**文件存储**

**使用文件系统**

Android使用的是基于Linux的文件系统，开发人员可以建立和访问程序自身的私有文件，也可以访问保存在资源目录中的原始文件和XML文件，还可以在SD卡等外部存储设备中保存文件

Android允许应用程序创建仅能够自身访问的私有文件，文件保存在设备的内部存储器上，在Linux系统上的/data/data/<package\_name>/中

使用SDCard

（1）文件操作

常用FileInputStream，FileOutputStream类结合openFileOutput，openFileInput方法完成对文件的读写

（2）使用文件资源

原始资源文件位于res/raw目录下，程序部署时会将其完整地打包进apk，在程序中可通过资源ID访问

（3）访问SD卡

判断SD卡是否存在

Environment.getExternalStorageState()

Environment.MEDIA\_MOUNTED则说明存在即可读写

获得SD卡目录

Environment.getExternalStorageDirectory()

错误整理：

（1）读取SD卡以及写入SD卡时，即使在AndroidManifest.xml声明权限，仍然抛出读写异常：

W/System.err:java.io.FileNotFoundException:/storage/emulated/0/test.txt(Permissiondenied)

这是由于在Android 6.0中，新增了有些权限属于Protected Permission，这类权限的获取不仅仅要在AndroidManifest.xml中获取到，而且还需要在我们的代码进行动态获取，在app第一次运行读写SD卡操作时，会触发权限许可弹出对话框，用户点击同意/允许之后，才能执行读写SD卡的操作。

在执行读写SD卡的代码块前，添加如下代码，即可解决。

**if** (Build.VERSION.***SDK\_INT*** >= 23) {  
 **int** REQUEST\_CODE\_CONTACT = 101;  
 String[] permissions = {Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***};  
 *//验证是否许可权限* **for** (String str : permissions) {  
 **if** (**this**.checkSelfPermission(str) != PackageManager.***PERMISSION\_GRANTED***) {  
 *//申请权限* **this**.requestPermissions(permissions, REQUEST\_CODE\_CONTACT);  
 **return**;  
 }  
 }  
}

AndroidManifest.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 package="com.hjs.sharedpreferencedemo.sharedpreference"**>  
  
  
 *<!--用户写入外部存储设备SD卡的权限-->* <**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"**/>  
  
 *<!--读取SD卡数据的权限-->* <**uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"** />  
  
 *<!--在SD卡中创建与删除文件权限-->* <**uses-permission android:name="android.permission.MOUNT\_FORMAT\_FILESYSTEMS"**/>  
  
  
 *<!-- <uses-sdk  
 android:maxSdkVersion="23"  
 android:minSdkVersion="16"/>-->* <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
 <**activity android:name=".MainActivity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 </**application**>  
  
</**manifest**>

activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**Button  
 android:id="@+id/btnWrite"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="36dp"  
 android:layout\_marginLeft="36dp"  
 android:layout\_marginTop="206dp"  
 android:onClick="write"  
 android:text="写入"** />  
  
 <**EditText  
 android:id="@+id/etContent"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignLeft="@+id/btnWrite"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginLeft="-2dp"  
 android:layout\_marginTop="41dp"  
 android:ems="10"** />  
  
 <**Button  
 android:id="@+id/btnRead"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignTop="@+id/btnWrite"  
 android:layout\_centerHorizontal="true"  
 android:text="读出"  
 android:onClick="read"**/>  
  
 <**Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignLeft="@+id/btnWrite"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginLeft="-1dp"  
 android:layout\_marginTop="291dp"  
 android:onClick="readRaw"  
 android:text="读出Raw中的文件"** />  
  
 <**Button  
 android:id="@+id/btnSDWrite"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignRight="@+id/btnWrite"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginBottom="76dp"  
 android:text="写入SD卡"  
 android:onClick="operatorSDCard"**/>  
  
