

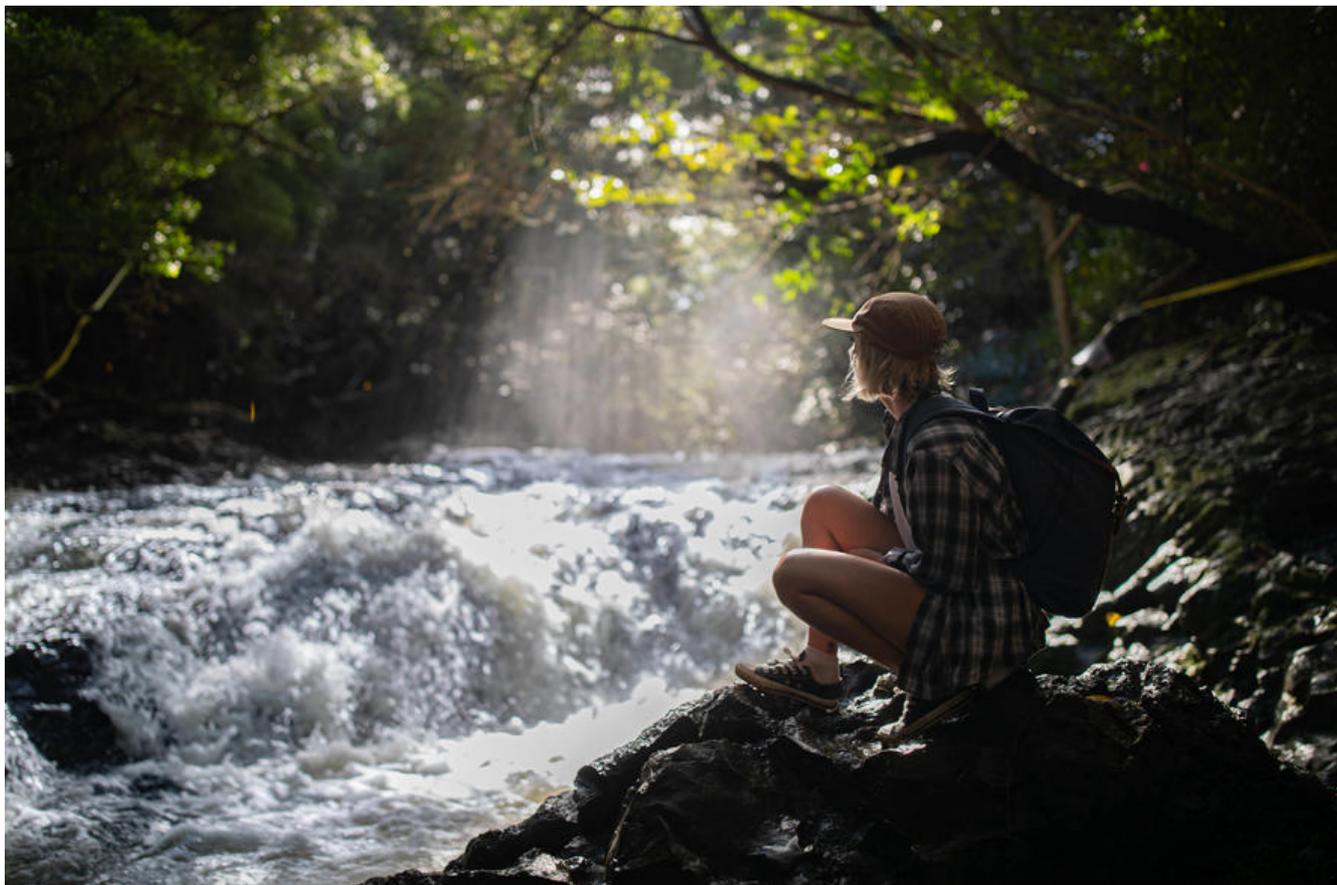
大二经历人生中第二场面试 -- 钉钉一面凉经

作者: [Gouzhong1223](#)

