第5章　后台服务与系统服务调用

2024年5月

5.1后台服务Service

- Android系统的Service是一种类似于Activity的组件，但Service没有用户操作界面，也不能自己启动，其主要作用是提供后台服务调用。Service不像Activity那样，当用户关闭应用界面就停止运行，Service会一直在后台运行，除非另有明确命令其停止。
- 通常使用Service为应用程序提供一些只需在后台运行的服务，或不需要界面的功能，例如，从Internet下载文件、控制Video播放器等。

Service的生命周期中只有三个阶段：onCreate, onStartCommand, onDestroy。

方 法	说 明
onCreate()	创建后台服务。
onStartCommand (Intent intent, int flags, int startId)	启动一个后台服务。
onDestroy()	销毁后台服务，并删除所有调用。
sendBroadcast(Intent intent)	继承父类Context的sendBroadcast（）方法，实现发送广播机制的消息。
onBind(Intent intent)	与服务通信的信道进行绑定，服务程序必须实现该方法。
onUnbind(Intent intent)	撤销与服务信道的绑定。

- 通常Service要在一个Activity中启动，调用Activity的startService(Intent)方法启动Service。
- 若要停止正在运行的Service，则调用Activity的stopService(Intent)方法关闭Service。
- 方法startService()和stopService()均继承于Activity及Service共同的父类android.content.Context。

- 一个服务只能创建一次，销毁一次，但可以开始多次，即onCreate()和onDestroy()方法只会被调用一次，而onStartCommand()方法可以被调用多次。后台服务的具体操作一般应该放在onStartCommand()方法里面。如果Service已经启动，当再次启动Service时则不调用onCreate()而直接调用onStartCommand()。

设计一个后台服务的应用程序步骤：

（1）创建Service的子类：

- 编写onCreate()方法，创建后台服务；
- 编写onStartCommand()方法，启动后台服务；
- 编写onDestroy()方法，终止后台服务，并删除所有调用。

(2) 创建启动和控制Service的Activity：

- 创建Intent对象，建立Activity与Service的关联；
- 调用Activity的startService(Intent)方法启动Service后台服务；
- 调用Activity的stopService(Intent)方法关闭Service后台服务。

(3) 修改配置文件AndroidManifest.xml：

- 在配置文件AndroidManifest.xml的<application>标签中添加如下代码：
- <service android:enabled="true" android:name=".AudioSrv" />

【例5-1】一个简单的后台音乐服务程序示例。

- 本例通过一个按钮启动后台服务，在服务程序中播放音乐文件，演示服务程序的创建、启动，再通过另一按钮演示服务程序的销毁过程。新建项目ex5-1后，将一个音频文件mtest1.mp3复制到应用程序的资源res/raw目录下。

百年未有之大变局，你我当负重前行!

2024年5月14日，美国政府宣布，在原有对华301关税的基础上，将进一步提高一系列中国进口商品的关税，包括电动汽车、锂电池、光伏电池、关键矿产、半导体，以及钢铝、港口起重机和个人防护装备等产品。此举预计将影响价值高达180亿美元的中国进口商品。

大家，继续努力，直面挑战吧!

商品	原关税	新关税
钢铁和铝	0-7.5%	25%
半导体	25%	50%
电动汽车	25%	100%
锂离子电池和电池零部件	7.50%	25%
关键矿物	0%	25%
太阳能电池	25%	50%
港口起重机	0%	25%
医疗产品：注射器和针头	0%	50%
医疗产品：呼吸器和口罩	0-7.5%	25%
医疗产品：BER 医疗和外科手套	7.50%	25%

