

作者：椰丝柔丝2

链接：<https://www.nowcoder.com/discuss/578614>

来源：牛客网

synchronized? mutex lock怎么实现的

1、Synchronized用过吗，其原理是什么？

这是一道Java面试中几乎百分百会问到的问题，因为没有任何写过并发程序的开发者会没听说或者没接触过Synchronized。Synchronized是由JVM实现的一种实现互斥同步的一种方式，如果你查看被Synchronized修饰过的程序块编译后的字节码，会发现，被Synchronized修饰过的程序块，在编译前后被编译器生成了monitorenter和monitorexit两个字节码指令。这两个指令是什么意思呢？在虚拟机执行到monitorenter指令时，首先要尝试

获取对象的锁：如果这个对象没有锁定，或者当前线程已经拥有了这个对象的锁，把锁的计数器+1；当执行monitorexit指令时将锁计数器-1；当计数器为0时，锁就被释放了。如果获取对象失败了，那当前线程就要阻塞等待，直到对象锁被另外一个线程释放为止。Java中Synchronize通过对对象头设置标记，达到了获取锁和释放锁的目的。

Java看过哪些源码？

线程池好处？多创线程就怎么了？压测的时候创建几千个线程才几毫秒这点儿开销有必要节省吗？内存开销，时间开销？线程池参数，execute执行流程，work？没有工作会删除吗？睡眠状态？idle怎么配置的？最大线程满了之后？

池化好处

1、线程池的重用

线程的创建和销毁的开销是巨大的，而通过线程池的重用大大减少了这些不必要的开销，当然既然少了这么多消费内存的开销，其线程执行速度也是突飞猛进的提升。

2、控制线程池的并发数

控制线程池的并发数可以有效的避免大量的线程池争夺CPU资源而造成堵塞。

3、线程池可以对线程进行管理

线程池可以提供定时、定期、单线程、并发数控制等功能。比如通过ScheduledThreadPool线程池来执行S秒后，每隔N秒执行一次的任务。

线程池参数：

18、创建线程池的几个核心构造参数？

Java 中的线程池的创建其实非常灵活，我们可以通过配置不同的参 数，创建出行为不同的线程池，这几个参数包括：

- corePoolSize:线程池的核心线程数。
- maximumPoolSize:线程池允许的最大线程数。
- keepAliveTime:超过核心线程数时闲置线程的存活时间。
- workQueue:任务执行前保存任务的队列，保存由 execute 方法提交的 Runnable 任务 。

作者： 椰丝柔丝2.note作者： 椰丝柔丝2.note

有哪些Map？ 还有啥Map？ 用的jdk几？ 说一下HashMap数据结构， put值散列冲突怎么解决？

链表树化转移数量？ 为什么是8为什么是6？ ？ 为什么数组要是二次幂？ 怎么扩容的？ 扩容 rehash的流程？

concurrentHashMap的散列流程？ concurrentHashMap怎么实现的线程安全？ CAS什么意思，怎么实现的？ Unsafe怎么实现？ concurrentHashMap什么时候用到CAS？ 并发情况下两个线程都到之后怎么插入的？ 初始化的时候两个线程都检测到需要初始化了，然后怎么做的？

Map->HashMap ->1.7、1.8

->HashTable

->LinkedHashMap

->TreeMap

->ConcurrentHashMap

HashMap 数据+链表/红黑树

put值散列冲突怎么解决->哈希冲突？

线性探测法（Linear Probing）

链表法

冲突解决方法对比

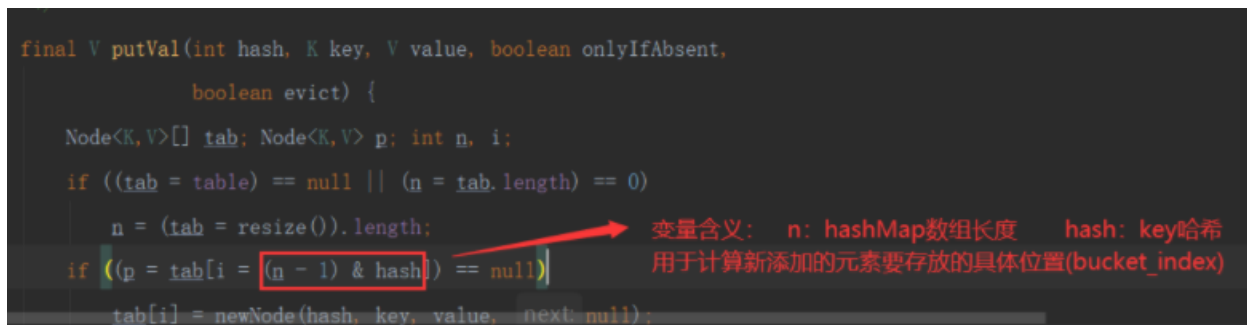
	优点	缺点	适用场景	案例
开放寻址法	1、数据存储在数组，可以有效利用空间	1、删除需要特殊标记已删除数据	1、数据量小 2、存储空间小	Java的 ThreadLocal

数组	组，可以有双利用CPU缓存加速查询速度 2、序列化简单	数据 2、所有数据存储在一个数组，发生冲突时，解决的代价更高，造成装载因子不能太大，使得更加浪费内存空间	2、装载因子小	ThreadLocalMap
链表法	1、内存利用率高，需要时再申请 2、对大装载因子容忍度高，可大于1	1、因为链表需要存储指针，存储指针需要消耗内存，不适合小对象存储（万一存储数据没有指针大??? 多浪费） 2、链表节点不是连续空间，因此CPU缓存不友好	1、存储大对象、大数据量的散列表 2、支持更多优化策略，如红黑树代替链表。	Java的LinkedHashMap

为什么数组要是二次幂？

https://blog.csdn.net/qq_39914581/article/details/85256935

https://blog.csdn.net/weixin_44421311/article/details/109250410



通过源码我们可以看到，HashMap新添加的元素是通过 $((\text{数组长度} - 1) \& \text{key的hashCode})$ 取模运算来计算槽位的（也就是新元素需要放在数组的哪个下标位置）取模按位与运算，减少冲突。

因此让数组的长度等于二次幂可以有效的减少hash冲突的概率。

怎么扩容的？扩容rehash的流程（jdk 1.7）

-单线程 -多线程

<https://www.jianshu.com/p/13c650a25ed3>

<https://www.cnblogs.com/williamjie/p/9360370.html>

头插-尾插

<https://zhuanlan.zhihu.com/p/342061904>

[算法题](#) coins硬币 / 一句都没写出来，做了20min白卷交的，（我：该用[动态规划](#)做，但不知道怎么下手了，面试官：转移方程怎么写？我：对不起，忘记了）