Android异步方式之一 Handler

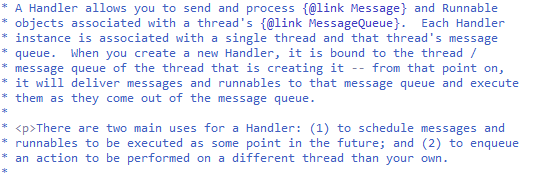
说明：

前面看了《Launcher启动流程详解》，对桌面的启动流程有了比较详细的理解。当时对Handler的内部实现机制不是很理解，后面私下查了下源码，基本对流程有了一个完整的理解，这个文档是一个总结。

由于篇幅问题，HandlerThread和DeferredHandler就不解释了。理解了下面的文档后相信很容易理解这两部分的实现原理。如果需要，下次开会的时候我再详细介绍。

第一部分概念

Handler



按照字面意思，Hanlder的作用就是处理Message和Runnable的工具。在开发的时候主要有两个地方会用到Handler：处理计划的消息（schedule message，这个好理解不好翻译）和runnable。

Message

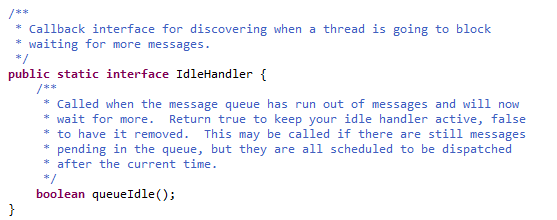
C:\Users\admin\AppData\Roaming\Tencent\Users\441699771\QQ\WinTemp\RichOle\_0_G%F2[P}E]LRN_A5XFO%1.png

Message定义了一个内容的描述信息和任意类型数据的data对象。消息是用于线程之间传递信息的载体。

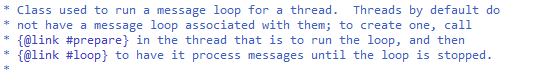
MessageQueue



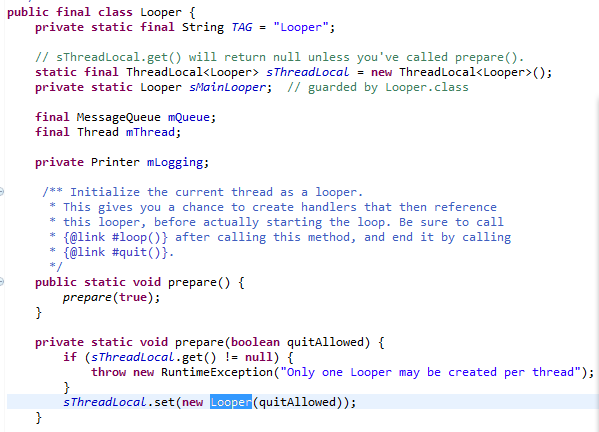
用于保存Message的一个列表。这个类里面有一个重要的接口，这个接口在桌面中有用到，后面再说：



Looper



Looper是来不断获取Message对象的一个类。



这个类中有一个很关键的变量:



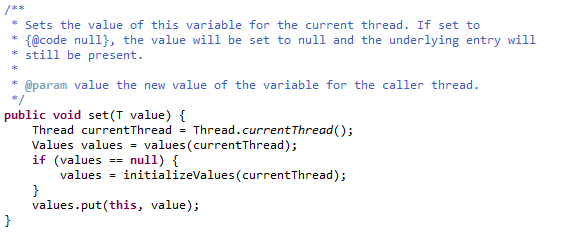
网上有很多关于Looper的介绍，基本都忽略了这个变量的介绍。其实如果不明白ThreadLocal这个类的作用，就很难真正理解Looper是怎么操作的。要不然怎么理解Android在任何线程中都调用Looper.prepare()函数去初始化Looper，换句话说：Looper是怎么处理不同线程的MessageQueue的？

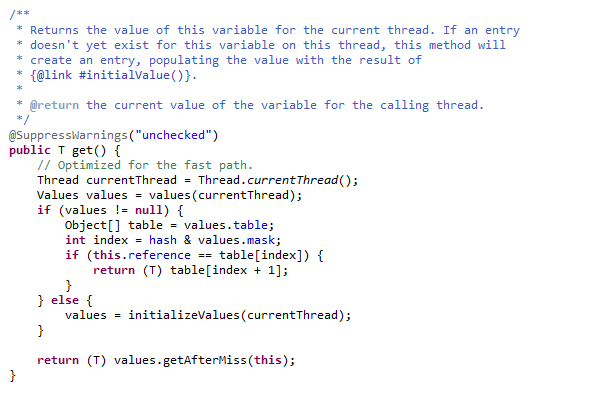
ThreadLocal



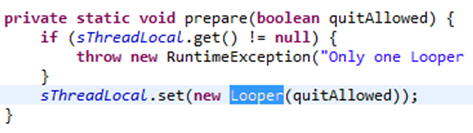
这是一个线程变量的保存类，从源码上看就是把各个线程的变量值保存在了一个Map中，其中key是线程的id，值是需要保存的对象。在Looper中，保存的对象就是Looper对象。