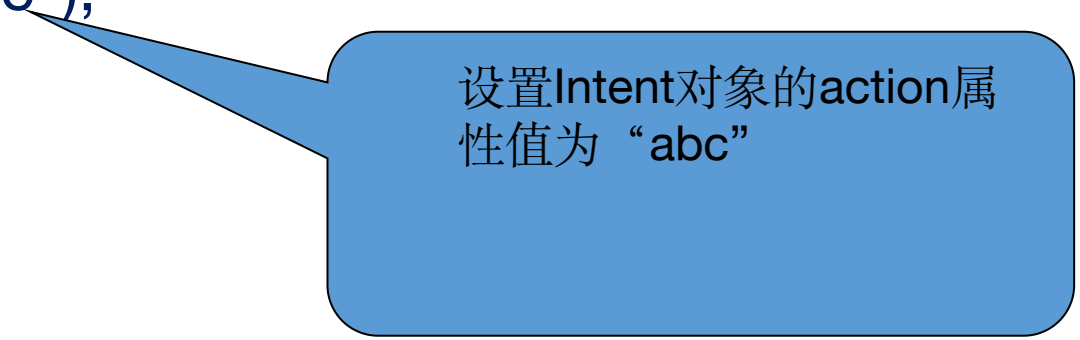
5.2信息广播机制Broadcast

- Broadcast是Android系统应用程序之间传递信息的一种机制。当系统之间需要传递某些信息时，不是通过诸如单击按钮之类组件来触发事件，而是由系统自身通过系统调用来引发事件。这种系统调用是由BroadcastReceiver类的实现的，把这种系统调用称为广播。
- BroadcastReceiver也就是“广播接收者”的意思，顾名思义，它就是用来接收来自系统和应用中的广播信息。

实现广播和接收机制的步骤：

- （1）创建Intent对象，设置Intent对象的action属性。这个action属性是接收广播数据的标识。注册了相同action属性的广播接收器才能收到发送的广播数据。

```
Intent intent = new Intent();
intent.setAction("abc");
```



- （2）编写需要广播的信息内容，将需要播发的信息封装到Intent中，通过Activity或Service继承其父类Context的sendBroadcast（）方法将Intent广播出去

键值对方式封装广播信息内容

```
intent.putExtra("hello", "这是广播信息!");
    sendBroadcast(intent);
```

- （3）编写一个继承BroadcastReceiver的子类作为广播接收器，该对象是接收广播信息并对信息进行处理的组件。在子类中要重写接收广播信息的onReceive()方法。

```
class TestReceiver extends  BroadcastReceiver
 {
    public void onReceive(Context context, Intent intent)
    {
        /*   接收广播信息并对信息作出响应的代码   */
     }
}
```

(4) 在配置文件AndroidManifest.xml中注册广播接收类。

注册广播接收类

```
<service  android:name=".TestReceiver">
    <intent-filter>
      <action android:name="abc" />
    </intent-filter>
 </service>
```

action属性值相同才能接收到广播数据

(5) 销毁广播接收器。

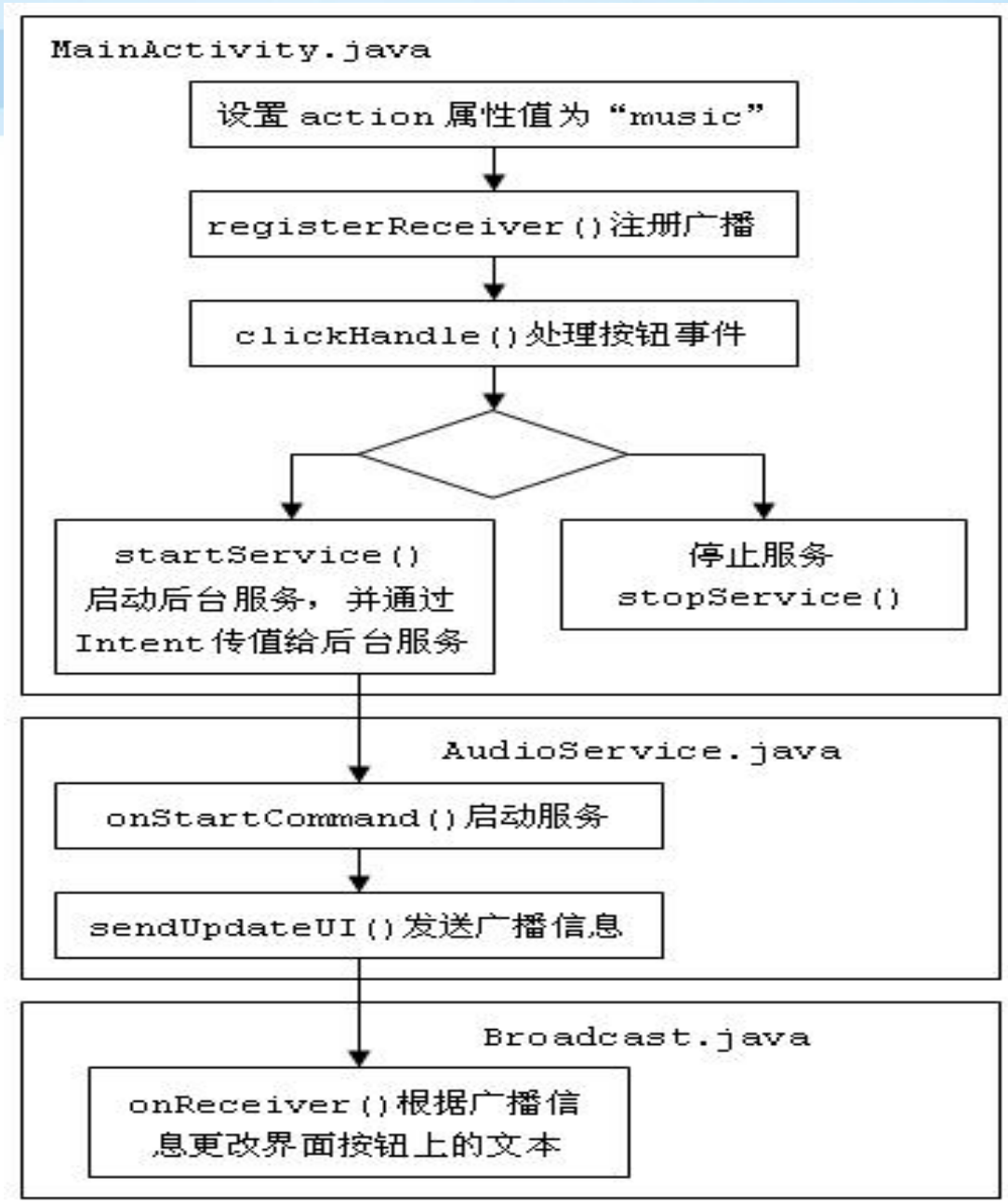
- Android系统在执行onReceive（）方法时，会启动一个程序计时器，在一定时间内，广播接收器的实例会被销毁。因此，广播机制不适合传递大数据量的信息。

