

superjunjin的专栏

记录android的学习之路

 目次视图

 摘要视图

 RSS 订阅

个人资料



superjunjin

恒

访问： 686506次

积分： 6342

等级： 6

排名： 第2375名

原创： 58篇 转载： 70篇

译文： 0篇 评论： 91条

文章搜索

文章分类

[android UI](#) (18)

[android 其他](#) (31)

[android 错误备忘](#) (25)

[android 面试](#) (8)

[java](#) (8)

[网络](#) (4)

[Linux](#) (7)

[杂文（经验&资讯&备忘）](#) (6)

[HTML5](#) (1)

[node](#) (2)

[angularJS](#) (7)

[ionic](#) (7)

[cloud](#) (1)

[js](#) (2)

[算法](#) (1)

[git](#) (1)

[eclipse](#) (1)

[cordova](#) (2)

[gulp](#) (1)

文章存档

[2016年03月](#) (3)

[2016年02月](#) (2)

[2015年04月](#) (12)

[2015年03月](#) (13)

[2015年02月](#) (3)

[【免费公开课】Gulp前端自动化教程](#) [【专家问答】陈绍英：大型IT系统性能测试实战](#) [【博客活动】有奖征文--走进VR开发世界](#)

android面试题总结加强再加强版（三）

标签： android 面试 手机 layout service 存储

2012-08-13 16:10 60769人阅读 [评论\(6\)](#) [收藏](#) [举报](#)

分类： [android 面试](#) (7) ▾

版权声明：本文为博主原创文章，未经博主允许不得转载。

[目录\(?\)](#)

[+]

26.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？

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...

27.如何退出Activity

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。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](#)。

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

展开

阅读排行

android面试题总结加强	(60709)
TCP/IP四层模型和OSI七层模型	(54703)
android面试题总结加强	(27460)
ionic的侧滑和下拉刷新效果	(23889)
关于listview的setAdapter	(21783)
handler和message传值	(20015)
ionic项目之图片的选择与	(19739)
关于android布局的两个属性	(19739)
Java中的字符串分割	(16798)
9.png malformed , No message	(16797)

评论排行

android最全面面试题71道题	(9)
错误：Binary XML file line 11	(7)
关于listview的setAdapter	(7)
ionic框架环境配置	(6)
android面试题总结加强	(6)
listview中有checkbox 滑动后	(6)
android面试题总结加强	(6)
PullToRefresh自定义下拉刷新	(4)
ionic项目之图片的选择与	(3)
android-BroadcastReceiver	(3)

推荐文章

*Android属性动画
ObjectAnimator源码简单分析
* Apache Flink fault tolerance源码剖析(一)
*HTTP 头部信息和错误码汇总详细解释，必收藏！
*Rebound-Android的弹簧动画库
*大型分布式网站术语分析
*Hadoop中Map端shuffle源码解析

最新评论

Android Service生命周期及用法
qq_26974719: 启动service也不一定会启动一个新的线程，默认启动的service都是在主线程的
ionic的侧滑和下拉刷新效果
willz2015: 设置spinner属性就可以设置svg动画了，spinner列表： http://ionicframe...
PullToRefresh自定义下拉刷新动画微笑: 学习了，多谢楼主！
android 混淆问题 (导出apk时报错)
xis238282: 你导出apk后安装到手机后怎么看报错信息呢，正式版本在控制台不输出错误信息了
使用return结束一个方法
Wendell_Sir: 顶
listview中有checkbox 滑动后状态
Just_Zian: 666
TCP/IP四层模型和OSI七层模型的区别
DonToretto: 学习了
云服务器--leancloud
LiuZhen_93: 赞哦
错误：Binary XML file line #23: \
风泊月: 收到速回啊
错误：Binary XML file line #23: \
风泊月: 你是怎么精确到那个布局文件的....

