

# **Android 应用软件设计**

## **E 4 Multi-thread Programming**

学号：SA17225268

姓名：彭明

报告撰写时间：2017/10/21

# 1.主题概述

本次主题是围绕多线程编程来展开的，让我们学会使用多线程编程，把一些比较耗时的操作从 UI 主线程中分离出来，放在另外一个线程中，待操作完成之后再主线程中完成更新 UI 的操作。

## 2.假设

主题假设的场景是用户处于登录状态时，loginState 为 1，否则为 0。

### 3.实现或证明

1. 修改 LoginOrRegister 代码，当导航到该页面时，使用 SharedPreferences 保存或读取用户登录状态

在 LoginOrRegister.java 的 onCreate 方法中加入如下代码：

```
SharedPreferences pref = getSharedPreferences("userdata",
MODE_PRIVATE);

if(pref.getString("userName", "") == ""){
    mEmailSignInButton.setVisibility(View.GONE);
} else {
    mEmailSignUpButton.setVisibility(View.GONE);
    mEmailView.setText(pref.getString("userName", ""));
}
```

2. 修改 LoginOrRegister 代码，当用户点击登录或注册成功后，将 username 和 loginState 为 1 写入到 SharedPreferences

由于我的登录和注册都调用了同一个方法来判断是否符合规则，所以我在那个方法中判断符合规则之后，加入如下代码：

```
SharedPreferences.Editor editor =
getSharedPreferences("userdata", MODE_PRIVATE).edit();

editor.putString("userName", username);
editor.putInt("loginState", 1);
editor.apply();
```

3. 修改 LoginOrRegister 代码，当用户点击返回后判断有无用户名记录，有则将 loginState 为 0 写入到 SharedPreferences

```
if (item.getItemId() == android.R.id.home) {
    SharedPreferences pref = getSharedPreferences("userdata",
MODE_PRIVATE);

    if(pref.getString("userName ", "") != ""){
```

```
        SharedPreferences.Editor editor =  
getSharedPreferences("userdata", MODE_PRIVATE).edit();  
        editor.putInt("loginState", 0);  
        editor.apply();  
    }  
    finish();  
}
```

4. 修改 MainScreen 和 SCOSEntry 代码，将用户是否登录的判断逻辑改成使用 SharedPreferences 的 loginState 值进行功能实现

删除之前 SCOSEntry 中启动 MainScreen 的 Intent 的 putExtra 方法，不用再传值，将 MainScreen 中根据 SCOSEntry 传过来的值是否匹配，来更新 GridView 的代码改为如下所示：

```
        SharedPreferences pref = getSharedPreferences("userdata",  
MODE_PRIVATE);  
        if(pref.getInt("loginState", 0) == 0){  
            list.remove(1);  
            list.remove(0);  
            myAdapter.notifyDataSetChanged();  
        }
```

5. 在 SCOS 的 es.source.code.activity 包中，新建 SCOSHelper.java 类为 Activity 子类

```
public class SCOSHelper extends AppCompatActivity {  
    @BindView(R.id.toolbar)  
    Toolbar toolbar;  
    @BindView(R.id.gv_helper)  
    GridView gvHelper;  
  
    private static final int REQUEST_CALL_PHONE = 0;  
    private static final int REQUEST_SEND_SMS = 1;
```

```
private String[] data = {
    "用户使用协议",
    "关于系统",
    "电话人工帮助",
    "短信帮助",
    "邮件帮助"
};

private Handler handler;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_scoshelper);
    ButterKnife.bind(this);

    setSupportActionBar(toolbar);
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);

    //1) 使用 GridView 显示该页面帮助选项
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, data);
    gvHelper.setAdapter(adapter);

    //4) 邮件帮助, 使用 Handler 接收消息
    handler = new Handler() {
        @Override
        public void handleMessage(Message msg) {
            switch (msg.what) {
                case 1:
                    if(msg.arg1 == 1){
                        Toast.makeText(SCOSHelper.this, "求助邮件已发送成功", Toast.LENGTH_SHORT).show();
                    }
                    break;
            }
        }
    };
}
```

```
        }

    }

};

gvHelper.setOnItemClickListener(new
AdapterView.OnItemClickListener() {

    @Override

    public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {

        switch (position) {

            case 0:

                break;

            case 1:

                break;

            case 2:

                callPhone();

                break;

            case 3:

                sendSMS();

                break;

            case 4:

                MailSender mailSender = new MailSender();

                mailSender.start();

                break;

        }

    }

});

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {

    switch (requestCode) {
```

```
        case REQUEST_CALL_PHONE:
            if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                callPhone();
            }
            break;
        case REQUEST_SEND_SMS:
            if (grantResults[1] == PackageManager.PERMISSION_GRANTED) {
                sendSMS();
            }
            break;
        default:
            super.onRequestPermissionsResult(requestCode,
permissions, grantResults);
    }
}

//2) 电话人工帮助
private void callPhone() {
    try{
        if (ContextCompat.checkSelfPermission(SCOSHelper.this,
CALL_PHONE)
            != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(SCOSHelper.this, new
String[]{CALL_PHONE, SEND_SMS}, REQUEST_CALL_PHONE);
        } else {
            Intent ImplicitIntent = new Intent(Intent.ACTION_CALL,
Uri.parse("tel:" + 5554));
            startActivity(ImplicitIntent);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



## //3) 短信帮助