这个类最主要的函数就是get（）和set（T value），看源码就知道是怎么做到的：

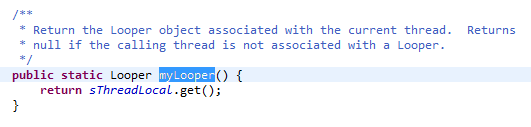




通过对这个类的理解，我们可以知道，在任何线程中对Looper.prepare()调用后，Looper的全局对象都会存储一个新的Looper对象。



后面通过myLooper()函数获取各线程的Looper对象，获取这个Looper对象的目的主要是为了拿到Looper对象中的消息队列。



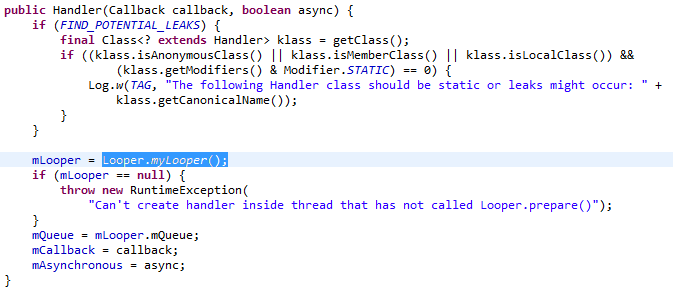
源码分析

一般在一个线程中初始化一个Handler实例都是这样的：



第一步：按照前面的介绍Looper.prepare()函数先生存一个新的Looper对象，并且保存在一个全局线程变量列表中。各个线程获取各自的Looper都是根据线程id获取的，所以每个线程的MessageQueue都是相对独立的。

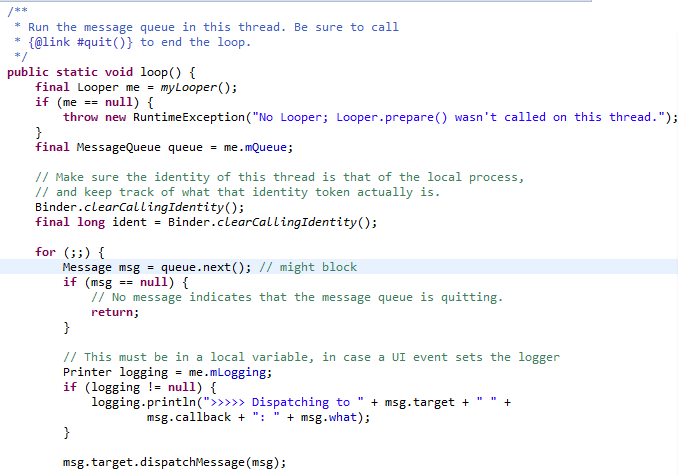
第二步：生成一个新的Handler。这个对象是怎么关联Looper的呢？Handler的构造函数最终会调用到下面这个构造函数。



通过前面介绍Looper.myLooper()函数是获取当前线程的Looper对象。所以这里把该Handler对象关联到了当前线程的Looper。我们还可以指定一个自己的Looper对象给一个Handler。这个部分代码在Turbo的LauncherMode.java中有用到，后面介绍如何使用。

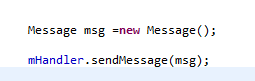
第三步：启动Looper的无限循环函数loop()去不断的侦听从其他线程发送过来的Message。

Loop()是怎么实现的呢？

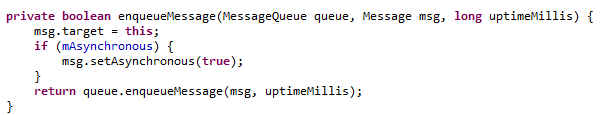


这个函数就是通过一个无限循环不断的去遍历关联的消息队列是否有新的消息内容。其中msg.target就是关联的Handler对象。这个对象在发送消息的时候被初始化。

我们往handler中发送一个消息的时候一般这么调用：



sendMessage最终会调用下面这个函数：

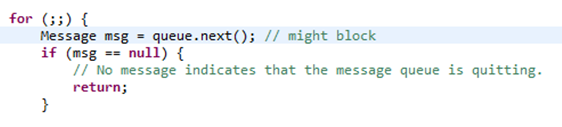


从这里就可知道Looper类的loop函数可以是直接调用了Handler对象的handleMessage函数：



之前说过，MessageQueue中有个一个很重要的接口叫做IdleHandler。顾名思义这个接口就是在线程没事做后开始调用的。

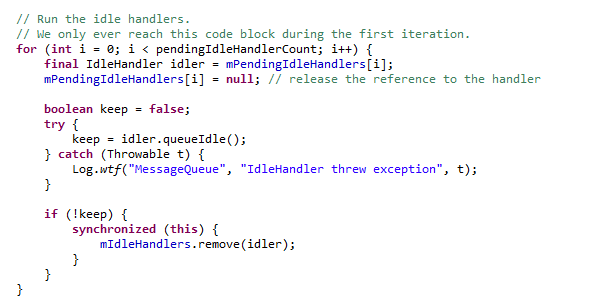
Loop函数是通过调用MessageQueue的next函数获取Message的。



Next函数是如果获取消息的呢。

Message next(){

//如果没有获取到Message，就寻找是否有IdlerHandler



}

注意：这里的处理是直接调用IdlerHandler对象的queueIdle函数。

这部分代码在桌面的DeferredHandler中使用到了。

第三部分：