



链滴

Android源码:Looper和Handler

作者: [wind](#)

原文链接: <https://ld246.com/article/1441808152630>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Android和其他系统一样，都是靠消息驱动来工作的。原理如下：1、消息队列(MessageQueue)，以添加消息。2、消息循环，从队列中取出消息进行处理，无限循环。Looper类：封装消息的循环程。Handler类：辅助类，用于添加消息和处理消息 (SendMessage和HandleMessage)。

####Looper类：

Looper类既然封装的是消息循环，就肯定有一个线程让它在不停的运作，代码如下：

```
public class LooperThread extends Thread{
public void run(){
....
Looper.prepare();
Looper.loop();
}
}
```

主要函数是prepare()和loop()，prepare很容易理解，就是将looper对象绑定到当前的线程中来。下来看loop()代码：

```
public static final void loop(){
Looper me=myLooper();
MessageQueue queue=me.mQueue;
while(true){
Message msg=queue.next();
if(msg!=null){
if(msg.target==null){
return;
}
msg.target.dispatchMessage(msg);
msg.recycle();
}
}
}
```

看到while(true)循环了吧，这里就是不断的从队列中提取消息，然后调用dispatchMessage来处理息的地方。这里的MessageQueue就是消息队列了。

####Handler类：

Handler类中也有mQueue和mLooper，和Looper中指的都是同一个。下面我们看Handler是如何为辅助类来帮助添加消息的：

```
public boolean sendMessageAtTime(Message msg,long uptimeMills){
boolean sent=false;
MessageQueue queue=mQueue;
```

```
if(queue!=null){
msg.target=this;
sent=queue.enqueueMessage(msg,uptimeMills);
}
return sent;
}
```

这里把target设为自己，是因为，Looper循环后处理的函数为msg.target.dispatchMessage(msg)而Handler同时也是消息处理的执行者，所以target应设为自己。

其实这两个类都比较简单，看源码自己也可以看懂了。用过Handler的人都知道，Handler是用来多程之间传递消息的。

Looper对象是和一个线程相挂钩，TLS(Thread Local Save)，默认是挂在主线程上。所以，如果有需，不想Looper对象挂在主线程上，可以自己重写Looper.prepare()和Looper.loop()来自定义一个消机制。