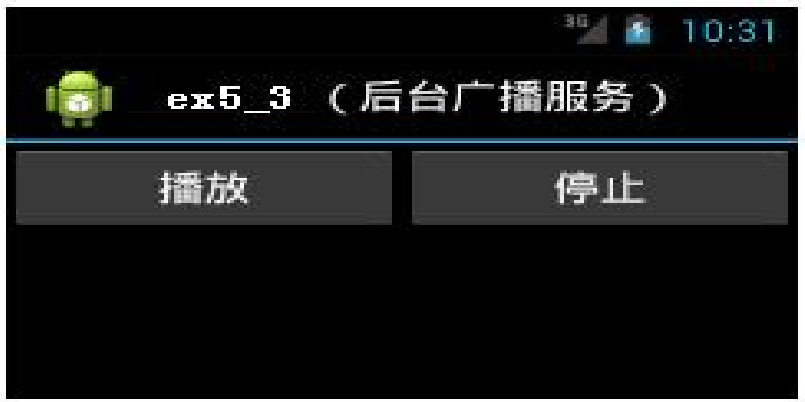
【例5-2】一个简单的信息广播程序示例。

- 为了识别Intent对象的action，有时在IntentFilter对象中设置Intent对象的action，而注册广播接收器的工作由registerReceiver()方法完成。
- registerReceiver(mBroadcast, filter)方法有两个参数，其中参数mBroadcast是广播接收器BroadcastReceiver对象，filter是IntentFilter对象。

【例5-3】由一个后台服务广播音乐的播放或暂停信息，接收器接收到信息后，执行改变用户界面按钮上文本的操作。

- 在本例中，创建了3个类：MainActivity、AudioService和Broadcast，MainActivity负责用户的交互界面，并启动后台服务；AudioService是Service的子类，在后台提供播放音乐或暂停、停止音乐等工作，同时发送改变交互界面的广播信息；Broadcast是BroadcastReceiver的子类，负责接收广播信息，更改交互界面。


MainActivity.java
设置 action 属性值为“music”
registerReceiver()注册广播
clickHandle()处理按钮事件
startService()
启动后台服务，并通过
Intent 传值给后台服务
停止服务
stopService()
AudioService.java
onStartCommand()启动服务
sendUpdateUI()发送广播信息
Broadcast.java
onReceiver()根据广播信
息更改界面按钮上的文本


