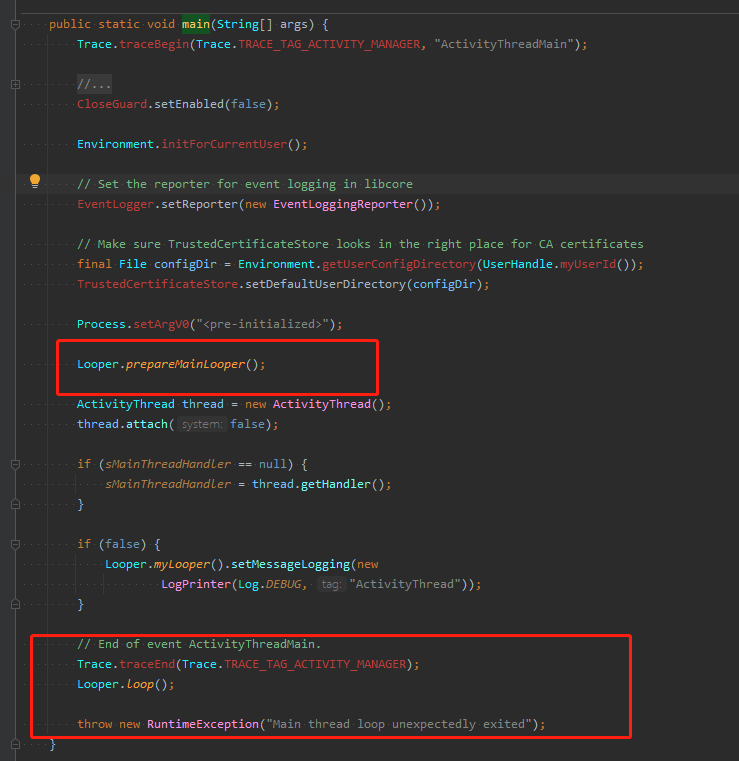
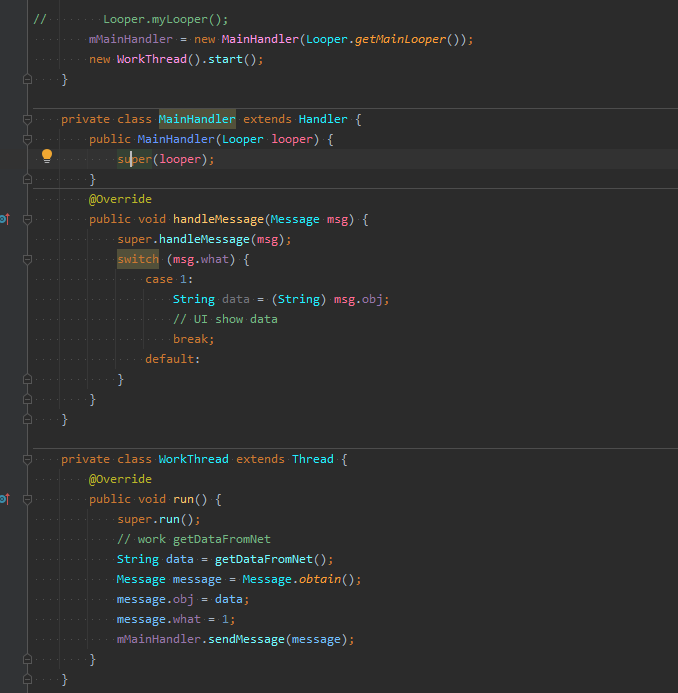
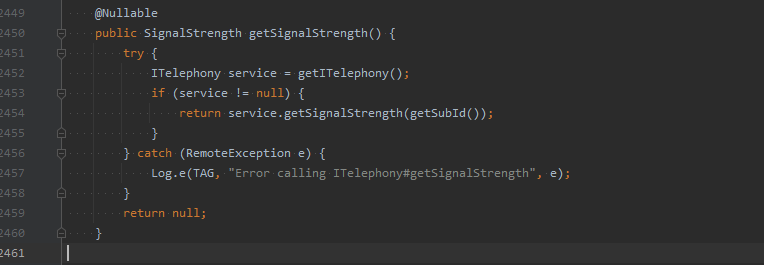
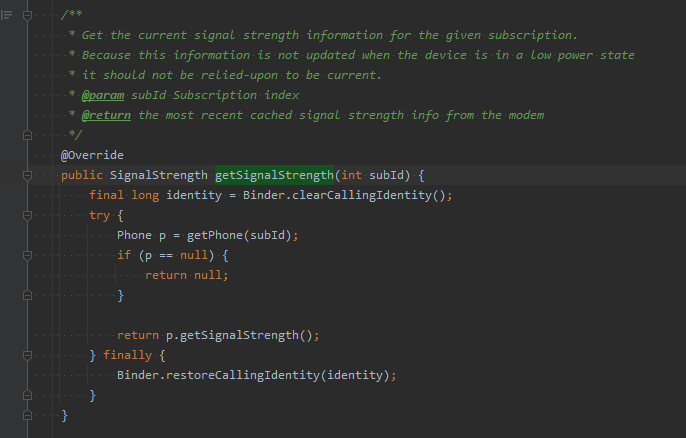
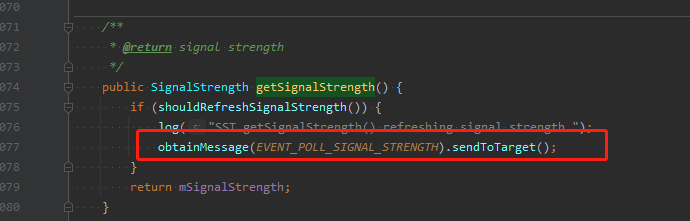
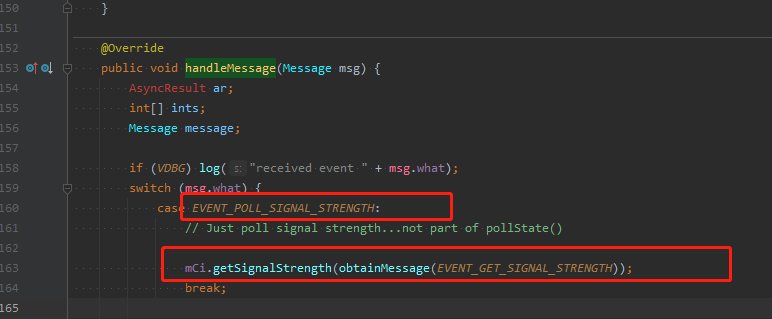
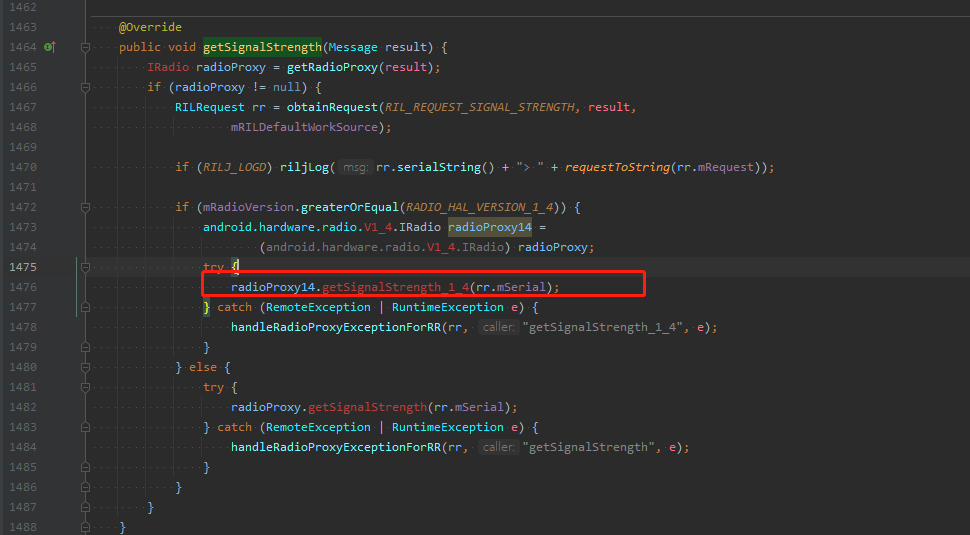
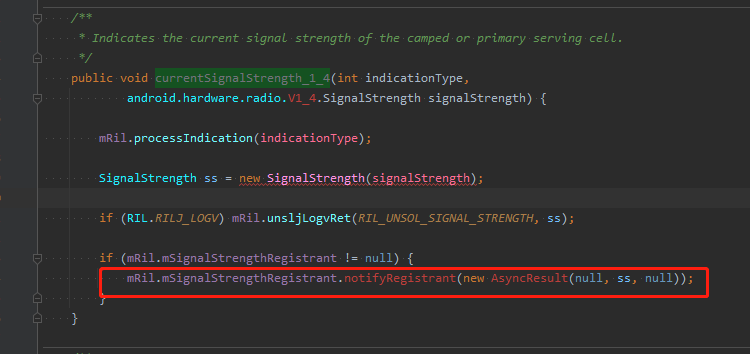
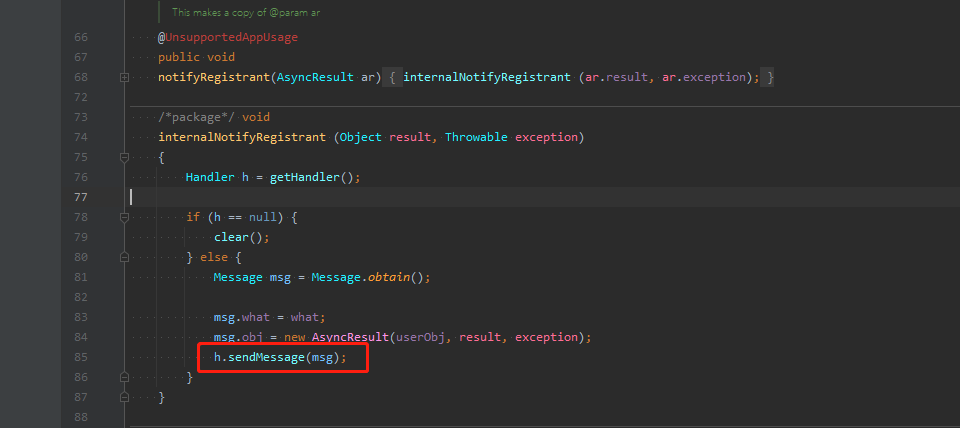
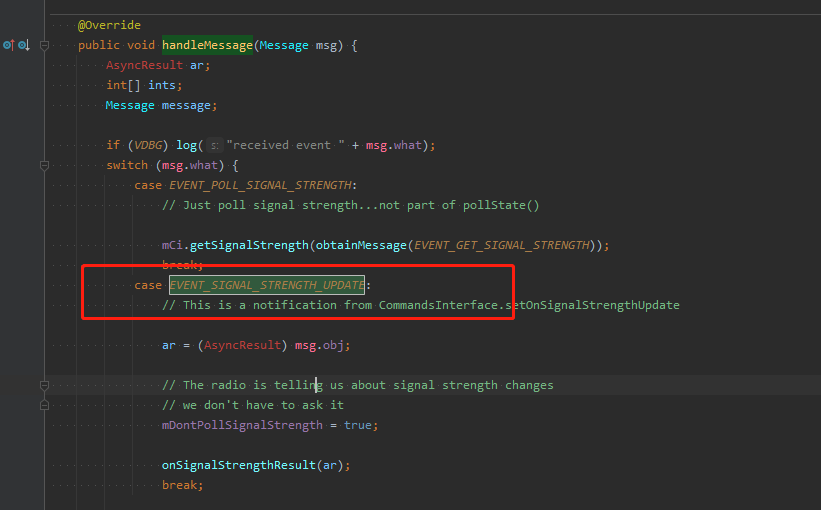
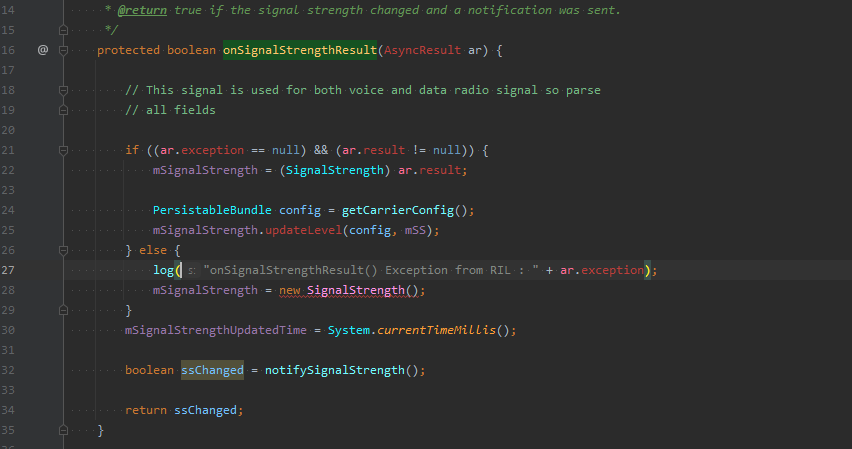
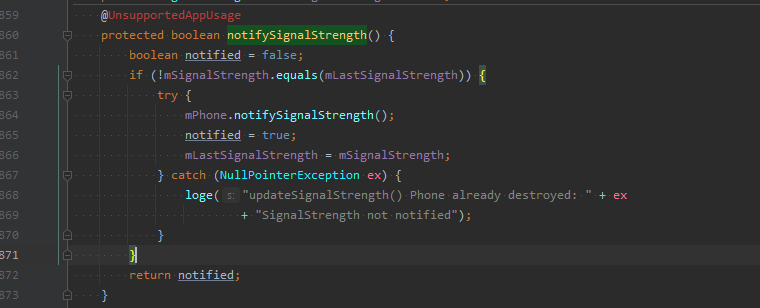
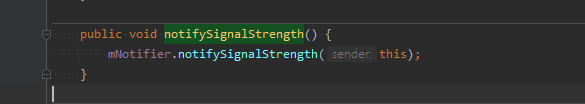
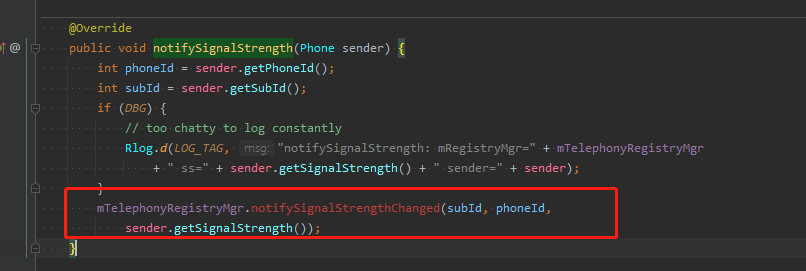
Android异步通信Handler

* 定义：Android多线程环境消息传递机制，多用于工作线程更新UI的操作信息。
* 背景：在Android开发中，为了UI操作是线程安全的，规定了至于徐UI线程更新Activity里的UI组件（UI线程即主线程，主线程即android.app.ActivityThread#main，主线程tid和进程的pid一致）。
