**【安卓】主线程与子线程的五种通信方式**

要简单滴说一下android中主线程与子线程之间进行通信的五种方式。大家一定都知道，android中的关于UI的处理都必须在主线程(UI线程)上进行，而android中的主线程如果被阻塞超过5s没有响应就会出现ANR（应用无响应），所以耗时的操作都不应该放在主线程上。因此子线程（工作线程）就正式出场了，可以把耗时操作都放在子线程中。但是这时候也会出现一个问题。就是当需要处理耗时操作时，主线程如何切换到子线程去操作，处理完耗时操作子线程又如何回到主线程上呢。下边就这个问题讨论下。

有以下五种方式：  
1.Activity.runOnUiThread(Runnable)  
2.View.post(Runnable)  
3.View.postDelayed(Runnable, long)  
4.Handler  
5.AsyncTask

我写了个小demo来简单描述下  
首先是一个主界面，有五个按钮，然后针对这5种方式简单写了个页面，点击五个按钮分别跳到这5个页面。下边上代码：  
1.BaseActivity

package com.example.cyy.interthread.base;

import android.app.ActivityManager;

import android.content.Context;

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

import android.widget.TextView;

import java.util.List;

/\*\*

\* Created by user on 2017/8/17.

\*/

public class BaseActivity extends AppCompatActivity {

public ActivityManager manager;

private StringBuilder builder;

public static final String THREAD\_TAG = "thread\_tag";

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

init();

}

private void init() {

if(manager == null) {

manager = (ActivityManager) getSystemService(Context.ACTIVITY\_SERVICE);

}

}

public void getProcessInfo(boolean isShow){

List<ActivityManager.RunningAppProcessInfo> processes = manager.getRunningAppProcesses();

if(builder == null) {

builder = new StringBuilder();

}

for (ActivityManager.RunningAppProcessInfo processInfo : processes) {

builder.append("当前进程id:")

.append(processInfo.pid)

.append("\n当前线程id:")

.append(Thread.currentThread().getId())

.append("\n当前线程状态:")

.append(Thread.currentThread().getState())

.append("\n");

}

if(isShow){

((TextView) findViewById(android.R.id.text1)).setText(builder.toString());

Log.e(THREAD\_TAG, builder.toString());

}

}

}

2.MainActivity

package com.example.cyy.interthread;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import com.example.cyy.interthread.base.BaseActivity;

public class MainActivity extends BaseActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

public void doClick(View v) {

switch (v.getId()) {

case R.id.btn\_thread\_inter\_a:

Intent intent\_a = new Intent(this, RunOnUiTestActivity.class);

startActivity(intent\_a);

break;

case R.id.btn\_thread\_inter\_b:

Intent intent\_b = new Intent(this, PostTestActivity.class);

startActivity(intent\_b);

break;

case R.id.btn\_thread\_inter\_c:

Intent intent\_c = new Intent(this, PostDelayTestActivity.class);

startActivity(intent\_c);

break;

case R.id.btn\_thread\_inter\_d:

Intent intent\_d = new Intent(this, HandlerTestActivity.class);

startActivity(intent\_d);

break;

case R.id.btn\_thread\_inter\_e:

Intent intent\_e = new Intent(this, AsyncTaskTestActivity.class);

startActivity(intent\_e);

break;

}

}

}

3.activity\_main

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/activity\_main"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

tools:context="com.example.cyy.interthread.MainActivity">

<Button

android:id="@+id/btn\_thread\_inter\_a"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/thread\_inter\_a"

android:textAllCaps="false"

android:onClick="doClick"/>

<Button

android:id="@+id/btn\_thread\_inter\_b"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/thread\_inter\_b"

android:textAllCaps="false"

android:onClick="doClick"/>

<Button

android:id="@+id/btn\_thread\_inter\_c"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/thread\_inter\_c"

android:textAllCaps="false"

android:onClick="doClick"/>

<Button

android:id="@+id/btn\_thread\_inter\_d"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/thread\_inter\_d"

android:textAllCaps="false"

android:onClick="doClick"/>

<Button

android:id="@+id/btn\_thread\_inter\_e"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/thread\_inter\_e"

android:textAllCaps="false"

android:onClick="doClick"/>

</LinearLayout>

资源Strings.xml文件

<resources>

<string name="app\_name">InterThread</string>

*<!--五种子线程与主线程之间通信方式-->*

<string name="thread\_inter\_a">RunOnUiThread实现主&amp;子线程间通信</string>

<string name="thread\_inter\_b">View.post(runnable)实现主&amp;子线程间通信</string>

<string name="thread\_inter\_c">View.post(runnable, delayTime)实现主&amp;子线程间通信</string>

<string name="thread\_inter\_d">Handler实现主&amp;子线程间通信</string>

<string name="thread\_inter\_e">AsyncTask实现主&amp;子线程间通信</string>

</resources>

接下来就是这五个类了  
1.RunOnUiTestActivity

package com.example.cyy.interthread;

import android.os.Bundle;

import android.support.annotation.Nullable;

import com.example.cyy.interthread.base.BaseActivity;

/\*\*

\* Created by user on 2017/8/17.