答：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。

Message Queue(消息队列)：用来存放通过Handler发布的消息，通常附属于某一个创建它的线程，可以通过Looper.myQueue()得到当前线程的消息队列

Handler：可以发布或者处理一个消息或者操作一个Runnable，通过Handler发布消息，消息将只会发送到与它关联的消息队列，然也只能处理该消息队列中的消息

Looper：是Handler和消息队列之间通讯桥梁，程序组件首先通过Handler把消息传递给Looper，Looper把消息放入队列。Looper也把消息队列里的消息广播给所有的

Handler：Handler接受到消息后调用handleMessage进行处理

Message：消息的类型，在Handler类中的handleMessage方法中得到单个的消息进行处理

在单线程模型下，为了线程通信问题，Android设计了一个Message Queue(消息队列)，线程间可以通过该Message Queue并结合Handler和Looper组件进行信息交换。下面将对它们进行分别介绍：

1. 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处理器，是Message的主要处理器，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。

3. 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messagequeue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MessageQueue，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：这个Message机制的大概流程：

1. 在Looper.loop()方法运行开始后，循环地按照接收顺序取出Message Queue里面的非NULL的Message。

2. 一开始Message Queue里面的Message都是NULL的。当Handler.sendMessage(Message)到Message Queue，该函数里面设置了那个Message对象的target属性是当前的Handler对象。随后Looper取出了那个Message，则调用该Message的target指向的Handler的dispatchMessage函数对Message进行处理。在dispatchMessage方法里，如何处理Message则由用户指定，三个判断，优先级从高到低：

1) Message里面的Callback，一个实现了Runnable接口的对象，其中run函数做处理工作；

2) Handler里面的mCallback指向的一个实现了Callback接口的对象，由其handleMessage进行处理；

3) 处理消息Handler对象对应的类继承并实现了其中handleMessage函数，通过这个实现的handleMessage函数处理消息。

由此可见，我们实现的handleMessage方法是优先级最低的！

3. Handler处理完该Message (updateUI) 后，Looper则设置该Message为NULL，以便回收！

在网上有很多文章讲述主线程和其他子线程如何交互，传送信息，最终谁来执行处理信息之类的，个人理解是最简单的方法——判断Handler对象里面的Looper对象是属于哪条线程的，则由该线程来执行！

1. 当Handler对象的构造函数的参数为空，则为当前所在线程的Looper；

2. Looper.getMainLooper()得到的是主线程的Looper对象，Looper.myLooper()得到的是当前线程的Looper对象。

29. 你如何评价Android系统？优缺点。

答：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。开放性对于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这样一个播放器程序，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。

30.谈谈android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；它是Android提供的用来存储一些简单配置信息的一种机制，采用了XML格式将数据存储到设备中。只能在同一个包内使用，不能在不同的包之间使用。

(2) 文件存储数据；文件存储方式是一种较常用的方法，在Android中读取/写入文件的方法，与Java中实现I/O的程序是完全一样的，提供了openFileInput()和openFileOutput()方法来读取设备上的文件。

(3) SQLite数据库存储数据；SQLite是Android所带的一个标准的数据库，它支持SQL语句，它是一个轻量级的嵌入式数据库。

(4) 使用ContentProvider存储数据；主要用于应用程序之间进行数据交换，从而能够让其他的应用保存或读取此Content Provider的各种数据类型。

(5) 网络存储数据；通过网络上提供给我们的存储空间来上传(存储)和下载(获取)我们存储在网络空间中的数据信息。

31. Android中Activity, Intent, Content Provider, Service有什么区别。

Activity：活动，是最基本的Android应用程序组件。一个活动就是一个用户可以操作的可视化用户界面，每一个活动都被实现为一个独立的类，并且从活动基类继承而来。

Intent：意图，描述应用想干什么。最重要的部分是动作和动作对应的数据。

Content Provider：内容提供器，Android应用程序能够将它们的数据保存到文件、SQLite数据库中，甚至是任何有效的设备中。当你想将你的应用数据和其他应用共享时，内容提供器就可以发挥作用了。

