

转载请注明链接：https://blog.csdn.net/feather_wch/article/details/81136078

总结所有Handler消息机制相关的面试题。在其他博文已经有答案的会提供[猎羽的博文链接](#)进行参考。没有答案的会在文章内给出。

Android面试题-Handler消息机制(50题)

版本号：2018/09/02-1(14:30)

注意

- [☆] 标记的题目，是额外补充的题目，会直接给出答案。
- 没有标记的题目，详细答案请参考：[答案参考-Handler消息机制详解](#)
- [Android面试题-Handler消息机制\(50题\)](#)
 - [注意](#)
 - [Handler\(17\)](#)
 - [MessageQueue\(9\)](#)
 - [Looper\(17\)](#)
 - [ThreadLocal\(7\)](#)

Handler(17)

1. Handler是什么？
2. 消息机制是什么？
3. 为什么不能在子线程中访问UI？
4. 在子线程中创建Handler报错是为什么？
5. 如何在子线程创建Looper？

```
Looper.prepare();
```

6. 为什么通过Handler能实现线程的切换？
7. [☆] Handler.post的逻辑在哪个线程执行的，是由Looper所在线程还是Handler所在线程决定的？
 1. 由Looper所在线程决定的
 2. 最终逻辑是在Looper.loop()方法中，从MsgQueue中拿出msg，并且执行其逻辑，这是在Looper中执行的，因此有Looper所在线程决定。
8. [☆] Looper和Handler一定要处于一个线程吗？子线程中可以用MainLooper去创建Handler吗？
 1. 可以的。

2. 子线程中 `Handler handler = new Handler(Looper.getMainLooper());` , 此时两者就不在一个线程中。

9. Handler的post/send()的原理

1. 通过一系列 `sendMessageXXX()` 方法将 `msg` 通过消息队列的 `enqueueMessage()` 加入到队列中。

10. Handler的post方法发送的是同步消息吗？可以发送异步消息吗？

1. 用户层面发送的都是同步消息
2. 不能发送异步消息
3. 异步消息只能由系统发送。

11. [☆]Handler的post()和postDelayed()方法的异同？

1. 底层都是调用的 `sendMessageDelayed()`
2. `post()` 传入的时间参数为0
3. `postDelayed()` 传入的时间参数是需要的时间间隔。

12. Handler的postDelayed的底层机制

13. MessageQueue.next()会因为发现了延迟消息，而进行阻塞。那么为什么后面加入的非延迟消息没有被阻塞呢？

14. Handler的dispatchMessage()分发消息的处理流程？

15. Handler为什么要有Callback的构造方法？

不需要派生Handler

16. Handler构造方法中通过 `Looper.myLooper()`；是如何获取到当前线程的Looper的？

`myLooper()` 内部使用 `ThreadLocal` 实现，因此能够获取各个线程自己的Looper

17. 主线程如何向子线程发送消息？

MessageQueue(9)

1. MessageQueue是什么？

2. MessageQueue的主要两个操作是什么？有什么用？

3. [☆] MessageQueue中底层是采用的队列？

错误！

采用 单链表 的数据结构来维护消息队列，而不是采用队列

4. MessageQueue的enqueueMessage()方法的原理，如何进行线程同步的？

1. 就是单链表的插入操作
2. 如果消息队列被阻塞回调用 `nativeWake` 去唤醒。
3. 用 `synchronized` 代码块去进行同步。

5. MessageQueue的next()方法内部的原理？

分为三种情况进行处理。

6. next()是如何处理一般消息的?
7. next()是如何处理同步屏障的?
8. next()是如何处理延迟消息的额?
9. [☆]Looper.loop()是如何阻塞的? MessageQueue.next()是如何阻塞的?
通过native方法: nativePollOnce()进行精准时间的阻塞。

Looper(17)

1. Looper是什么?
2. [☆] 如何开启消息循环?

```
Looper.loop();
```
3. Looper的构造内部会创建消息队列。
4. 主线程ActivityThread中的Looper的创建和获取
5. Looper的两个退出方法?
6. quit和quitSafely有什么区别
7. 子线程中创建了Looper, 在使用完毕后, 终止消息循环的方法?
8. Looper.loop()的源码流程?
 1. 获取到Looper和消息队列
 2. for无限循环, 阻塞于消息队列的next方法
 3. 取出消息后调用 `msg.target.dispatchMessage(msg)` 进行消息分发
9. [☆] Looper.loop()在什么情况下会退出?
 1. next方法返回的`msg == null`
 2. 线程意外终止
10. MessageQueue的next方法什么时候会返回null?
11. [☆] Looper.quit/quitSafely的本质是什么?
 1. 让消息队列的next()返回null, 依次来退出Looper.loop()
12. Looper.loop()方法执行时, 如果内部的myLooper()获取不到Looper会出现什么结果?
13. 主线程是如何准备消息循环的?
14. ActivityThread中的Handler H的作用?
15. 如何获取主线程的MainLooper
16. Android如何保证一个线程最多只能有一个Looper? 如何保证只有一个MessageQueue
17. Handler消息机制中, 一个looper是如何区分多个Handler的?

ThreadLocal(7)

1. ThreadLocal是什么?
2. ThreadLocal的作用?
3. ThreadLocal的两个应用场景?

4. ThreadLocal的使用

同一个ThreadLocal调用set(xxx)和get()

5. ThreadLocal的原理

1. `thread.threadLocals` 就是当前线程thread中的 ThreadLocalMap
2. ThreadLocalMap中有一个table数组，元素是Entry。根据 ThreadLocal (需要转换获取到 Hash Key)能get到对应的Entry。
3. Entry中 key 为ThreadLocal，value 就是存储的数值。

6. [☆]如何获取到当前线程

`Thread.currentThread()` 就是当前线程。

7. [☆]如何在ThreadLocalMap中，ThreadLocal如何作为键值对中的key?

通过ThreadLocal计算出Hash key，通过这个哈希值来进行存储和读取的。