继续撩一撩Java synchronized

在印象中Java synchronized关键字对每个Java工程狮来说都耳熟能起痒，讲起来滔滔不绝！但今天还是要再聊聊synchronized。

synchronized关键字有各种用法，这里就只说一种。先看下面问题：

*将synchronized关键放在一个类A的普通public方法func前，生成A的两个实例a1和a2。分别将这个两个实例交给两个线程，同时调用对应的方法func，它们可以“并发”运行吗？*

Thread1: a1.func();

Thread2: a2.func();

相信90%的Java工程师看到这个问题，嘴角都翘起来了。

而我面试过程中我所得到的答案统计结果是：选择“不可以”的结果数大于50%。

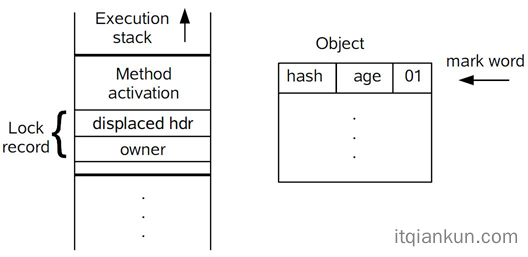
先不管答案是什么，无论回答什么，接下来的问题一般是问：为什么呢？可以解释一下吗？

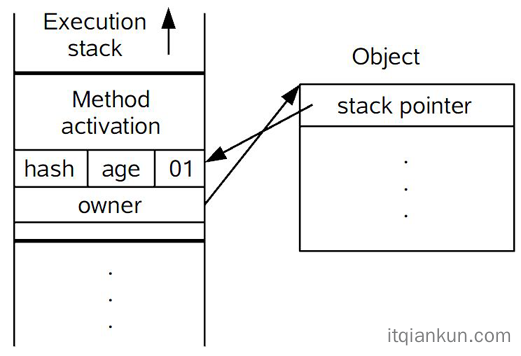
而这个问题听到的答案就更不标准了，甚至原因和答案都是矛盾的！

抛开前面的问题回到Java语言本身，Java语言支持多线程，而支持的核心是线程同步：即在不同线程之间协调运行和数据访问。synchronized就是用来协调线程运行和数据访问的。

运行synchronized对应的方法是以获取锁开始，释放锁结束。那么锁又在哪里呢？

简单说，锁就在对象头的Mark Word中。或者准确的说，在对象没有膨胀为重量级锁之前，锁就在每个对象头的Mark Word中（可参考截图）。





如果两个不同线程并发同步访问同一个对象，轻量级锁将膨胀为重量级Object Monitor。

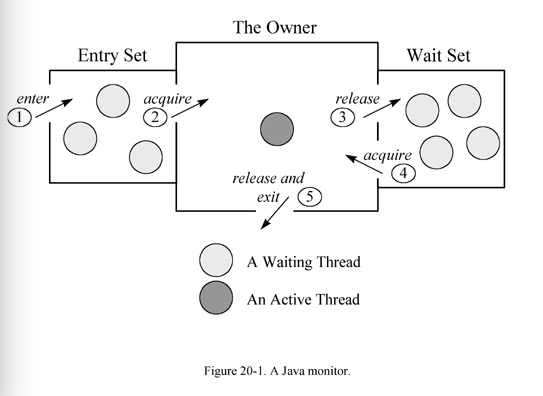
Monitor包含三部分：entry set，Owner，wait set。

当线程到达Monitor区域时，它先被放置到entry set，entry set相当于大楼的前门大厅。如果大厅为空且Monitor没有主人，则线程获取Monitor成为Owner继续执行，执行完成后退出Monitor。

当Owner运行过程中，调用wait则会释放monitor进入wait set；如果调用notify则不会立即释放owner，如果稍后释放Owner，则某一个等待线程会复活。

如果前一个Owner不调用notify，那么只有entry set中的等待线程有资格成为下一任owner。如果调用了notify，则entry set中线程和wait set中的线程同时竞标下一任Owner。





<https://www.artima.com/insidejvm/ed2/threadsynch.html>

<https://wiki.openjdk.java.net/display/HotSpot/Synchronization>

<https://george24601.github.io/2018/12/23/synchronized.html>

<https://www.itqiankun.com/article/bias-lightweight-synchronized-lock>

[Agesen99] O. Agesen, D. Detlefs, A. Garthwaite, R. Knippel, Y. S. Ramakrishna, D. White: An Efficient Meta-lock for Implementing Ubiquitous Synchronization. In Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, pages 207-222. ACM Press, 1999. [doi:10.1145/320384.320402](http://dx.doi.org/10.1145/320384.320402)

[Bacon98] D. F. Bacon, R. Konuru, C. Murthy, M. Serrano: Thin Locks: Featherweight Synchronization for Java. In Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation, pages 258-268. ACM Press, 1998. [doi:10.1145/277650.277734](http://dx.doi.org/10.1145/277650.277734)