**西 安 邮 电 大 学**

**（计算机学院）**

《Android应用开发》课内实验报告

# 实验四：多线程应用

**专业名称： 计算机科学与技术**

**班 级： 计科1608**

**学 号： 05168066**

**学生姓名： 祁泽宇**

**指导教师： 张庆生**

**起止时间： 2019年5月13日**

1. 课程设计目的
2. 掌握首选项方式的存储和读取
3. 掌握位于内存上的数据文件的存储和读取
4. 掌握位于SD卡上的数据文件的存储和读取
5. 实验环境

JDK:1\_8\_40

IDE:Android Studio

1. 实验内容

实现基于文件的日程安排应用，运行结果，通过单击增加图标打开新增活动界面，通过单击保存按钮将活动内容写入到文件，并返回首页。

1. 实验过程及分析
2. 修改主布局代码如下：

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="0dp"  
 android:layout\_marginTop="2dp"  
 android:gravity="center"  
 android:text="通讯录"  
 android:textSize="25dp"  
 android:textStyle="bold" />  
  
 <ListView  
 android:id="@+id/listview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="629dp"  
  
  
 android:layout\_alignTop="@+id/title"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginStart="0dp"  
 android:layout\_marginTop="97dp"  
 android:layout\_marginBottom="3dp"></ListView>  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="63dp"  
 android:onClick="add"  
 android:layout\_below="@+id/title"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginStart="0dp"  
 android:layout\_marginTop="1dp"  
 android:text="新建" />  
  
</RelativeLayout>

1. 修改主Activity文件，通过getItemList()方法从item中读取数据显示到Listview组件中，单击新增打开增加活动界面，新增活动界面关闭时返回onActivityResult()方法中再次调用getItemList()方法获取最新数据并显示，代码如下。

package com.example.class5;  
  
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
  
import java.io.BufferedReader;  
import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.util.ArrayList;  
import java.util.List;  
  
public class MainActivity extends AppCompatActivity {  
 private ListView itemlist;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 itemlist = (ListView) findViewById(R.id.listview);  
 getItemList();  
 }  
 //该方法是其他子Activity回到主Activity时，有可能时  
 // 返回一些子模块完成的数据交给主Activity处理。  
 // 这样的数据交流就要用到回调函数onActivityResult()。  
// requestCode 请求码，即调用startActivityForResult()传递过去的值  
// resultCode 结果码，结果码用于标识返回数据来自哪个新Activity  
 @Override  
 protected void onActivityResult(int requestCode,int resultCode,Intent data){  
 super.onActivityResult(requestCode,resultCode,data);  
 if(requestCode==0x111 && resultCode==0x111)  
 {  
 getItemList();  
 }  
 }  
  
 public void add(View v)  
 {  
 Intent intent = new Intent(MainActivity.this,AddItemActivity.class);  
 startActivityForResult(intent,0x111);//打开新界面，0x111是请求码，  
}  
 private void getItemList(){  
 ArrayList<String> list = new ArrayList<String>();  
 try{  
 FileInputStream in = openFileInput("item");  
 BufferedReader br = new BufferedReader(new InputStreamReader(in));  
 String line ="";  
 while((line=br.readLine())!=null)  
 {  
 list.add(line);  
 }  
 br.close();  
 in.close();  
 String[] contents = {};  
 String[] allItem = list.toArray(contents); //将动态数组转化成字符串数组  
 ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,android.R.layout.simple\_list\_item\_1,allItem);  
 itemlist.setAdapter(adapter);  
  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
  
 }  
 }  
}

1. 创建增加活动界面布局文件additem.xml,代码如下

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:baselineAligned="false"  
 android:text="新增活动"  
 android:textSize="20dp"  
 android:textStyle="bold"/>  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="日期"  
 android:textSize="20dp"  
 android:textStyle="bold"/>  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/date"  
 android:onClick="chooseDate"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 />  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="开始时间"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 />  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/startTime"  
 android:onClick="chooseStartTime"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 />  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="结束时间"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 />  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/endTime"  
 android:onClick="chooseEndTime"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 />  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="活动说明"  
 android:textSize="20dp"  
 android:textStyle="bold"/>  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/item"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 />  
 <Button android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/save"  
 android:onClick="save"  
 android:text="保存"  
 android:textSize="20dp"  
 android:textStyle="bold"/>  
  
