



SUN YAT-SEN UNIVERSITY

中山大學



手机应用平台软件开发

6、Broadcast使用



Broadcast（广播）有何用？（1）

SUN YAT-SEN UNIVERSITY

□ 场景一

如果你在等快递，你是每天24小时守候在快递网点等待包裹到来，还是该干嘛干嘛去，等到包裹到了有人打电话通知你？

□ 场景二

我们需要程序在手机来电或接收短信时，显示电话号码和号码归属地，或者号码在我们黑名单中的话自动挂断或删除短信，要如何实现？



Broadcast（广播）有何用？（2）

SUN YAT-SEN UNIVERSITY

□ 场景一 相信答案很明确

□ 场景二

在Symbian、Windows Mobile中，应用若需等待一个来电或短信，来实现显示归属地之类的功能，必须让自己的应用保证开机启动、潜伏在后台运行、监控相关事件。

在Android平台中，考虑到广泛存在这类需求，在框架中设计了BroadcastReceiver。当发生这类事件时，系统会自动唤醒负责接收对应事件的Receiver，处理完事件后，Receiver就马上退出，这对手机有限的资源来说是一种极好的解决方案



Broadcast Receiver 简介

SUN YAT-SEN UNIVERSITY

- Android中的四大组件分别是 Activity、Service、Broadcast和 Content Provider。
 - 而Intent是一个对动作和行为的抽象描述，负责组件之间与程序之间进行消息传递。
 - 因此，Broadcast Receiver组件提供了一种把Intent作为一个消息广播出去，由所有对其感兴趣的程序对其作出反应的机制。
-



还记得什么是Intent吗？（1）

SUN YAT-SEN UNIVERSITY

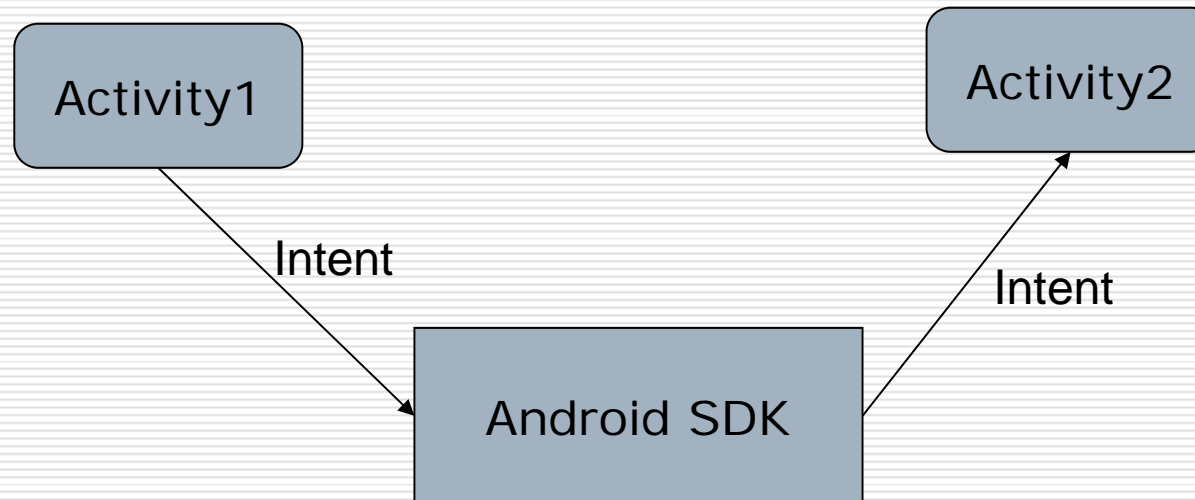
- **Intent**作为一种操作系统级别的消息传递机制，
能够在不同的进程之间传递**结构化消息**。
-



还记得什么是Intent吗？（2）

SUN YAT-SEN UNIVERSITY

- 不同Activity之间的通讯，属于不同UI线程之间的通讯，如果要在不同的Activity之间传递消息，则需要用到Intent。在Intent中告诉操作系统要选择哪个目标Activity进行实例化，并传递消息。





还记得什么是Intent吗？（3）

SUN YAT-SEN UNIVERSITY

□ 以前的实验代码

```
Intent intent = new Intent();  
intent.setClass(SrcActivity.this, DstActivity.class);  
intent.putExtra("msg", "message from SrcActivity");  
startActivity(intent);
```