Service：服务，具有一段较长生命周期且没有用户界面的程序组件。

32. View, surfaceView, GLSurfaceView有什么区别。

view是最基础的，必须在UI主线程内更新画面，速度较慢。

SurfaceView 是view的子类，类似使用双缓机制，在新的线程中更新画面所以刷新界面速度比view快

GLSurfaceView 是SurfaceView的子类，opengl 专用的

33.Manifest.xml文件中主要包括哪些信息？

manifest：根节点，描述了package中所有的内容。

uses-permission：请求你的package正常运作所需赋予的安全许可。

permission：声明了安全许可来限制哪些程序能你package中的组件和功能。

instrumentation：声明了用来测试此package或其他package指令组件的代码。

application：包含package中application级别组件声明的根节点。

activity：Activity是用来与用户交互的主要工具。

receiver：IntentReceiver能使的application获得数据的改变或者发生操作，即使它当前不在运行。

service：Service是能在后台运行任意时间的组件。

provider：ContentProvider是用来管理持久化数据并发布给其他应用程序使用的组件。

34.根据自己的理解描述下Android数字签名。

(1)所有的应用程序都必须有数字证书，Android系统不会安装一个没有数字证书的应用程序

(2)Android程序包使用的数字证书可以是自签名的，不需要一个权威的数字证书机构签名认证

(3)如果要正式发布一个Android程序，必须使用一个合适的私钥生成的数字证书来给程序签名，而不能使用adt插件或者ant工具生成的调试证书来发布。

(4)数字证书都是有有效期的，Android只是在应用程序安装的时候才会检查证书的有效期。如果程序已经安装在系统中，即使证书过期也不会影响程序的正常功能。

35. 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文件自动生成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>

36.android:gravity与android:layout_gravity的区别

LinearLayout有两个非常相似的属性：android:gravity与android:layout_gravity。他们的区别在于：android:gravity用于设置View组件的对齐方式，而android:layout_gravity用于设置Container组件的对齐方式。

举个例子，我们可以通过设置`android:gravity="center"`来让`EditText`中的文字在`EditText`组件中居中显示；同时我们设置`EditText`的`android:layout_gravity="right"`来让`EditText`组件在`LinearLayout`中居右显示。来实践以下：正如我们所看到的，在`EditText`中，其中的文字已经居中显示了，而`EditText`组件自己也对齐到了`LinearLayout`的右侧。

```
[html]  
01. <LinearLayout  
02.     xmlns:android="http://schemas.android.com/apk/res/android"  
03.     android:orientation="vertical"  
04.     android:layout_width="fill_parent"  
05.     android:layout_height="fill_parent">  
06.     <EditText  
07.         android:layout_width="wrap_content"  
08.         android:gravity="center"  
09.         android:layout_height="wrap_content"  
10.         android:text="one"  
11.         android:layout_gravity="right"/>  
12.     </EditText>  
13. </LinearLayout>
```

38. 注册广播接收者两种方式的区别，及优缺点

答：首先写一个类要继承`BroadcastReceiver`

第一种：在清单文件中声明，添加

```
<receiver android:name=".IncomingSMSReceiver" >  
<intent-filter>  
    <action android:name="android.provider.Telephony.SMS_RECEIVED" />  
</intent-filter>  
<receiver>
```

第二种使用代码进行注册如：

```
IntentFilter filter = new IntentFilter("android.provider.Telephony.SMS_RECEIVED");  
IncomingSMSReceiver receiver = new IncomingSMSReceiver();  
registerReceiver(receiver, filter);
```

两种注册类型的区别是：

1) 第一种是常驻型（静态注册），也就是说当应用程序关闭后，如果有信息广播来，程序也会被系统调用自动运行。

2) 第二种不是常驻型广播（动态注册），也就是说广播跟随程序的生命周期。

注册的方法有两种，一种是静态注册，一种是动态注册。

动态注册优点：在 Android 的广播机制中，动态注册的优先级是要高于静态注册优先级的，因此在必要的情况下，我们是需要动态注册广播接收器的。

静态注册优点：动态注册广播接收器还有一个特点，就是当用来注册的 Activity 关掉后，广播也就失效了。同时反映了静态注册的一个优势，就是无需担忧广播接收器是否被关闭，只要设备是开启状态，广播接收器就是打开着的。

39. Dalvik 基于 JVM 的改进

1. 几个 class 变为一个 dex，constant pool，省内存
2. Zygote，copy-on-write shared，省内存，省 CPU，省电
3. 基于寄存器的 bytecode，省指令，省 CPU，省电
4. Trace-based JIT，省 CPU，省电，省内存

40. 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。

Ø DOM 解析

优点：

1. XML 树在内存中完整存储，因此可以直接修改其数据和结构。
2. 可以通过该解析器随时访问 XML 树中的任何一个节点。
3. DOM 解析器的 API 在使用上也相对比较简单。

缺点：如果 XML 文档体积比较大时，将文档读入内存是非常消耗系统资源的。

使用场景：DOM 是用与平台和语言无关的方式表示 XML 文档的官方 W3C 标准。DOM 是以层次结构组织的节点的集合。这个层次结构允许开发人员在树中寻找特定信息。分析该结构通常需要加载整个文档和构造层次结构，然后才能进行任何工作。DOM 是基于对象层次结构的。

Ø SAX 解析

优点：

SAX 对内存的要求比较低,因为它让开发人员自己来决定所要处理的标签.特别是当开发人员只需要处理文档中所包含的部分数据时,SAX 这种扩展能力得到了更好的体现.