10:31
ex5_3 （后台广播服务）
播放
停止

5.3 系统服务

5.3.1 Android的系统服务

系统服务	作用
WINDOW_SERVICE ("window")	窗体管理服务
LAYOUT_INFLATER_SERVICE ("layout_inflater")	布局管理服务
ACTIVITY_SERVICE ("activity")	Activity管理服务
POWER_SERVICE ("power")	电源管理服务
ALARM_SERVICE ("alarm")	时钟管理服务
NOTIFICATION_SERVICE ("notification")	通知管理服务
KEYGUARD_SERVICE ("keyguard")	键盘锁服务
LOCATION_SERVICE ("location")	基于地图的位置服务
SEARCH_SERVICE ("search")	搜索服务
VIBRATOR_SERVICE ("vibrator")	振动管理服务
CONNECTIVITY_SERVICE ("connection")	网络连接服务
WIFI_SERVICE ("wifi")	Wi-Fi连接服务
INPUT_METHOD_SERVICE ("input_method")	输入法管理服务
TELEPHONY_SERVICE ("telephony")	电话服务
DOWNLOAD_SERVICE ("download")	HTTP协议的下载服务

5.3.2 系统通知服务Notification

- Notification是Android系统的一种通知服务，当手机来电、来短信、闹钟铃声时，在状态栏显示通知的图标和文字，提示用户处理。当拖动状态栏时，可以查看这些信息。
- Notification提供了声音、振动等属性。

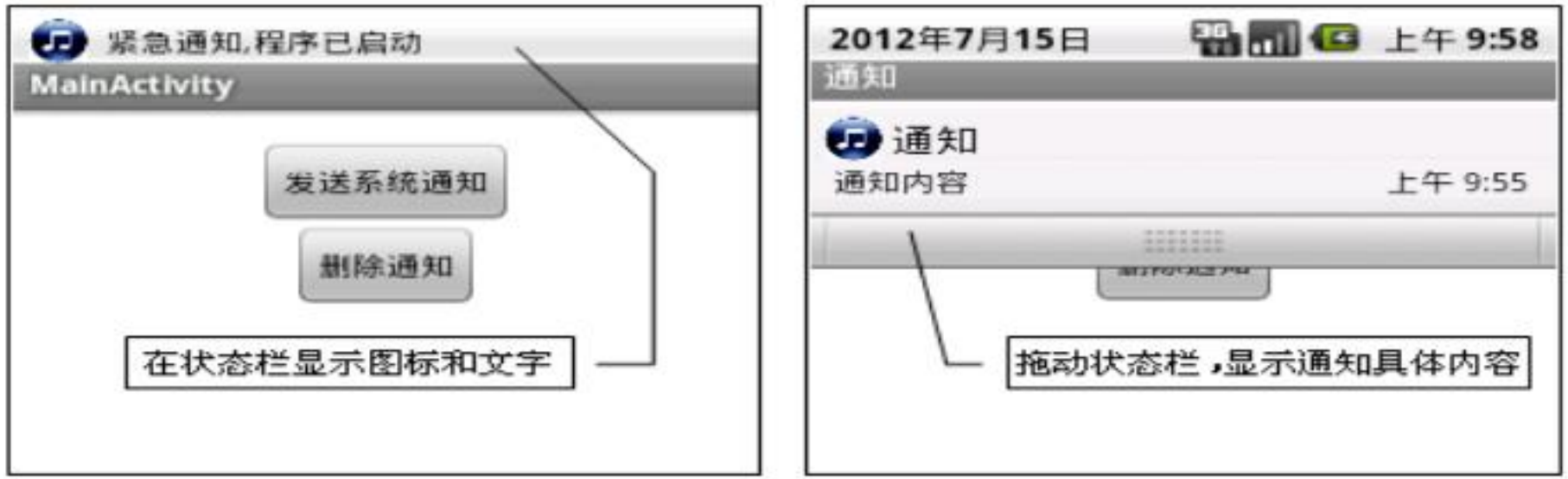
属 性	说 明
audioStreamType	所用的音频流的类型
contentIntent	设置单击通知条目所执行的Intent
contentView	设置状态栏显示通知的视图
defaults	设置成默认值
deleteIntent	删除通知所执行的Intent
icon	设置状态栏上显示的图标
iconLevel	设置状态栏上显示图标的级别
ledARGB	设置LED灯的颜色
ledOffMS	设置关闭LED时的闪烁时间（以毫秒计算）
ledOnMS	设置开启LED时的闪烁时间（以毫秒计算）
sound	设置通知的声音文件
tickerText	设置状态栏上显示的通知内容
vibrate	设置振动模式
when	设置通知发生的时间