□ Intent中包括

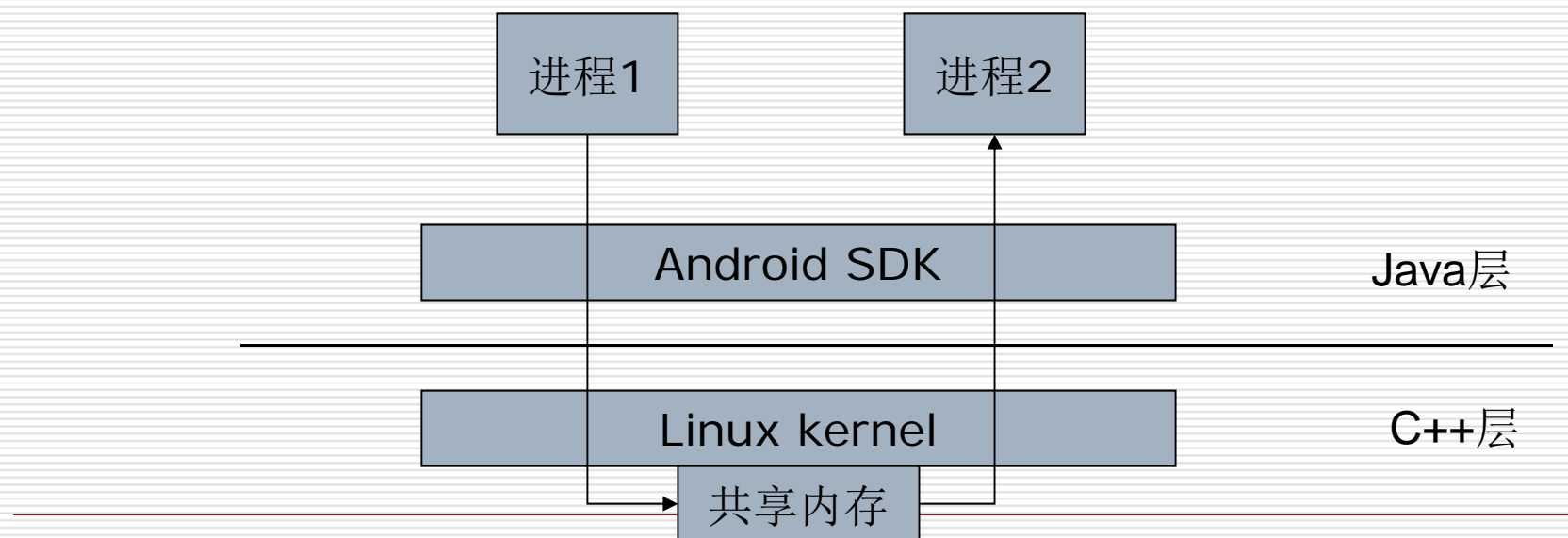
- 源Activity
 - 目标Activity
 - 要传递的消息
-



关于消息传递

SUN YAT-SEN UNIVERSITY

- 在Service实验中也涉及到进程间的通讯，但是在Service实验中我们用的是Binder。
- Binder实质上是以IPC（Inter-Process Communication，进程间通信）框架为基础。可以简单按下图理解，其实质就是通过共享内存实现进程间的通讯。





Broadcast广播机制（1）

SUN YAT-SEN UNIVERSITY

Android广播机制包含三个基本要素：

- Broadcast（广播） ---发送广播；
 - BroadcastReceiver（广播接收器）
---接收广播；
 - Intent（意图） ---保存广播信息的媒介
-



Broadcast广播机制（2）

SUN YAT-SEN UNIVERSITY

- Broadcast是一种广泛运用的在应用程序之间传输信息的机制。
 - BroadcastReceiver是对发送出来的Broadcast进行过滤, 接收并响应的一类组件。
-



Broadcast 广播机制 (3)

SUN YAT-SEN UNIVERSITY

广播的详细过程

- **发送广播** - 在需要发送信息的地方，把要发送的信息和用于过滤的信息(如Action、Category)装入一个Intent对象，调用Context.sendBroadcast() 方法把Intent对象以广播方式发送出去。
 - **接收广播** - 当Intent发送以后，所有已经注册的BroadcastReceiver会检查注册时的IntentFilter是否与发送的Intent相匹配，若匹配则就会调用BroadcastReceiver的void onReceive(Context curContext, Intent broadcastMsg)方法。
-



Broadcast--最简单的例程(1)

SUN YAT-SEN UNIVERSITY

实现一个自定义的Broadcast

■ 发送广播类: TestActivity.java

```
mBtnTest = (Button)findViewById(R.id.btnTest);  
mBtnTest.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
  
        Intent intent = new Intent("TheStringUsedAsFilter");  
        sendBroadcast(intent);  
    }  
});
```

■ 接收广播类: MyBroadcast1.java

```
public class MyBroadCast1 extends BroadcastReceiver {  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        // TODO Auto-generated method stub  
        Log.v("BroadCast1", "Receive");  
    }  
  
}
```



Broadcast----最简单的例程(2)

SUN YAT-SEN UNIVERSITY

- 配置文件AndroidManifest.xml:
用<Receiver>标签注册一个BroadcastReceiver, 还需要有一个字符串作为filter, 通过filter选择接收广播的类。

```
<receiver android:name=".MyBroadCast1">  
  <intent-filter>  
    <action android:name="TheStringUsedAsFilter"></action>  
  </intent-filter>  
</receiver>
```

- 要捕捉的信息名称为TheStringUsedAsFilter
-



Broadcast----最简单的例程(3)