缺点:

用SAX方式进行XML解析时,需要顺序执行,所以很难访问到同一文档中的不同数据.此外,在基于该方式的解析编码过程也相对复杂.

使用场景:

对于含有数据量十分巨大,而又不用对文档的所有数据进行遍历或者分析的时候,使用该方法十分有效.该方法不用将整个文档读入内存,而只需读取到程序所需的文档标签处即可.

Ø Xmlpull解析

android SDK提供了xmlpull api,xmlpull和sax类似,是基于流 (stream) 操作文件,然后根据节点事件回调开发者编写的处理程序.因为是基于流的处理,因此xmlpull和sax都比较节约内存资源,不会象dom那样要把所有节点以对橡树的形式展现在内存中.xmlpull比sax更简明,而且不需要扫描完整个流.

41.Android系统中GC什么情况下会出现内存泄露呢?

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]

42.谈谈对Android NDK的理解

NDK全称 : Native Development Kit。

1、NDK是一系列工具的集合。

* NDK提供了一系列的工具 , 帮助开发者快速开发C (或C++) 的动态库 , 并能自动将so和java应用一起打包成apk。这些工具对开发者的帮助是巨大的。

* NDK集成了交叉编译器 , 并提供了相应的mk文件隔离CPU、平台、ABI等差异 , 开发人员只需要简单修改mk文件 (指出“哪些文件需要编译”、“编译特性要求”等) , 就可以创建出so。

* NDK可以自动地将so和Java应用一起打包 , 极大地减轻了开发人员的打包工作。

2、NDK提供了一份稳定、功能有限的API头文件声明。

Google明确声明该API是稳定的 , 在后续所有版本中都稳定支持当前发布的API。从该版本的NDK中看出 , 这些API支持的功能非常有限 , 包含有 : C标准库 (libc) 、 标准数学库 (libm) 、 压缩库 (libz) 、 Log库 (liblog) 。

顶 踩

21 1

上一篇 [Android面试之----谈谈你对Java中Native关键字的认识!](#)

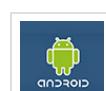
下一篇 [android面试题总结加强再加强版 \(四\)](#)

我的同类文章

android 面试 (7)

- [android面试题总结加强再... 2012-08-13 阅读 12428](#)
- [Android面试之--谈谈你对... 2012-08-13 阅读 3324](#)
- [android面试题总结加强再... 2012-08-12 阅读 8927](#)
- [android最全面面试题71道题 ... 2012-07-22 阅读 9518](#)
- [Android面试之----谈谈你对... 2012-08-13 阅读 5252](#)
- [说一下你眼中的Android的... 2012-08-12 阅读 4714](#)
- [android面试题总结加强再... 2012-08-11 阅读 27424](#)

参考知识库



Android知识库

13735 关注 | 1525 收录



Java EE知识库

2398 关注 | 618 收录



MySQL知识库

9627 关注 | 1360 收录



Java Web知识库

10867 关注 | 1078 收录



Java SE知识库

10547 关注 | 454 收录



Docker知识库

2542 关注 | 185 收录

猜你在找

Android入门实战教程

android面试题总结加强版二

数据结构 (C版)

android面试题总结加强版一

8小时学会HTML网页开发

android面试题总结加强版三

Android高手进阶

Android面试题总结加强版三

微信公众平台开发入门

Android之面试题总结加强版一

查看评论

6楼 AudienL 2015-08-19 01:25发表



好东西

5楼 dxpqxb 2012-08-27 17:03发表



真的没什么好帖子可以顶了

4楼 小苏菲 2012-08-24 22:41发表



这种文章也能上首页？

3楼 ting0117 2012-08-23 17:16发表



正在找工作，很好很强大。

2楼 ParadiseOK 2012-08-23 07:46发表



不错的资料啊

1楼 大明ZeroSon 2012-08-22 09:16发表



Content Provider这个是数据共享，把数据库共享出去，不是存储。把这个说成存储有点勉强。

Re: 葫芦娃今晚打老虎 2014-07-09 09:34发表



回复大明ZeroSon：我这正想说，ContentProvider只是一个数据共享的标准接口，不能说是存储

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