```

private void sendSMS() {
    try{
        if(ContextCompat.checkSelfPermission(SCOSHelper.this,
SEND_SMS)

            != PackageManager.PERMISSION_GRANTED) {

            ActivityCompat.requestPermissions(SCOSHelper.this, new
String[]{CALL_PHONE, SEND_SMS}, REQUEST_SEND_SMS);

        } else {

            SmsManager.getDefault().sendTextMessage("5554", null,
"test scos helper", null, null);

            Toast.makeText(this, "求助短信发送成功",
Toast.LENGTH_SHORT).show();

        }

    } catch (Exception e) {

        e.printStackTrace();

    }

}

```

## //4) 邮件帮助

```

class MailSender extends Thread{

    @Override

    public void run() {

        Intent intent = new Intent(Intent.ACTION_SENDTO);

        intent.setData(Uri.parse("mailto:pengming@gamil.com"));

        intent.putExtra(Intent.EXTRA_SUBJECT, "您的建议");

        intent.putExtra(Intent.EXTRA_TEXT, "我们很希望能得到您的建议!!!

");

        startActivity(intent);

        Message msg = new Message();

        msg.what = 1;

        msg.arg1 = 1;

        handler.sendMessage(msg);

```

```
}  
  
}  
  
}
```

## 6. 修改中 FoodOrderView 代码，使用 AsyncTask 模拟结账功能

在结账按钮的 onClick 方法中加入如下代码：

```
OrderPayTask orderPayTask = new OrderPayTask();  
orderPayTask.execute((Void) null);
```

在 FoodOrderView 的 PlaceholderFragment 类中加入如下代码：

```
public class OrderPayTask extends AsyncTask<Void, Integer, Boolean>{  
  
    @Override  
    protected void onPreExecute() {  
        progressBar.setVisibility(View.VISIBLE);  
    }  
  
    @Override  
    protected Boolean doInBackground(Void... params) {  
        for(int i = 0; i < 100; i++){  
            if(isCancelled()){  
                break;  
            }  
            publishProgress(i);  
  
            try{  
                Thread.sleep(60);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
                return false;  
            }  
        }  
    }  
}
```

```
        return true;
    }

    @Override
    protected void onProgressUpdate(Integer... values) {
        if(isCancelled()){
            return;
        }

        progressBar.setProgress(values[0]);
    }

    @Override
    protected void onPostExecute(Boolean success) {
        progressBar.setVisibility(View.GONE);

        if(success){
            orderSubmit.setEnabled(false);
            Toast.makeText(getApplicationContext(), "本次结账金额为: ",
Toast.LENGTH_SHORT).show();
        }
    }
}
```

## 4. 结论

1. Android 应用开发中，提高程序性能的方法有：
  - 1) 采用合理的数据结构和算法提高程序性能
  - 2) 界面布局的优化
  - 3) 采用多线程、缓存数据、延迟加载、提前加载等手段
  
2. Android 应用开发中，避免 ANR 异常出现的方法有：
  - 1) 避免在主线程上进行复杂耗时的操作
  - 2) `broadcastReceiver` 要进行复杂操作的的时候，可以在 `onReceive()`方法中启动一个 `Service` 来处理
  - 3) 在设计及代码编写阶段避免出现同步/死锁或者错误处理不恰当等情况
  
3. 在多线程编程中，保证线程安全的方法有：
  - 1) 将整个类加锁（代码为 `synchronized(this)`）
  - 2) 定义一个 `object`，访问某些相关变量是，对该对象加锁（代码为 `synchronized(lockobj)`）

## 5.参考文献

1. Android 开发官方文档中关于 SharedPreferences 的介绍  
<https://developer.android.google.cn/reference/android/content/SharedPreferences.html>
2. android.permission.CALL\_PHONE 的使用  
[http://blog.csdn.net/weixin\\_37077539/article/details/56279789](http://blog.csdn.net/weixin_37077539/article/details/56279789)
3. android.permission.SEND\_SMS 的使用  
<http://www.cnblogs.com/llw1996/p/6721348.html>
4. Android 开发官方文档中关于 Handler 的介绍  
<https://developer.android.google.cn/reference/android/os/Handler.html>
5. Android 开发官方文档中关于 AsyncTask 的介绍  
<https://developer.android.google.cn/reference/android/os/AsyncTask.html>