* 相关概念

|  |  |
| --- | --- |
| **概念** | **定义作用** |
| 主线程MainThread | android.app.ActivityThread#main创建的线程即为主线程，更新UI |
| 子线程 | 非主线程，耗时工作 |
| 消息Message | 线程间通讯的数据单元，封装存储徐操作的通信信息 |
| 消息队列MessageQueue | 队列数据结构，存储消息，先进先出 |
| 处理者Handler | 线程间的通信媒介，线程消息的处理者，sendMessage和handleMessage |
| 循环器Looper | 消息队列和处理者的通信媒介。  1.消息获取，从MessageQueue取出消息Message  2.消息分发，将取出的消息发送给对应的处理者Handler  3.一个线程对应一个Looper一个MessageQueue多个Handler |

* 工作流程
  1. 异步通信准备
  2. 消息发送
  3. 消息循环
  4. 消息处理

|  |  |  |
| --- | --- | --- |
| **步骤** | **具体描述** | **备注** |
| 1.异步通信准备 | 在主线程中创建：   1. 处理器对象Looper 2. 消息队列对象MessageQueue 3. Handler对象 | 1. Looper和MessageQueue都属于主线程 2. 创建MessageQueue后，Looper自动进入消息循环 3. 此时Handler自动绑定了主线程Looper、MessageQueue |