SUN YAT-SEN UNIVERSITY

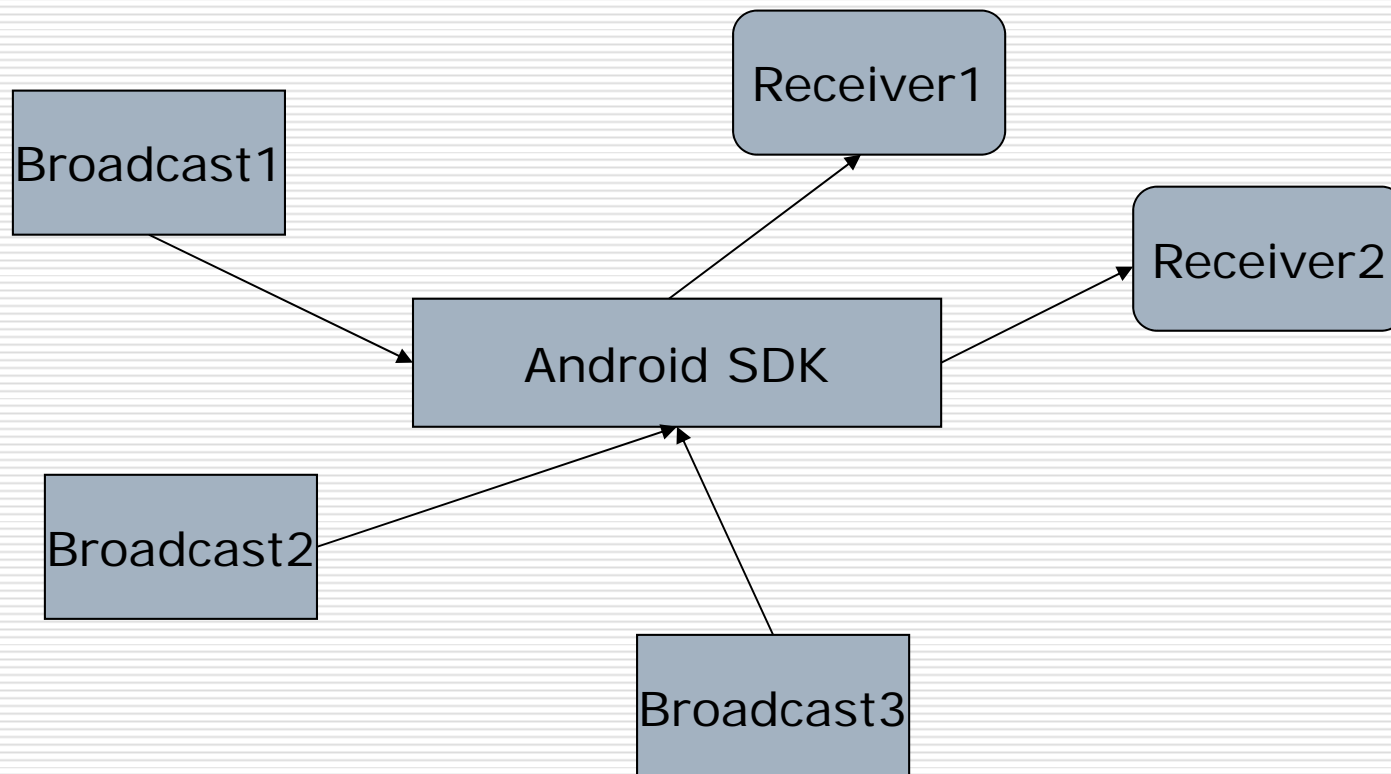
整个流程如下:

- 在AndroidManifest.xml配置文件中, 用<Receiver>标签注册一个BroadcastReceiver, 还需要有一个字符串作为过滤filter, 通过filter选择接收广播的类。
 - TestActivity.java中将filter字符串放入intent中, 再通过广播发出去, 等待系统接收。
 - 系统通过xml文件, 查找到对应的filter, 映射到对应的BroadcastReceiver类。
-



Broadcast机制图示

SUN YAT-SEN UNIVERSITY

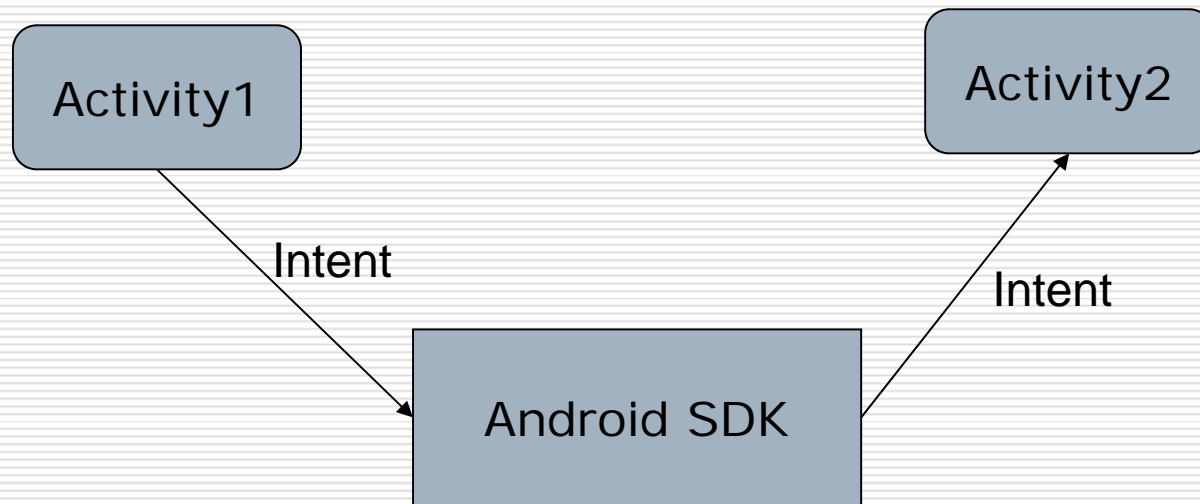




Broadcast机制

SUN YAT-SEN UNIVERSITY

- Broadcast机制与Intent消息机制的图是一致的。





Broadcast机制与Intent机制

SUN YAT-SEN UNIVERSITY

□ 通过Intent实现Activity间的跳转有两种方式:

1. 显式Intent: 即在构造Intent对象时就指定接收者;
2. 隐式Intent: 即Intent的发送者在构造Intent对象时, 并不知道也不关心接收者是谁, 只通过filter来选择接收者, 有利于降低发送者和接收者之间的耦合。

□ 而Broadcast的实现机制, 与通过隐式的Intent启动Activity的方式是类似的。



Broadcast的生命周期

SUN YAT-SEN UNIVERSITY

- 广播接收器**仅在它执行这个方法时处于活跃状态**。当**onReceive()**返回后，它即为失活状态。
- 拥有一个活跃状态的广播接收器的进程被保护起来而不会被杀死，但**仅拥有失活状态组件**的进程则会在其它进程需要它所占有的内存的时候随时被杀掉。所以，如果响应一个广播信息需要很长的一段时间，我们一般会将其纳入一个衍生的线程中去完成，而不是在主线程内完成它，从而保证用户交互过程的流畅。





Broadcast的生命周期

SUN YAT-SEN UNIVERSITY

- ❑ 当系统通过filter字符串找到匹配的Receiver，onReceive方法就会被执行。
 - ❑ onReceive方法必须在5秒内执行完毕退出，否则会导致FC（Force Close强制关闭）。
-



BroadCast的应用

SUN YAT-SEN UNIVERSITY

- 通常来说，BroadcastReceiver用来更新content、启动service、更新UI或者通过notification manager在状态栏中提示。
 - 5秒的限制保证了主要处理任务不会、也不应该在BroadcastReceiver中完成。
 - 如果需要执行大量任务，可以在BroadcastReceiver中启用Service去处理。
-



BroadCast应用--手机自动服务

SUN YAT-SEN UNIVERSITY

- 在Android操作系统中，有许多与手机相关的事件会对系统发送广播信息。当系统发出广播后，就会搜索是否注册了负责处理该广播的BroadcastReceiver。
 - 因此，只要了解系统的广播类型，就可以实现很多手机自动服务功能。例如，收到新短信进行提示、手机来电时自动拒接等等。
-



Android中Broadcast的Intent（部分）

SUN YAT-SEN UNIVERSITY

各SDK版本Intent各有增减

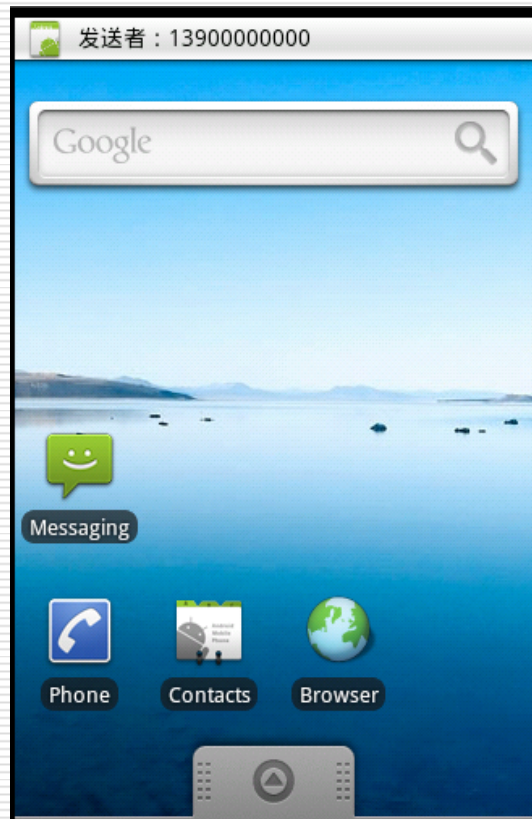
```
android.bluetooth.a2dp.intent.action.SINK_STATE_CHANGED
android.bluetooth.intent.action.BLUETOOTH_STATE_CHANGED
android.bluetooth.intent.action.BOND_STATE_CHANGED_ACTION
android.bluetooth.intent.action.DISCOVERY_COMPLETED
android.bluetooth.intent.action.DISCOVERY_STARTED
android.bluetooth.intent.action.HEADSET_AUDIO_STATE_CHANGED
android.bluetooth.intent.action.HEADSET_STATE_CHANGED
android.bluetooth.intent.action.NAME_CHANGED
android.bluetooth.intent.action.PAIRING_CANCEL
android.bluetooth.intent.action.PAIRING_REQUEST
android.bluetooth.intent.action.REMOTE_DEVICE_CONNECTED
android.bluetooth.intent.action.REMOTE_DEVICE_DISAPPEARED
android.bluetooth.intent.action.REMOTE_DEVICE_DISAPPEARED
android.bluetooth.intent.action.REMOTE_DEVICE_DISCONNECTED
android.bluetooth.intent.action.REMOTE_DEVICE_DISCONNECT_REQUESTED
android.bluetooth.intent.action.REMOTE_DEVICE_FOUND
android.bluetooth.intent.action.REMOTE_NAME_FAILED
android.bluetooth.intent.action.REMOTE_NAME_UPDATED
```

```
android.bluetooth.intent.action.SCAN_MODE_CHANGED
android.intent.action.AIRPLANE_MODE
android.intent.action.BATTERY_CHANGED
android.intent.action.BATTERY_LOW
android.intent.action.BOOT_COMPLETED
android.intent.action.CAMERA_BUTTON
android.intent.action.CONFIGURATION_CHANGED
android.intent.action.DATA_SMS_RECEIVED
android.net.wifi.NETWORK_IDS_CHANGED
android.net.wifi.RSSI_CHANGED
android.net.wifi.SCAN_RESULTS
android.net.wifi.STATE_CHANGE
android.net.wifi.WIFI_STATE_CHANGED
android.net.wifi.suplicant.CONNECTION_CHANGE
android.net.wifi.suplicant.STATE_CHANGE
android.provider.Telephony.SIM_FULL
android.provider.Telephony.SMS_RECEIVED
android.provider.Telephony.WAP_PUSH_RECEIVED
com.google.gservices.intent.action.GSERVICES_CHANGED
```