</LinearLayout>

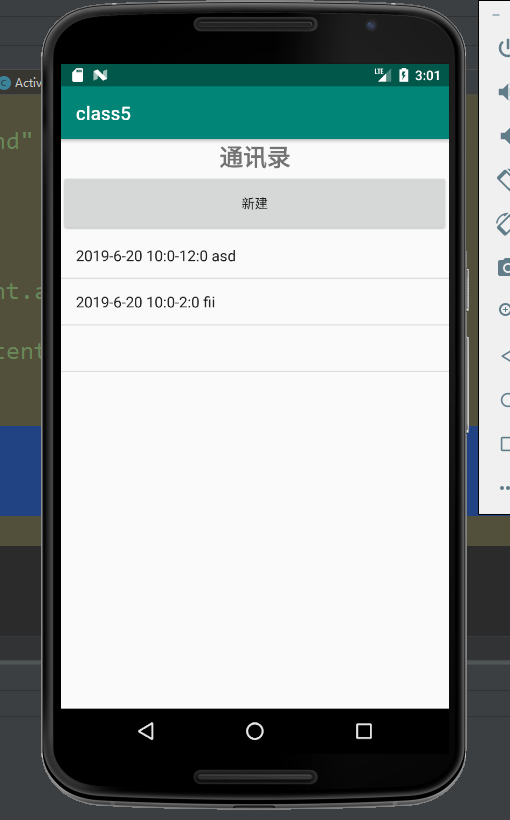
1. 创建增加活动界面Activity文件AdditemActivity.java，输入新增活动的日期，开始时间、结束时间、保存文件到item中，代码如下

package com.example.class5;  
  
import android.app.DatePickerDialog;  
import android.app.TimePickerDialog;  
import android.content.Intent;  
import android.os.Bundle;  
import android.support.v7.app.AppCompatActivity;  
import android.view.View;  
import android.widget.DatePicker;  
import android.widget.EditText;  
import android.widget.TimePicker;  
import android.widget.Toast;  
  
import java.io.FileNotFoundException;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.io.PrintStream;  
  
public class AddItemActivity extends AppCompatActivity {  
 private EditText date,startTime,endTime,item;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_additem);  
 date = (EditText) findViewById(R.id.date);  
 startTime = (EditText) findViewById(R.id.startTime);  
 endTime = (EditText) findViewById(R.id.endTime);  
 item = (EditText) findViewById(R.id.item);  
  
 }  
 //通过日期对话框选择日期  
 public void chooseDate(View view){  
 DatePickerDialog datepd = new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() {  
 @Override  
 public void onDateSet(DatePicker dp, int year, int month, int day) {  
  
 date.setText(year+"-"+(month+1)+"-"+day); //写入到日期框  
 }  
 },2019,5,20); //默认时间  
 datepd.setMessage("请选择日期");  
 datepd.show();  
  
 }  
 //通过时间对话框选择开始时间  
 public void chooseStartTime(View view){  
 TimePickerDialog timepd = new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {  
 @Override  
 public void onTimeSet(TimePicker view, int hourOfDay, int minute) {  
 String t = "0";  
 t = t+Integer.toString(minute);  
 if(minute<10)  
 startTime.setText(hourOfDay+":"+t);  
 else  
  
 startTime.setText(hourOfDay+":"+minute);  
 }  
 },10,0,true);//设置时间对话框，并确定默认时间  
 timepd.setMessage("请选择开始时间");  
 timepd.show();  
 }  
 //通过时间对话框选择结束时间  
 public void chooseEndTime(View view){  
 TimePickerDialog timepd = new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {  
 @Override  
 public void onTimeSet(TimePicker view, int hourOfDay, int minute) {  
 String t = "0";  
 t = t+Integer.toString(minute);  
 if(minute<10)  
 endTime.setText(hourOfDay+":"+t);  
 else  
  
 endTime.setText(hourOfDay+":"+minute);  
  
 }  
 },12,0,true);  
 timepd.setMessage("请选择结束时间");  
 timepd.show();  
 }  
 //保存活动消息  
 public void save(View view){  
 StringBuilder result = new StringBuilder();  
 result.append(date.getText().toString()+" ");  
 result.append(startTime.getText().toString()+"-");  
 result.append(endTime.getText().toString()+" ");  
 result.append(item.getText().toString());  
 try{  
 //以追加的方式保存信息到item文件中  
 FileOutputStream out = openFileOutput("item",MODE\_APPEND);  
 PrintStream ps = new PrintStream(out);  
 ps.println(result.toString());  
 ps.close();  
 out.close();  
 Toast.makeText(this,"保存完毕！",Toast.LENGTH\_LONG).show();  
 Intent intent = getIntent();  
 setResult(0x111,intent);//设置结果，返回值，用于MainActivity的onActivityResult接收  
 finish(); //关闭当前页面，返回前页  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
  
