.bindService()方法启动服务时才会回调该方法。该方法在调用者与 服务绑定时被调用， 当调用者与 服务已经绑定，多次调用 Context.bindService()方法并不会导致该方法被多次调用。 onUnbind()只有采用 Context.bindService()方法启动服务时才会回调该方法。 该方法在调用者 与服务解除绑定时被调用。 备注： 1. 采用 startService()启动服务 Intent intent = new Intent(DemoActivity.this, DemoService.class); startService(intent); 2.Context.bindService()启动 Intent intent = new Intent(DemoActivity.this, DemoService.class); bindService(intent, conn, Context.BIND\_AUTO\_CREATE); //unbindService(conn);//解除绑定

2013-4-7号更新：

问的问题都比较底层，总结一下大体如下：

- 1.对多线程的运用和理解，及多线程之间handle的传值。  
我觉得这个问题问的好。多线程的理解很重要。但是开发的时候一般应该是用不到的。
- 2.对android虚拟机的理解，包括内存管理机制垃圾回收机制。
- 3.framework工作方式及原理，Activity是如何生成一个view的，机制是什么。
- 4.android本身的一些限制，比如apk包大小限制，读取大文件时的时间限制。



5.ANR是什么？哪些情况下会发生？开发时如何寻找ANR？

ANR, Application not respons, 我个人认为就是在UI线程进行耗时操作所造成的。Android的Activity反应时间是5秒，BroadCast是10秒，所以尽量不要做耗时的操作。开发的时候如果遇到了ANR，就看你操作的地方是否进行了耗时操作，是否在UI线程进行了耗时的网络请求或者I/O操作，避免这些就 OK了。

6.这个问题有必要看看，想当初毕业的时候对这个是多么的熟悉啊，现在都忘了，平时用的确实比较少。还有String,StringBuilder,StringBuffer这三个的区别。都有必要好好看看。

HashCode,equals 和 == : [http://blog.sina.com.cn/s/blog\\_4e90b73e01008pzz.html](http://blog.sina.com.cn/s/blog_4e90b73e01008pzz.html)

7.java集合的理解

<http://www.cnblogs.com/raymond19840709/archive/2008/08/18/1270062.html>

<http://blog.csdn.net/softwave/article/details/4166598>

8.Java:重写equals()和hashCode()

<http://zhangjunhd.blog.51cto.com/113473/71571/>

<http://oznyang.iteye.com/blog/30690>

结束语：

总结一点，我和朋友最近面试了几乎所有国内的稍大点的互联网公司（百度、腾讯、阿里、金山、小米、360、美团、去哪、网易、搜狐等等吧）、国企、电商（万达电商）、还有些小公司、安全类（360）的等等吧，上面都是我一点点整理出来的，希望大家认真的把每一个题弄懂，你去面试任何一家公司，我认为都是没有问题的。说几个点，注意一下。

1、数据结构和算法

一般情况下，工作中是用不到的，但是就是问，每个公司都问，而且有的公司还问的特别深，还让你手写代码。我觉得这个确实有点恶心，我个人也是很讨厌算法的，但是真是没办法，人家就问，你说怎么办？只能学呗。列个表，大家可以按照下面的去学习。不要一下看完，你看完也没用，你没有深入理解。每三天学一种，半年后，我觉得你应该会进步很多。然后可以看看“编程之美”等算法书，应该会有较大的提高。

常见的数据结构

线性：数组、链表、队列、堆栈、块状数组（数组+链表）、hash表、双端队列、位图（bitmap）

树：堆（大顶堆、小顶堆），trie树（字母树或字典树），后缀树，后缀数组，二叉排序/查找树，B+ /B- /B\*，AVL树，Treap，红黑树，splay树，线段树，树状数组

图：图

其它：并查集

常见算法

1) 基本思想：枚举，递归，分治，模拟，贪心，动态规划，剪枝，回溯

2) 图算法：深度优先遍历与广度优先遍历，最短路径，最小生成树，拓扑排序

3) 字符串算法：字符串查找，hash算法，KMP算法

4) 排序算法：冒泡、快排、桶排序

5) 动态规划：背包问题，最长公共子序列，最优二分检索树

6) 数论问题：素数问题，整数问题，进制转换，同余模运算

7) 排列组合：排序和组合算法

8) 其它：LCA与RMQ问题

2.网络层

HTTP协议、TCP/IP协议，这个在Android的面试中问的比较少，但是了解一下，或者你学的比较深入，优势就比较明显了。会是你的加分点。所以，系统并且深入的学习一下就非常有必要了。

3.Android的基础

这个肯定是必须的了，比如listview优化、UI特效的实现、数据库层的搭建、JNI的使用等等。参照上面的面试题就可以。

4.注意一些面试细节

1) 为什么离职？

2) 对薪水的期望值

3) 手机记得静音或者关机

4) 表达能力的提高

最后希望大家都能找到一份好工作。

2

1

 

顶 踩

刚表态过的朋友 (3 人)

  
superRiceK

  
15934726...

  
烽火戏佳人

  
yfr's`qq

  
城市猎人1...

  
Mys\_hu

  
xuhaiyue

  
做有意义...

  
lts321

  
serenity\_a...



评论 (6 个评论)

发表评论

作者的其他最新日志

全部

- 

liusy

2013-5-13 10:15

牛B

回复

- Android 仿大众点评团购购买条浮
  - Intent 调用大全+ 地图调用
  - Android 利用ViewPager、
  - Fragment + ListFragment +
  - 从1.5k到18k, 一个程序员的5年成
  - 分享一款最近很火源码-新浪微博
- 

分外清扬

2013-11-15 17:09

不简单啊

回复
- 

SnailGo

2015-4-17 17:26

非常谢谢分享。。。

回复
- 

SnailGo

2015-4-17 17:30

请问楼主现在在哪高就啊？？

回复
- 

justdoing

2015-7-3 16:29

很全面，要是每个问题都有简单答案或者思路就更好了

回复
- 

justdoing

2015-7-3 16:30

很全面，要是每个问题都有简单答案或者思路就更好了

回复

😊

您需要登录后才可以评论

登录 | 立即注册

评论