3.BroadcastReceiver

3.1.广播的分类和使用场景

Android 广播分为两个角色：广播发送者、广播接受者

广播接收器的注册分为两种：静态注册、动态注册。

静态广播接收者：通过AndroidManifest.xml的标签来申明的BroadcastReceiver。

动态广播接收者：通过AMS.registerReceiver()方式注册的BroadcastReceiver，动态注册更为灵活，可在不需要时通过unregisterReceiver()取消注册。

广播类型：根据广播的发送方式，

1.**普通广播**：通过Context.sendBroadcast()发送，可并行处理

2.**系统广播**：当使用系统广播时，只需在注册广播接收者时定义相关的action即可，不需要手动发送广播(网络变化,锁屏,飞行模式)

3.**有序广播**： 指的是发送出去的广播被 BroadcastReceiver 按照先后顺序进行接收 发送方式变为：sendOrderedBroadcast(intent);

广播接受者接收广播的顺序规则（同时面向静态和动态注册的广播接受者）：按照 Priority 属性值从大-小排序，Priority属性相同者，动态注册的广播优先。

4.**App应用内广播**（Local Broadcast）

背景 Android中的广播可以跨App直接通信（exported对于有intent-filter情况下默认值为true）

冲突 可能出现的问题：

其他App针对性发出与当前App intent-filter相匹配的广播，由此导致当前App不断接收广播并处理；

其他App注册与当前App一致的intent-filter用于接收广播，获取广播具体信息； 即会出现安全性 & 效率性的问题。

解决方案 使用App应用内广播（Local Broadcast）

App应用内广播可理解为一种局部广播，广播的发送者和接收者都同属于一个App。 相比于全局广播（普通广播），App应用内广播优势体现在：安全性高 & 效率高

具体使用1 - 将全局广播设置成局部广播

注册广播时将exported属性设置为false，使得非本App内部发出的此广播不被接收； 在广播发送和接收时，增设相应权限permission，用于权限验证；

发送广播时指定该广播接收器所在的包名，此广播将只会发送到此包中的App内与之相匹配的有效广播接收器中。

具体使用2 - 使用封装好的LocalBroadcastManager类

对于LocalBroadcastManage,方式发送的应用内广播，只能通过LocalBroadcastManager动态注册，不能静态注册

5.**粘性广播**（Sticky Broadcast） 由于在Android5.0 & API 21中已经失效，所以不建议使用，在这里也不作过多的总结。

应用场景

同一 App 内部的不同组件之间的消息通信（单个进程）；

不同 App 之间的组件之间消息通信；

Android系统在特定情况下与App之间的消息通信，如：网络变化、电池电量、屏幕开关等。

3.2.广播的两种注册方式的区别

静态注册：常驻系统，不受组件生命周期影响，即便应用退出，广播还是可以被接收，耗电、占内存。

动态注册：非常驻，跟随组件的生命变化，组件结束，广播结束。在组件结束前，需要先移除广播，否则容易造成内存泄漏。

3.3.广播发送和接收的原理

https://juejin.im/post/6844904057891471367#heading-0

http://gityuan.com/2016/06/04/broadcast-receiver/

动态注册

1.创建对象LoadedApk.ReceiverDispatcher.InnerReceiver的实例，该对象继承于IIntentReceiver.Stub（InnerReceiver实际是一个binder本地对象(BBinder：本地Binder，服务实现方的基类，提供了onTransact接口来接收请求)）。

2.将IIntentReceiver对象和注册所传的IntentFilter对象发送给AMS。 AMS记录IIntentReceiver、IntentFilter和注册的进程ProcessRecord，并建立起它们的对应关系。

3.当有广播发出时，AMS根据广播intent所携带的IntentFilter找到IIntentReceiver和ProcessRecord，然后回调App的ApplicationThread对象的scheduleRegisteredReceiver，将IIntentReceiver和广播的intent一并传给App，App直接调用IIntentReceiver的performReceive。

4.因为广播是通过binder线程回调到接收进程的，接收进程通过ActivityThread里的H这个Handler将调用转到主线程，然后回调BroadcastReceiver的onReceive。

静态注册

静态注册是通过在Manifest文件中声明实现了BroadcastReceiver的自定义类和对应的IntentFilter，来告诉PMS(PackageManagerService)这个App所注册的广播。

当AMS接收到广播后，会查找所有动态注册的和静态注册的广播接收器，静态注册的广播接收器是通过PMS(PackageManagerService)发现的，PMS找到对应的App

对应进程已经创建，直接调用App的ApplicationThread对象的scheduleReceiver

对应进程尚未创建，先启动App进程，App进程启动后回调AMS的attachApplication，attachApplication则继续派发刚才的广播App这边收到调用后会先通过Handler转到主线程，然后根据AMS传过来的参数实例化广播接收器的类，接着调用广播接收器的onReceive。

3.4.本地广播和全局广播的区别

BroadcastReceiver是针对应用间、应用与系统间、应用内部进行通信的一种方式

LocalBroadcastReceiver仅在自己的应用内发送接收广播，也就是只有自己的应用能收到，数据更加安全广播只在这个程序里，而且效率更高。

BroadcastReceiver采用的binder方式实现跨进程间的通信；

LocalBroadcastManager使用Handler通信机制。

LocalBroadcastReceiver 使用

LocalBroadcastReceiver不能静态注册，只能采用动态注册的方式。

在发送和注册的时候采用，LocalBroadcastManager的sendBroadcast方法和registerReceiver方法

http://gityuan.com/2017/04/23/local_broadcast_manager/

注册过程，主要是向mReceivers和mActions添加相应数据：

mReceivers：数据类型为HashMap<BroadcastReceiver, ArrayList>， 记录广播接收者与IntentFilter列表的对应关系；

mActions：数据类型为HashMap<String, ArrayList>， 记录action与广播接收者的对应关系

根据Intent的action来查询相应的广播接收者列表；

发送MSG_EXEC_PENDING_BROADCASTS消息，回调相应广播接收者的onReceive方法