}

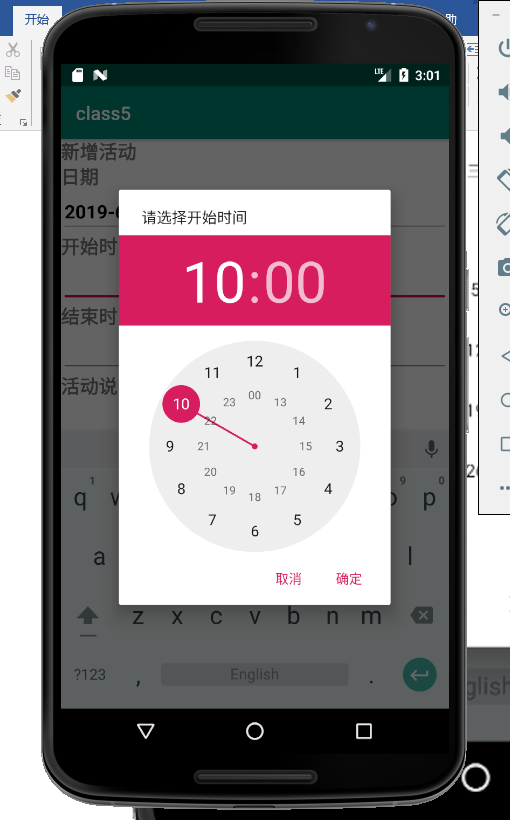
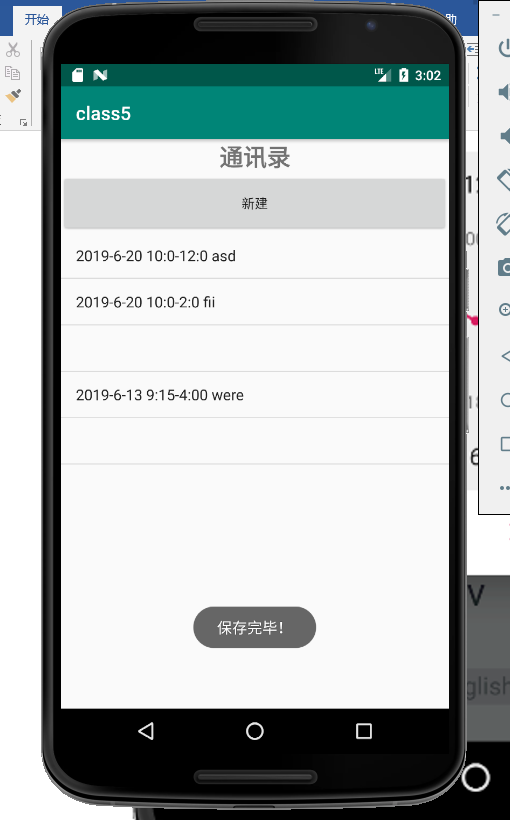
1. 在AndroidManifest.xml文件中添加新增活动界面的注册，代码如下

<activity android:name=".AddItemActivity"  
 android:label="@string/app\_name">  
  
</activity>

1. 实验结果

图一 主界面 图二 选择日期

图三 选择开始时间 图四 保存结果

1. 总结

本次实验我们用到了Acitivity中的其他方法onActivityResult(),该方法是其他子Activity回到主Activity时，有可能返回一些子模块完成的数据交给主Acticity处理。这样的数据交流就要使用该回调函数 requestCode是请求码，即调用startActivityForResult()传递过去的值，resultCode是结果码，结果码用于标识返回数据来自哪一个Activity。通过这种方式，将不同页面之间结合起来，来达到目的功能。这次实验我还用到了几个小组件DataPickerDialog与TimePickerDialog,使用这两个组件可以简便的设置时间和日期，十分的方便。这次因为只有一个子Activity与主Activity交流，如果交流的页面变多，那么就需要设计者的设计思想十分清晰才行，这样才能设计出功能完成的程序。