原文链接: <https://ld246.com/article/1584427018993>

来源网站: [链滴](#)

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



背景

先介绍一下背景，本人双非本科，然后非科班大二在读，从没想过投简历找实习的，但是想到同实验的一个大二的同学上学期面试头条拿到实习 Offer，加上一个在钉钉的师兄说可以给我模拟面，没过不进内网，过了按正常来就行，我就抱着试一试的想法把简历发了过去。

有点小紧张

昨天吃过晚饭之后，对方称是面试官，问我有没有时间，我说有，现在就有，然后面试官就说用钉钉频面。

开始了

Q: 先简单做个自我介绍吧

A: 吧啦吧啦讲一通。

Q: 好的我了解了，那你先跟我讲一讲 HashMap 的底层实现原理吧。

A: JDK1.7和 JDK1.8的 HashMap 底层实现有很大的改变，我先从 JDK1.7开始说起吧，然后吧啦吧吧讲

Q: 你刚刚说到了 JDK1.7的 HashMap里面链表用的是头插法，JDK1.8之后就变成了尾插法，这是什么呢？

A: 使用头插法的话，在 HashMap resize 之后还会进行 rehash，有一定几率会造成链表循环，线

调用的时候会出现死锁状态。。。。。。使用尾插法就能完美的避免这个问题

Q: 那你能跟我说一下, 为什么 JDK1.8之后 HashMap 链表转化为红黑树的临界值为什么是 8 么? 什么不是 7或者 9?

A: 额, 这个我不是很清楚

然后面试官就跟我普及了一下概率论里面的泊松分布, 他说我应该是学过概率论的, 这方面可以去了一下, 然后就跟我讲了下为什么是 8

Q: 刚才那个问题没答上没关系, HashMap 是线程安全的么? 如果不是, 那你们在开发过程中怎么决这个问题的呢?

A: HashMap 不是线程安全的, 在开发过程中, 一般有三种可以替代的方案, 使用Collections.synchronizedMap或者 Hashtable 还有ConcurrentHashMap代替, 但是 Hashtable 不怎么常用, 最常说的就是ConcurrentHashMap。

Q: 哦? 那为什么不使用 Hashtable 而用ConcurrentHashMap呢?

A: 我看过 Hashtable 的源码, 涉及到对象操作的部分都是直接暴力加锁, 操作的时候直接锁住了整个对象这样的并发消耗是非常大的, 然后ConcurrentHashMap在 JDK1.7使用的是ReentrantLock来对面分段加锁, 锁住的是Segment, 理论上来说, 默认有 16个Segment就支持 16个线程并发操作对。然后因为 JDK1.6之后 Java 对synchronized锁进行了大量的优化, 所以 JDK1.8之后就把ReentrantLock加锁替换成了synchronized加锁, 效率也提升了。(这中间还说了点其他的, 我给忘了说了啥)

Q: 你刚刚说到了synchronized和ReentrantLock, 你能跟我说一下他们两个的区别么?

A: synchronized是基于 JVM 层面的, ReentrantLock是基于 Jdk 层面的。ReentrantLock采用的显式获取锁, 采用乐观锁机制, CAS 自旋。JDK1.6之前的synchronized使用的是悲观锁, JDK1.6之后就优化成了一个锁升级的机制, 锁状态会慢慢的从无锁升级成偏向锁, 轻量级锁, 重量级锁。并且锁可单向膨胀。(这里应该减分了, 我不太清楚具体的东西)

面试官补充

Q: 你刚刚说到 CAS, 你了解 CAS 么?

A: 比较与交换吧啦啦啦啦, 然后会导致 ABA 问题, 啦啦啦啦, 解决 ABA 问题有两种办法, 加时间戳加版本号。(这个地方应该也减分了)

Q: 回到刚才那个问题, 你说synchronized是基于 JVM 层面的, 那你知道它在 JVM 里面是怎么实的么?