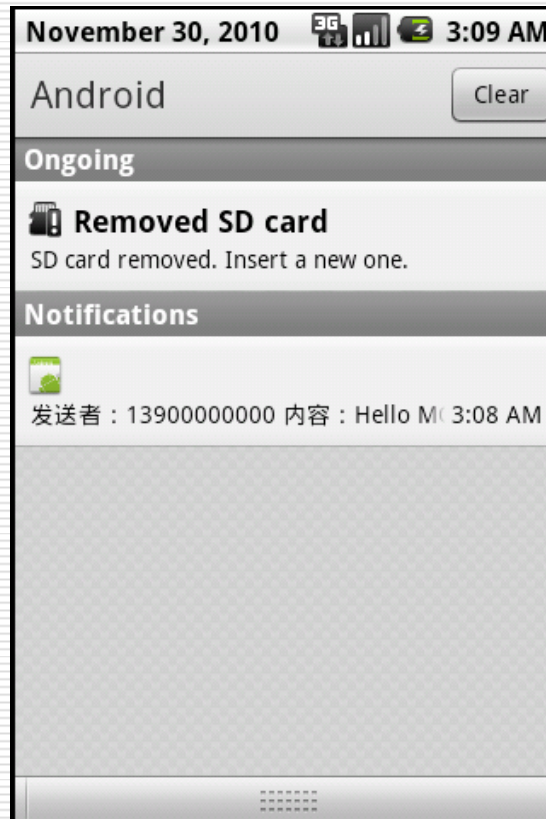


新短信提示（1）

SUN YAT-SEN UNIVERSITY



收到新短信，状态栏中有提示



拉下状态栏



点击查看详情



新短信提示-----AndroidManifest.xml (1)

SUN YAT-SEN UNIVERSITY

在AndroidManifest.xml里面，注册一个Receiver，并添加intent-filter节点，在节点下添加action，告诉系统这个Receiver用于处理哪些Broadcast（可以有多个action）

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




新短信提示-----AndroidManifest.xml (2)

SUN YAT-SEN UNIVERSITY

□ 定义一个继承**BroadcastReceiver**的子类**SmsReceiver**, 对系统“收到短信”这个广播进行处理。

□ 配置文件中

■ receiver名为“.SmsReceiver”

■ action名为“android.provider.Telephony.SMS_RECEIVED”

(查看系统广播intent大全)

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



新短信提示----SmsReceiver.java

SUN YAT-SEN UNIVERSITY

- ❑ 定义一个继承BroadcastReceiver的子类
- ❑ 重写onReceive方法

```
public class SmsReceiver extends BroadcastReceiver {  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
  
    }  
  
}
```



新短信提示----SmsReceiver.java

SUN YAT-SEN UNIVERSITY

- 如果一个Receiver可以处理多个广播，则需要在代码中对各个Action进行判断，分别处理。

```
if(intent.getAction().equals(mAction))
{
    /*在这里对action进行判断，例如我们要对“收信息”这个广播事件进行处理的话，
    * 则mAction = "android.provider.Telephony.SMS_RECEIVED" */
}
```

- 其中intent变量是由onReceive方法参数传入的，所以可以看出，系统广播其实也是通过sendBroadcast(intent)发出的。
 - 系统广播“收信息”中，还将短信内容附在Bundle中，随着intent传入到Receiver。
-