 <**Button  
 android:id="@+id/btnSDRead"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignLeft="@+id/btnRead"  
 android:layout\_alignTop="@+id/btnSDWrite"  
 android:text="从SD卡读出"  
 android:onClick="operatorSDCard"**/>  
</**RelativeLayout**>

MainActivity.java

**package** com.hjs.sharedpreferencedemo.sharedpreference;  
  
**import** android.Manifest;  
**import** android.content.Context;  
**import** android.content.SharedPreferences;  
**import** android.content.pm.PackageManager;  
**import** android.os.Build;  
**import** android.os.Environment;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**import** java.io.BufferedReader;  
**import** java.io.BufferedWriter;  
**import** java.io.File;  
**import** java.io.FileInputStream;  
**import** java.io.FileNotFoundException;  
**import** java.io.FileOutputStream;  
**import** java.io.IOException;  
**import** java.io.InputStream;  
**import** java.io.InputStreamReader;  
**import** java.io.OutputStream;  
**import** java.io.OutputStreamWriter;  
**import** java.io.UnsupportedEncodingException;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private** SharedPreferences **mPreferences**;  
 **private** EditText **etContent**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 *//1.得到SharedPreferences对象  
 //第一个参数：文件名name；第二个参数名：进入模式mode  
 //MODE\_PRIVATE 只有当前应用程序自身可用* **mPreferences**=getSharedPreferences(**"test"**,Context.***MODE\_PRIVATE***);*//私有模式，只可以当前自己的应用程序使用* **etContent**=(EditText)findViewById(R.id.***etContent***);  
 }  
  
 */\*\*  
 \* 写入SharedPreferences文件  
 \** ***@param view*** *\*/* **public void** write(View view){  
 String content = **etContent**.getText().toString();  
 FileOutputStream fos = **null**;  
 BufferedWriter writer = **null**;  
 **try** {  
 *//参数1：文件名 参数2：私有模式（只有app自身能够访问）* fos = openFileOutput(**"test.txt"**,Context.***MODE\_PRIVATE***);  
  
 *//OutputStreamWriter可把字符流和字节流连接在一起* writer = **new** BufferedWriter(**new** OutputStreamWriter(fos));  
 writer.write(content);  
 writer.flush();*//必须刷新才能一次写入文件* Toast.*makeText*(**this**,**"写入成功!"**,Toast.***LENGTH\_LONG***).show();  
 } **catch** (FileNotFoundException e) {  
 e.printStackTrace();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }**finally** {  
 **if** (writer!=**null**){  
 **try** {  
 writer.close();*//最后开的先关* } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 **if** (fos!=**null**){  
 **try** {  
 fos.close();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }  
  
 */\*\*  
 \* 读出SharedPreferences文件  
 \** ***@param view*** *\*/* **public void** read(View view){  
 BufferedReader br = **null**;  
 StringBuilder sb = **new** StringBuilder();  
 **try** {  
 br=**new** BufferedReader(**new** InputStreamReader(openFileInput(**"test.txt"**)));  
 String content = br.readLine();  
 **while** (content!=**null**){  
 sb.append(content);*//读出的内容放在StringBuilder之间* content = br.readLine();  
 }  
 **etContent**.setText(sb.toString());  
 } **catch** (FileNotFoundException e) {  
 e.printStackTrace();  
 } **catch** (IOException e){  
 e.printStackTrace();  
 }  
 **finally** {  
 **try** {  
 **if** (br!=**null**) {  
 br.close();*//因为是包装流，所以关闭顶层流，底层流就会自动关闭* }  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
  
 */\*\*  
 \* 从资源文件res/raw 中读文件  
 \** ***@param view*** *\*/* **public void** readRaw(View view){  
 InputStream is = getResources().openRawResource(R.raw.***rawtest***);  
 **try** {  
 BufferedReader br = **new** BufferedReader(**new** InputStreamReader(is,**"utf-8"**));  
 String content = **null**;  
 StringBuilder sb = **new** StringBuilder();  
 **try** {  
 content = br.readLine();  
 **while** (content != **null**) {  
 sb.append(content);  
 content = br.readLine();  
 }  
 **etContent**.setText(sb.toString());  
 }**catch** (IOException e){  
 e.printStackTrace();  
 }**finally** {  
 **try** {  
 **if** (br!=**null**) {  
 br.close();  
 }  
 **if** (is!=**null**){  
 is.close();  
 }  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 } **catch** (UnsupportedEncodingException e) {  
 e.printStackTrace();  
 }  
 }  
  