系统通知服务Notification由系统通知管理对象NotificationManager进行管理及发布通知。由getSystemService（NOTIFICATION_SERVICE）创建NotificationManager对象，

```
NotificationManager n_Manager =
NotificationManager)getSystemService(NOTIFICATION_SERVICE);
```

NotificationManager对象通过notify(int id, Notification notification) 方法把通知发送到状态栏。通过cancelAll() 方法取消以前显示的所有通知。

【例5-4】在状态栏显示系统通知服务的应用示例。



5.3.3 系统定时服务AlarmManager

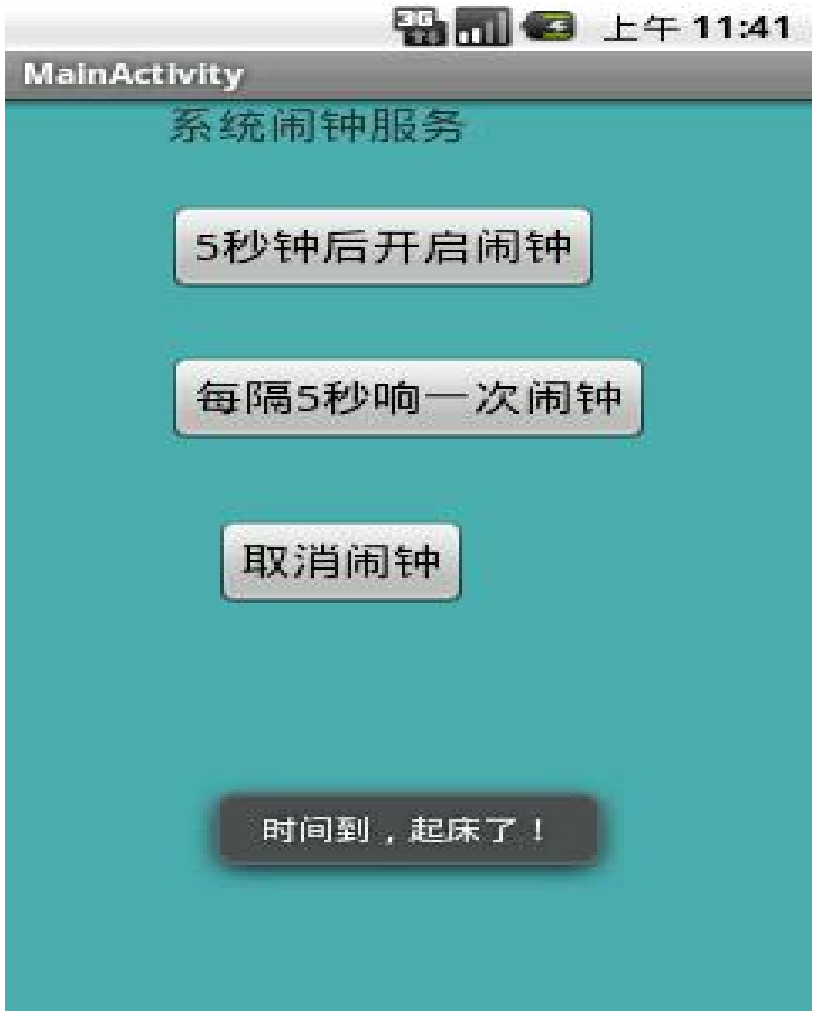
属性或方法名称	说 明
ELAPSED_REALTIME	设置闹钟时间，从系统启动开始
ELAPSED_REALTIME_WAKEUP	设置闹钟时间，从系统启动开始，如火设备休眠则唤醒
INTERVAL_DAY	设置闹钟时间，间隔一天
INTERVAL_FIFTEEN_MINUTES	间隔15分钟
INTERVAL_HALF_DAY	间隔半天
INTERVAL_HALF_HOUR	间隔半小时
INTERVAL_HOUR	间隔1小时

(续表)

RTC	设置闹钟时间，从系统当前时间开始(System.currentTimeMillis())
RTC_WAKEUP	设置闹钟时间，从系统当前时间开始，设备休眠则唤醒
set(int type,long tiggerAtTime, PendingIntent operation)	设置在某个时间执行闹钟
setRepeating(int type,long triggerAtTiem, long interval,PendingIntent operation)	设置在某个时间重复执行闹钟
setInexactRepeating(int type,long triggerAtTiem,long interval,PendingIntent operation)	是指在某个时间重复执行闹钟，但不是间隔固定时间
cancel(PendingIntent)	取消闹钟