新短信提示----SmsReceiver.java

SUN YAT-SEN UNIVERSITY

□ 获取短信

```
Bundle bundle = intent.getExtras();
if(bundle != null)
{
    //bundle通过key=pdu取得短信的内容
    //这里myObjpdu是一个Object数组，因为当短信长度超出限制时，
    //会被自动拆分成多条短信
    myObjpdu = (Object[])bundle.get("pdu");
    //根据Object数组长度创建SmsMessage数组
    messages = new SmsMessage[myObjpdu.length];
    for(int i=0;i<myObjpdu.length;i++)
    {
        //通过SmsMessage的静态方法createFromPdu通过myObjpdu创建SmsMessage对象
        messages[i] = SmsMessage.createFromPdu((byte[])myObjpdu[i]);
    }
}
```



新短信提示-----SmsReceiver.java

SUN YAT-SEN UNIVERSITY

- ❑ SmsMessage对象可以通过getDisplayOriginatingAddress(), getDisplayMessageBody()方法分别获得发送人手机号和短信内容。
 - ❑ 注意：如果长信息被拆成几个短信息时，发信人手机号会被保存在SmsMessage数组的最后一个元素中。
-

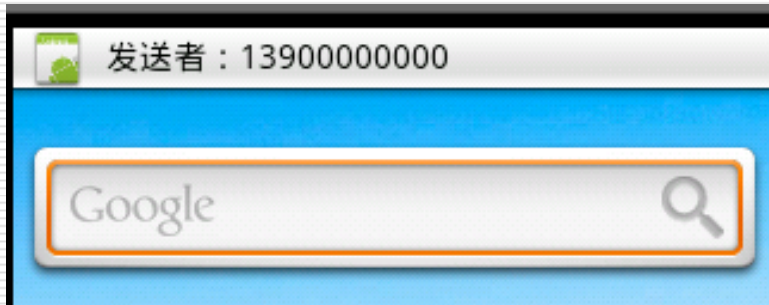


新短信提示----SmsReceiver.java

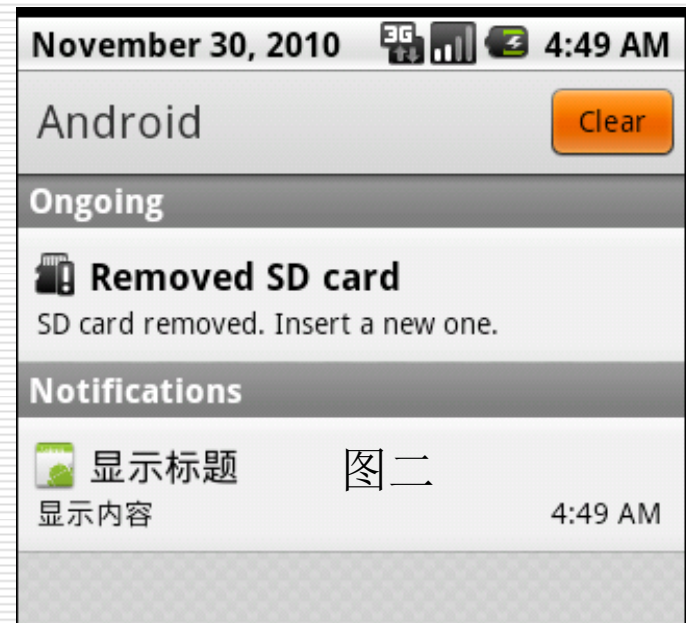
SUN YAT-SEN UNIVERSITY

□ 如果在状态栏显示提示信息? ----Notification

默认样式包含鲜明含义的图标，并且伴随着一段文字，通常还有一条扩展信息（通过下拉任务栏可以看到），这条任务栏可以与用户交互，通过点击它可以触发特定事件（例如启动Activity）



图一



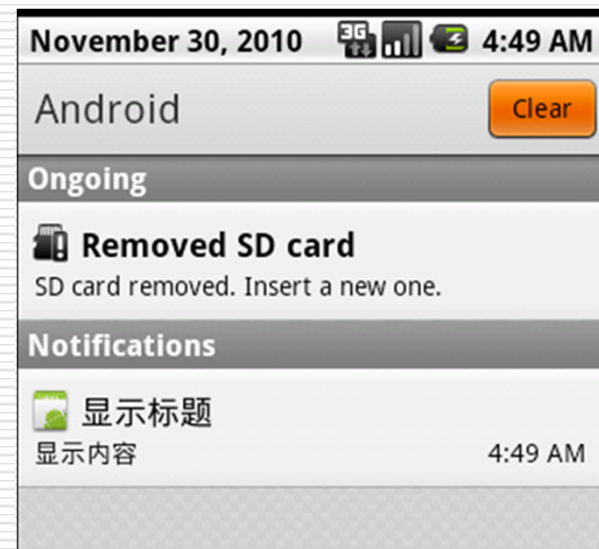
图二



新短信提示----SmsReceiver.java

SUN YAT-SEN UNIVERSITY

- ❑ 点击notification，跳转到intent中指定的activity。
- ❑ 若在目标activity显示短信内容等信息，则需要解析短信时将短信内容存入bundle，再赋给intent，并在目标activity中解析bundle。





新短信提示----SmsReceiver.java

SUN YAT-SEN UNIVERSITY

□ 如何在状态栏显示提示信息? ----Notification

//获取NotificationManager, 实例化Notification对象, 该对象代表一条具体的状态栏消息;

```
nm = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
```

