

最新Android面试题

1、 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？

会，比如NullPointerException。我遇到过，比如textView.setText()时，textView没有初始化。会导致程序无法正常运行出现forceclose。打开控制台查看logcat信息找出最新Android面试题异常信息并修改程序。

2、 IntentService有何优点？

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

3、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？

重写 onSaveInstanceState() 方法，在此方法中保存需要保存的数据，该方法将会在activity被回收之前调用。通过重写 onRestoreInstanceState() 方法可以从中提取保存好的数据

4、如何将一个Activity设置成窗口的样式。

<activity>中配置： android :theme="@android:style/Theme.Dialog"

另外 android:theme="@android:style/Theme.Translucent" 是设置透明

5、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。

对于多个 activity，1、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。2、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。3、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。为了编程方便，最好定义一个 Activity 基类，处理这些共通问题。

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，`killBackground Processes()`，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，第三万程序无法调用。

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。

6、AIDL的全称是什么？如何工作？能处理哪些类型的数据？

全称是：**AndroidInterface Define Language**

在 `Android` 中，每个应用程序都可以有自己的进程。在写 **UI** 应用的时候，经常要用到 **Service**。在不同的进程中，怎样传递对象呢？显然，**Java** 中不允许跨进程内存共享。因此传递对象，只能把对象拆分成操作系统能理解的简单形式，以达到跨界对象访问的目的。在

J2EE 中,采用 RMI 的方式,可以通过序列化传递对象.在 Android 中,则采用 AIDL 的方式.理论上 AIDL 可以传递 Bundle,实际上做起来却比较麻烦。

AIDL(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的目的. 如果需要在 Activity 中,访问另一个 Service 中的某个对象,需要先将对象转化成 AIDL 可识别的参数(可能是多个参数),然后使用 AIDL 来传递这些参数,在消息的接收端,使用这些参数组装成自己需要的对象。

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.

AIDL 的创建方法:

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.

7、请解释下Android程序运行时权限与文件系统权限的区别。

运行时权限 Dalvik(android 授权)

文件系统 linux 内核授权

8、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

9、android系统的优势和不足

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格购得心仪的手机。

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。

10、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。

11、sim卡的EF文件是什么？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

12、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

13、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。

14、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。

15、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像素点透明(0xffffffff的透明度调整为50%。

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

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

17、如何将打开res aw目录中的数据库文件？

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、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

19、java中如何引用本地语言

可以用JNI（java native interface java 本地接口）接口

20、谈谈Android的IPC（进程间通信）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。

21、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

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，通过它可以给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提供的数据库。

20、 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。

13.handler机制原理；

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。

Activity 的生命周期

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。

10. 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。

6、Intent传递数据时，下列的数据类型哪些可以被传递（abcd）(多选)

A、Serializable B、charsequence C、Parcelable D、Bundle

16、如何将一个Activity设置成窗口的样式。

在AndroidManifest.xml 中定义Activity的地方一句话android:theme="@android:style/Theme.Dialog"或android:theme="@android:style/Theme.Translucent"就变成半透明的

15、 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？

onSaveInstanceState()

当你的程序中某一个Activity A在运行时，主动或被动地运行另一个新的Activity B，这个时候A会执行onSaveInstanceState()。B完成以后又会来找A，这个时候就有两种情况：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()了。

17、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。

19、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.内容提供者（Content provider）方式

五. 网络存储方式

ShareaPreferences/Sqlite 文件、Content provider 网络

Service 启动方式

1.使用startService()方法启用服务，调用者与服务之间没有关联，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。

24、AIDL的全称是什么？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

29、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

31、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。

33、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

37、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。

44、说说mvc模式的原理,它在android中的运用

MVC(Model_view_controller)”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

45、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。

46、java中如何引用本地语言

可以用JNI (java native interface java 本地接口) 接口。

48、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

35. 简述 Android 应用程序结构是哪些? (7 分)

Android 应用程序结构是：

Linux Kernel(Linux 内核)、Libraries(系统运行库或者是 c/c++核心库)、Application

Framework(开发框架包)、Applications (核心应用程序)

55. 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

58. 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)

68. java 中如何引用本地语言

可以用 JNI (java native interface java 本地接口) 接口。

- 为什么要用 ContentProvider? 它和 sql 的实现上有什么差别?
使用 **ContentProvider** 可以将数据共享给其他应用，让除本应用之外的应用也可以访问本应用的数据。它的底层是用 **SQLite** 数据库实现的，所以其对数据做的各种操作都是以 **Sql** 实现，只是在上层提供的是 **Uri**。