- AlarmManager服务主要有2种应用：
  - 在指定时长后执行某项操作；
  - 周期性的执行某项操作。

【例5-5】AlarmManager时钟服务示例。

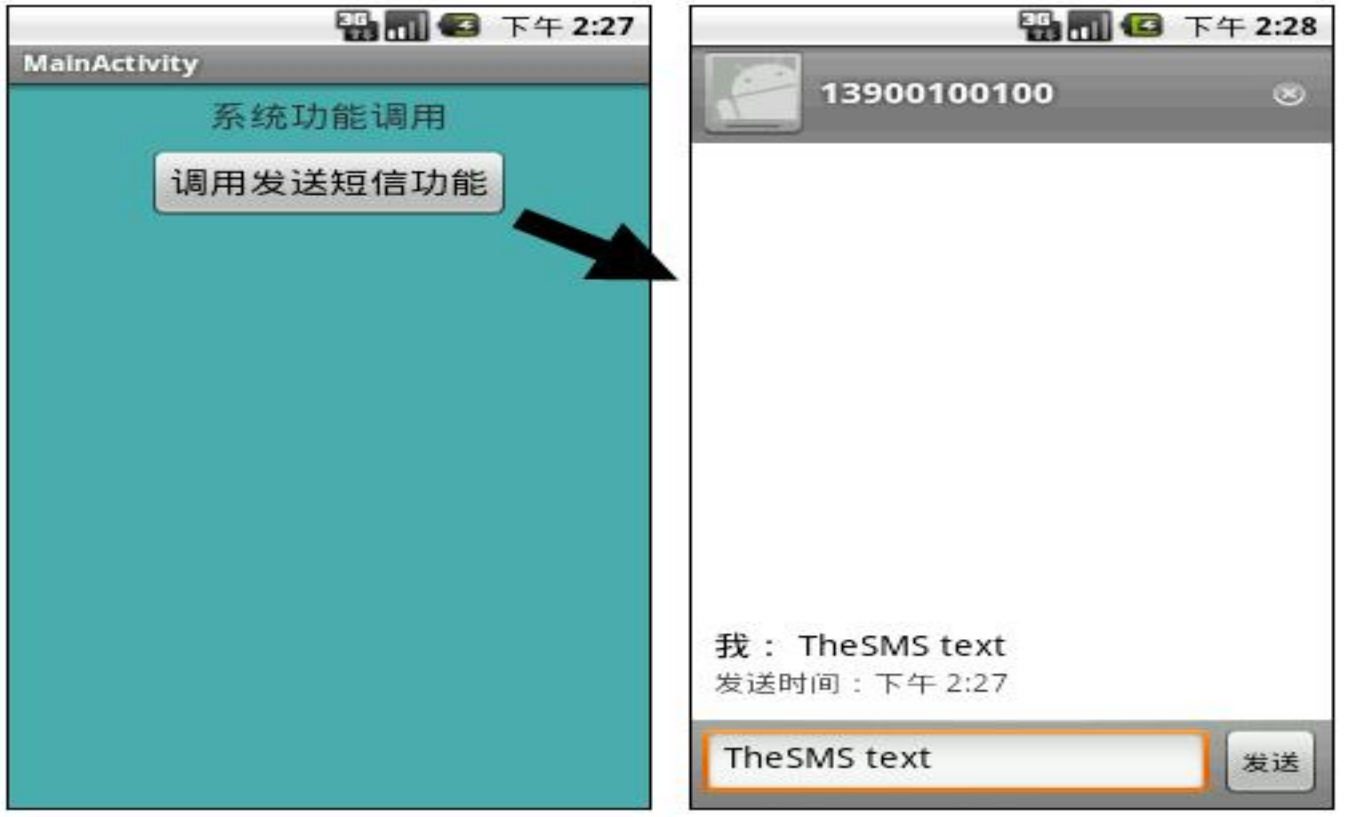


5.3.4 系统功能调用

- android 系统通过intent的action属性可以调用系统功能。
- （常用的系统功能及调用语句详见教材表5-5。）

【例5-6】调用系统的短信发送功能示例。

- 本示例仅设置一个按钮，在按钮的事件中，添加发送短信的代码。



习题五

1、结合例5-1和例5-3，编写一个具有较完善功能的后台音乐播放器。

2、编写一个短信服务平台。

Q&A

谢谢大家！