 */\*\*  
 \* 操作SD卡  
 \** ***@param view*** *\*/* **public void** operatorSDCard(View view){  
  
 *//动态获取内存存取权限* **if** (Build.VERSION.***SDK\_INT*** >= 23) {  
 **int** REQUEST\_CODE\_CONTACT = 101;  
 String[] permissions = {Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***};  
 *//验证是否许可权限* **for** (String str : permissions) {  
 **if** (**this**.checkSelfPermission(str) != PackageManager.***PERMISSION\_GRANTED***) {  
 *//申请权限* **this**.requestPermissions(permissions, REQUEST\_CODE\_CONTACT);  
 **return**;  
 }  
 }  
 }  
  
 *//1.判断SD卡状态* String state = Environment.*getExternalStorageState*();  
 **if** (state.equals(Environment.***MEDIA\_MOUNTED***)){  
 *//SD卡是可用的  
 //Toast.makeText(this,"当前手机的SD卡可用...",Toast.LENGTH\_LONG).show();* **switch** (view.getId()){  
 **case** R.id.***btnSDWrite***:  
 *//得到路径  
 //1.得到当前外部存储设备的目录* File dir = Environment.*getExternalStorageDirectory*();  
 */\*try {  
 Toast.makeText(this,dir.getCanonicalPath(),Toast.LENGTH\_LONG).show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }\*/  
 //2.目录中创建一个写在外部存储设备的文件对象* File file = **new** File(dir,**"test.txt"**);  
 FileOutputStream fos = **null**;  
  
 **try** {  
 fos = **new** FileOutputStream(file);  
 write(fos);  
 } **catch** (FileNotFoundException e) {  
 e.printStackTrace();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }**finally** {  
 **if** (fos!=**null**){  
 **try** {  
 fos.close();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 **break**;  
 **case** R.id.***btnSDRead***:  
 *//InputStream  
 //得到路径* File dir2 = Environment.*getExternalStorageDirectory*();  
 File file2 = **new** File(dir2,**"test.txt"**);  
 FileInputStream fis = **null**;  
 **try** {  
 fis = **new** FileInputStream(file2);  
 read(fis);  
 } **catch** (FileNotFoundException e) {  
 e.printStackTrace();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }**finally** {  
 **if** (fis!=**null**) {  
 **try** {  
 fis.close();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 **break**;  
 }  
 }**else**{  
 Toast.*makeText*(**this**,**"当前手机的SD卡有问题..."**,Toast.***LENGTH\_LONG***).show();  
 }  
 }  
  
 **public void** write(OutputStream os) **throws** IOException{  
 String content = **etContent**.getText().toString();  
 BufferedWriter writer = **null**;  
 **try** {  
 writer = **new** BufferedWriter(**new** OutputStreamWriter(os));  
 writer.write(content);  
 writer.flush();  
 Toast.*makeText*(**this**, **"写入完成......"**, Toast.***LENGTH\_LONG***).show();  
 }**finally** {  
 **if** (writer!=**null**){  
 writer.close();  
 }  
 }  
 }  
  
 **public void** read(InputStream is) **throws** IOException {  
 BufferedReader br = **null**;  
 StringBuilder sb = **new** StringBuilder();  
 **try** {  
 br = **new** BufferedReader(**new** InputStreamReader(is));  
 String content = br.readLine();  
 **while** (content!=**null**){  
 sb.append(content);  
 content=br.readLine();  
 }  
 **etContent**.setText(sb.toString());  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }**finally** {  
 **if** (br!=**null**){  
 br.close();  
 }  
 }  
  
 }  
}