A: synchronized 通过 Monitor 来实现线程同步, Monitor 是依赖于底层的操作系统的 Mutex Lock 来实现的线程同步。而Monitor又会根据对象头中的 Mark Word 判断该对象的锁状态, Mark Word 里面保存了当前对象的 HashCode, 锁状态, 锁标记, 以及 GC 年龄。

Q: 为什么有了锁状态还要有一个锁标记呢? 会不会多此一举?

A: 锁状态占有 2 个 bit, 按照排列组合最多能够表示 4 种状态, 但是我们还需要标识是否为偏向锁所以需要一个辅助字段来标识, 加上锁标记的1 bit, 刚好能够标识所有的锁状态 (这里应该也减分了)

面试官跟我讲了一下为什么, 我人裂开了

Q: 这些你都是通过什么途径了解到的?

A: 我看过一点 HotSpot 的源码，在上面看到的，里面有一个 mark.cpp和 mark.hpp有具体实现

Q: 哦? 那你跟我讲讲 mark.cpp具体是怎么实现锁标记和状态的

A: 额，这个我没怎么看懂

Q: 没关系没关系，你了解 JVM 内存模型么?

A: 有堆，本地方法栈，虚拟机栈，程序计数器还有方法区，其中程序计数器和虚拟机栈以及本地方法栈是线程私有的，堆是 GC 做用的主要区域。

Q: 你说堆是 GC 的工作的主要区域，你很了解 GC 么? 跟我讲讲 你知道的 GC 的常用算法

A: 有标记清除，标记整理，复制，剩下的不太了解。然后复制算法大面积的作用在堆上，堆上内存照 2: 3分为新生代和老年代，新生代按照 8: 1: 1分为 eden, from, to 三个区域，吧啦吧啦吧啦一堆。（这里应该也减分了）

Q: 你刚刚说到对象年龄加到 15就会移动到老年代，为什么是 15次不是其他的呢?

A: 哦我刚刚说到 Mark Word 里面会记录 GC 分代年龄，占用 4 bit，表示的范围是 0000->1111也是 0 到 15，这个可以调整，只能调小不能调大。

面试题补充

Q: 你了解数据库么? 了解储存引擎么?

A: 知道 MySQL，然后我说了一下 InnoDB 的一些东西，说了一下聚簇索引和非聚簇索引的区别。

Q: 你能跟我说一下 MySQL 中的 page 结构么?

A: 这里我有点映像，粗略地说了点，感觉应该说错了（减分!）

这里还问了很多数据库的问题，包括最左原则，为什么使用 B+Tree 不适用 BTree，我感觉我都知，就是用嘴巴说不清楚，这一块回答得不是很好

Q: 我看你在你的项目里面用到了 Redis，你能跟我说一下是怎么使用的么?

A: 吧啦吧啦吧啦回答，然后聊了项目（我感觉扣分了，主要是布隆过滤器不了解）

Q: 这个项目里面的持久层框架是你们实验室自研的么? 为什么抛弃传统的 ORM 框架?

A: 吧啦吧啦吧啦（乱吹逼，扣分）

Q: 能说一下 AOP 的底层实现么?

A: 我讲了 JDK 动态代理和 CgLib 动态代理的区别，说了一下性能差距。

Q: 做过均衡负载么? Nginx 均衡负载的一致性 Hash 那个一致性你怎么理解，到底是什么“一致性”?

后面问了啥我都忘了，我太紧张了，害sob

完了

Q: 你还有什么想问我么?

A: 我说你大概觉得我什么水平?

面试官说我基本功不是很扎实, 很多基础的东西有点欠缺, 然后看着有点年轻 (舒服了, 可能觉得我是01年的不靠谱吧), 经验不丰富。下来之后还是要多花时间补一补。

总结

我感觉是凉透了, 反正是试一试, 我就当扫盲了, 我现在感觉我像是没学过 Java 一样, 裂开了。