| 2.消息入列 | 工作线程通过Handler发送消息到消息队里中 |  |
| 3.消息循环 | 1.消息出列：Looper循环取出消息队列中的消息  2.消息分发：Looper将取出的消息发送给创建该消息的处理者 | 若消息循环过程中，消息队列为空，则线程阻塞等待消息 |
| 4.消息处理 | 1.处理者接收处理器发送过来的消息  2.处理者根据消息进行UI操作 |  |

* 使用实例
  1. 子线程发消息到主线程处理
  2. 子线程自己处理消息
* Tele中实例：信号获取到上报显示
  1. android.telephony.TelephonyManager#getSignalStrength 
  2. com.android.phone.PhoneInterfaceManager#getSignalStrength 
  3. com.android.internal.telephony.Phone#getSignalStrength
  4. com.android.internal.telephony.ServiceStateTracker#getSignalStrength 
  5. com.android.internal.telephony.ServiceStateTracker#handleMessage 
  6. com.android.internal.telephony.RIL#getSignalStrength(消息完成下发Modem) 
  7. com.android.internal.telephony.ServiceStateTracker#ServiceStateTracker(消息注册)  com.android.internal.telephony.BaseCommands#setOnSignalStrengthUpdate 
  8. com.android.internal.telephony.RadioIndication#currentSignalStrength\_1\_4(消息上报) 
  9. android.os.Registrant(消息注册分发封装) 
  10. com.android.internal.telephony.ServiceStateTracker#handleMessage: EVENT\_SIGNAL\_STRENGTH\_UPDATE 
  11. com.android.internal.telephony.ServiceStateTracker#onSignalStrengthResult 
  12. com.android.internal.telephony.ServiceStateTracker#notifySignalStrength 
  13. com.android.internal.telephony.Phone#notifySignalStrength 
  14. com.android.internal.telephony.DefaultPhoneNotifier#notifySignalStrength 
  15. com.android.server.TelephonyRegistry#notifySignalStrengthForPhoneId 