//添加Notification详细信息, 包括点击后产生的intent

```
Intent intent = new Intent(MainActivity.this, SmsReceiveActivity.class);
```

```
PendingIntent contentIntent = PendingIntent.getActivity(MainActivity.this, 0, intent, 0);
```

```
Notification noti = new Notification.Builder(  
    getApplicationContext()  
    .setTicker("有短信来啦!")  
    .setContentTitle("短信")  
    .setContentText("这里放着短信内容")  
    .setSmallIcon(R.drawable.statusbar_icon)  
    .setContentIntent(contentIntent).build();
```

//使用NotificationManager进行发布

```
nm.notify(1, noti);
```



新短信提示----SmsReceiverActivity.java

SUN YAT-SEN UNIVERSITY

- 在该activity中，采用两个TextView，一个用于显示发信人，一个用于显示短信内容。
- 在onCreate中，通过代码

```
Intent from = this getIntent();  
Bundle fromBundle = from.getExtras();
```

获取从Notification传过来的数据，进行解析，然后显示。



新短信提示-----AndroidManifest.xml

SUN YAT-SEN UNIVERSITY

- 由于程序使用到系统的收短信功能，因此需要在配置文件中，添加用户使用权限。

```
<uses-permission android:name="android.permission.RECEIVE_SMS"></uses-permission>
```



调试程序

SUN YAT-SEN UNIVERSITY

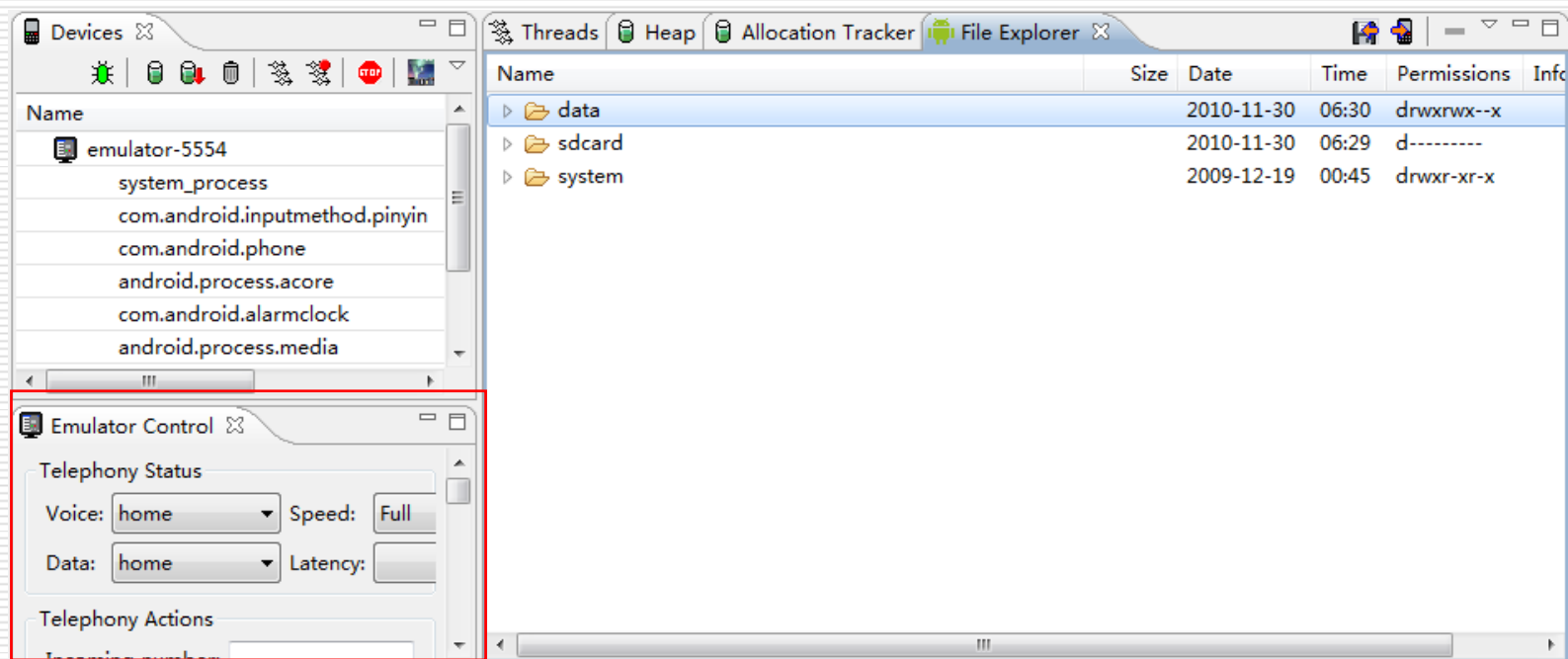
- 程序写完之后，就可以将apk装到手机上，测试下收短信的时候程序是否运行正常。
 - 但是.....没有android手机怎么办？或者别人没话费发短信给你调试程序怎么办？
 - 不怕，有DDMS.....
-



通过DDMS往模拟器发送短信（1）

SUN YAT-SEN UNIVERSITY

- DDMS 的全称是Dalvik Debug Monitor Service，它为我们提供例如：为测试设备截屏，针对特定的进程查看正在运行的线程以及堆信息、Logcat、广播状态信息、模拟电话呼叫、接收SMS、虚拟地理坐标等等。