\*/

public class RunOnUiTestActivity extends BaseActivity{

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(android.R.layout.test\_list\_item);

*//刚开始从主线程（UI线程）开始*

getProcessInfo(false);

new Thread(new Runnable() {

@Override

public void run() {

*//子线程（工作线程），处理耗时操作*

getProcessInfo(false);

runOnUiThread(new Runnable() {

@Override

public void run() {

*//回到主线程（UI线程），处理UI*

getProcessInfo(true);

}

});

}

}).start();

}

}

2.PostTestActivity

package com.example.cyy.interthread;

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.widget.TextView;

import com.example.cyy.interthread.base.BaseActivity;

/\*\*

\* Created by user on 2017/8/17.

\*/

public class PostTestActivity extends BaseActivity{

private TextView text;

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(android.R.layout.test\_list\_item);

*//刚开始从主线程（UI线程）开始*

getProcessInfo(false);

text = (TextView) findViewById(android.R.id.text1);

new Thread(new Runnable() {

@Override

public void run() {

*//子线程（工作线程），处理耗时操作*

getProcessInfo(false);

text.post(new Runnable() {

@Override

public void run() {

*//回到主线程（UI线程），处理UI*

getProcessInfo(true);

}

});

}

}).start();

}

}

3.PostDelayTestActivity

package com.example.cyy.interthread;

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.widget.TextView;

import com.example.cyy.interthread.base.BaseActivity;

/\*\*

\* Created by user on 2017/8/17.

\*/

public class PostDelayTestActivity extends BaseActivity{

private TextView text;

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(android.R.layout.test\_list\_item);

*//刚开始从主线程（UI线程）开始*

getProcessInfo(false);

text = (TextView) findViewById(android.R.id.text1);

new Thread(new Runnable() {

@Override

public void run() {

*//子线程（工作线程），处理耗时操作*

getProcessInfo(false);

text.postDelayed(new Runnable() {

@Override

public void run() {

*//回到主线程（UI线程），处理UI*

getProcessInfo(true);

}

}, 5000);

}

}).start();

}

}

7.HandlerTestActivity

package com.example.cyy.interthread;

import android.os.Bundle;

import android.os.Handler;

import android.os.Message;

import android.support.annotation.Nullable;

import android.widget.Toast;

import com.example.cyy.interthread.base.BaseActivity;

/\*\*

\* Created by user on 2017/8/17.

\*/

public class HandlerTestActivity extends BaseActivity{

private Handler handler = new Handler(){

@Override

public void handleMessage(Message msg) {

super.handleMessage(msg);

*//回到主线程（UI线程），处理UI*

getProcessInfo(true);

switch (msg.what) {

case 0:

Toast.makeText(HandlerTestActivity.this, "收到子线程发送的消息:" + msg.obj, Toast.LENGTH\_LONG).show();

}

}

};

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(android.R.layout.test\_list\_item);

*//刚开始从主线程（UI线程）开始*

getProcessInfo(false);

new Thread(new Runnable() {

@Override

public void run() {

*//子线程（工作线程），处理耗时操作*

Message msg = new Message();

msg.what = 0;

msg.obj = "我的id是" + Thread.currentThread().getId();

handler.sendMessage(msg);

getProcessInfo(false);

}

}).start();

}

}

5.AsyncTaskTestActivity

package com.example.cyy.interthread;

import android.os.AsyncTask;

import android.os.Bundle;

import android.support.annotation.Nullable;

import com.example.cyy.interthread.base.BaseActivity;

/\*\*

\* Created by user on 2017/8/17.

\*/

public class AsyncTaskTestActivity extends BaseActivity{

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(android.R.layout.test\_list\_item);

*//刚开始从主线程（UI线程）开始*

getProcessInfo(false);

*//执行异步任务*

new MyAsyncTask().execute();

}

class MyAsyncTask extends AsyncTask<String, Void, String>{

@Override

protected String doInBackground(String... params) {

*//子线程（工作线程），处理耗时操作*

getProcessInfo(false);

return null;

}

@Override

protected void onPostExecute(String s) {

super.onPostExecute(s);

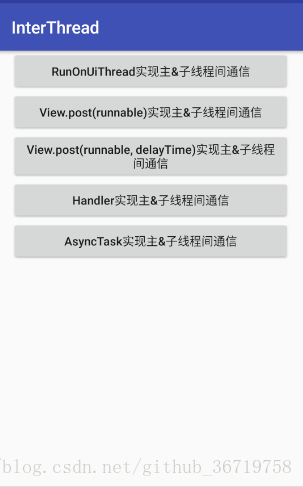
*//回到主线程（UI线程），处理UI*

getProcessInfo(true);

}

}

}

好了，代码上完了，由于代码没有难点，就不细讲了，下边上效果：  
  
主界面长这样子，然后我们再任意点击一个按钮看看：  
  
如上就是我们一个主线程->子线程->主线程的流程。当打开一个页面时默认是从主线程开始的。通过图可以看到，主线程的id是1。  
源码下载地址：<https://github.com/foreverYuan/InterThread>