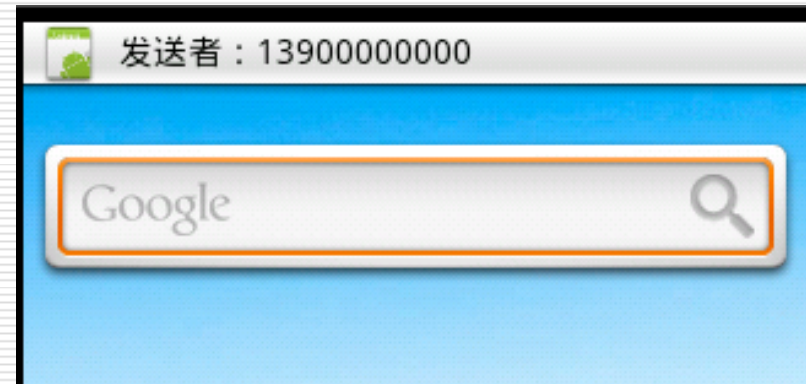
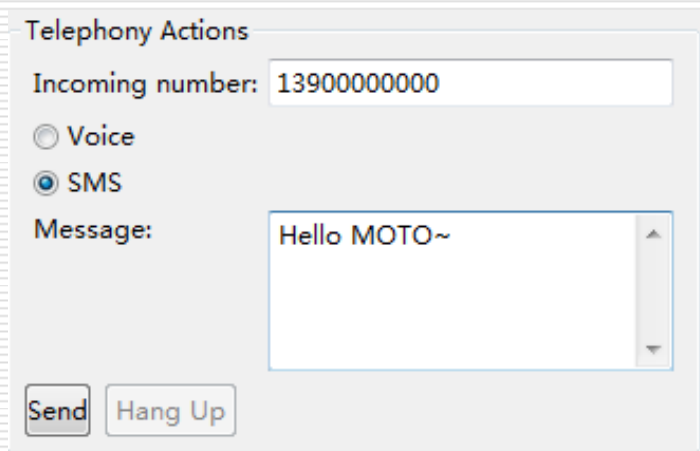




通过DDMS往模拟器发送短信（2）

SUN YAT-SEN UNIVERSITY

- 可以通过Emulator Control中的Telephony Actions往模拟器发送短信





如何注销Receiver

SUN YAT-SEN UNIVERSITY

- 当手机收到短信的时候Receiver就实例化，执行完onReceive函数之后Receiver对象就被注销。
 - 那我们如何注销这个Receiver呢？
 - 很遗憾，这个Receiver无法注销(Why?)
 - 唯一地关闭方法，就是进入手机设置，把我们安装的这个apk给删除掉
-



如何注销Receiver

SUN YAT-SEN UNIVERSITY

- 在AndroidManifest.xml注册Receiver，**是一种静态注册**，所以无法注销
 - **动态注册**：用户进入程序，按下注册按钮，注册一个Receiver，短信一来，接收并处理广播。如果用户想注销，则按下注销按钮，就可以把Receiver注销，手机收到短信再也不会自动接收广播，除非用户重新注册。
-



动态注册/注销Receiver

SUN YAT-SEN UNIVERSITY

- ❑ 动态地在代码中先定义并设置好一个 `IntentFilter`对象，然后在需要注册的地方调`Context.registerReceiver()`方法，如果取消时就调用`Context.unregisterReceiver()`方法。
 - ❑ 如果用动态方式注册`BroadcastReceiver`的`Context`对象被销毁时，`BroadcastReceiver`也就自动取消注册了。
-



修改前面例子程序 (1)

SUN YAT-SEN UNIVERSITY

- 新建一个activity的子类MainActivity.java，作为程序的入口。
(配置文件中添加intent-filter)
 - 删除配置文件中原先注册的Receiver
 - 为MainActivity新建一个布局文件，摆上两个按钮。
 - 在java文件中添加按钮点击时的处理代码。
-



修改前面例子程序 (2)

SUN YAT-SEN UNIVERSITY

```
mBtnRegister.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        BroadcastReceiver br = new SmsReceiver();
        IntentFilter filter = new IntentFilter();
        filter.addAction(SmsReceiver.mAction);
        registerReceiver(br, filter);
    });

mBtnUnRegister.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        BroadcastReceiver br = new SmsReceiver();
        unregisterReceiver(br);
    });

});
```



更多的手机自动服务（1）

SUN YAT-SEN UNIVERSITY

- `android.intent.action.BATTERY_CHANGED` 当电量改变时会触发的广播
 - `android.intent.action.TIME_TICK` 每分钟时间跳一次会触发的广播（注意，该广播的Receiver无法在配置文件中注册，只能在代码中动态注册）
-



更多的手机自动服务（2）

SUN YAT-SEN UNIVERSITY

- ❑ `android.intent.action.CAMERA_BUTTON`按下相机键会触发的广播（注意，该广播只在home screen下才会触发，且必须没有相机程序运行）
 - ❑ 还有例如蓝牙设备的状态改变、低电量、运营商改变等等广播，这些广播的使用主要靠经验的积累。
-

Questions?



ANDROID