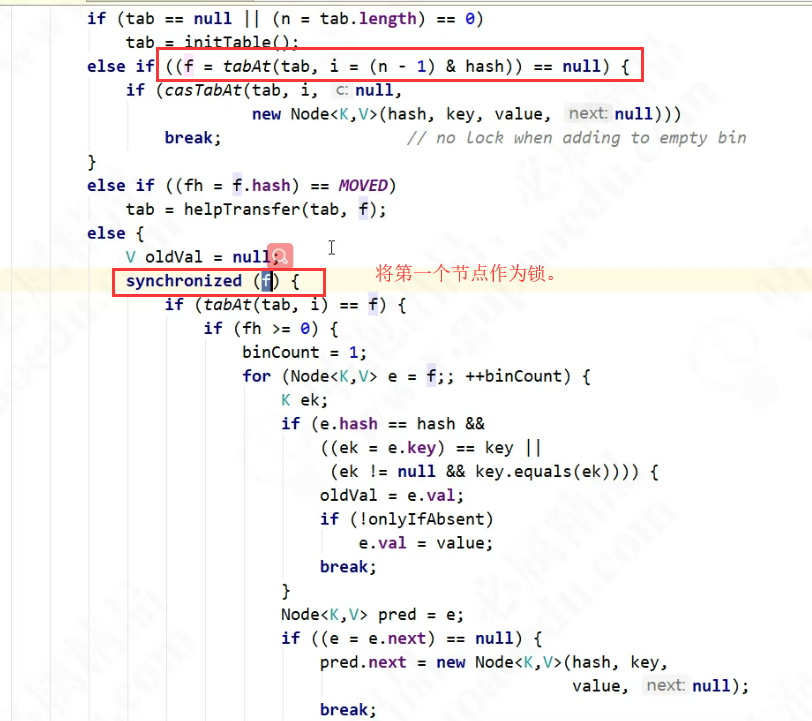


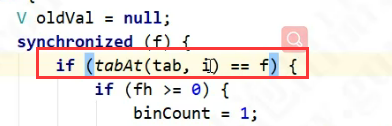
Volatile

地址是最新的，但是无法保证该地址出的元素值是最新的。

面试经常问

将锁的粒度锁定在第一个节点，其他线程都可以访问第一个节点，而无法访问被锁住的链表或红黑树。第一个节点其实就是数组中的元素。

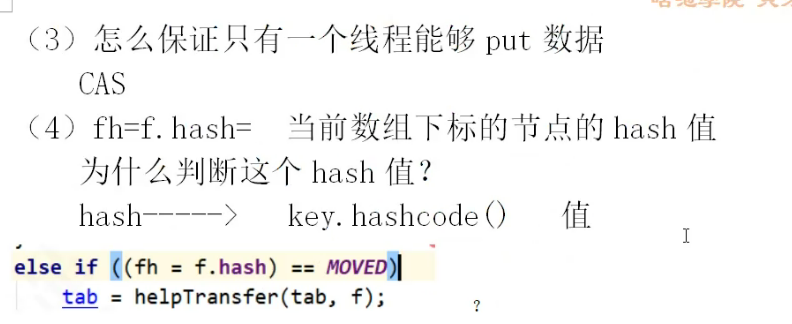




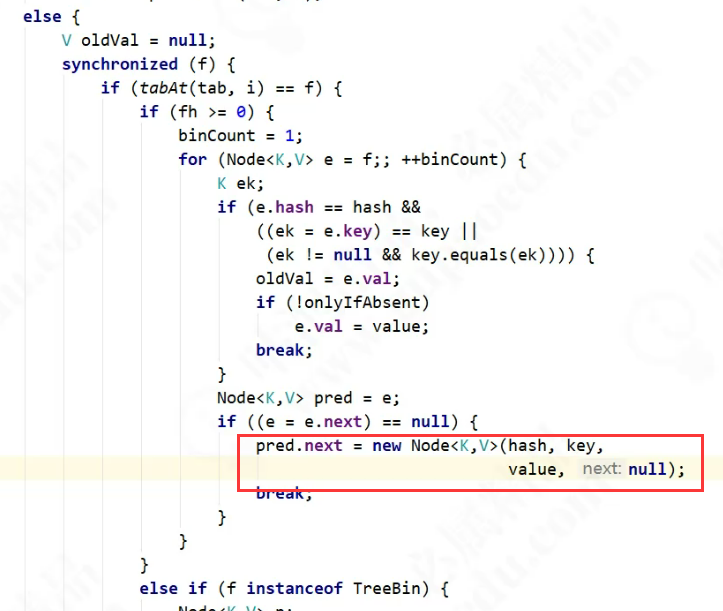
有点像Double Check的味道：

注意区分分段锁：

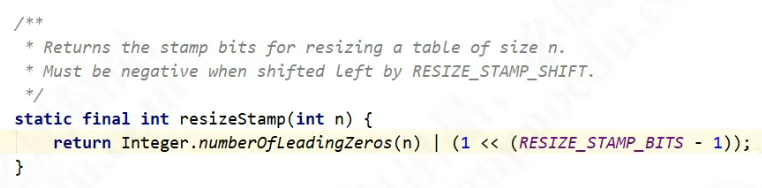
每一个分段锁对应的下面都是一个一个hashMap。

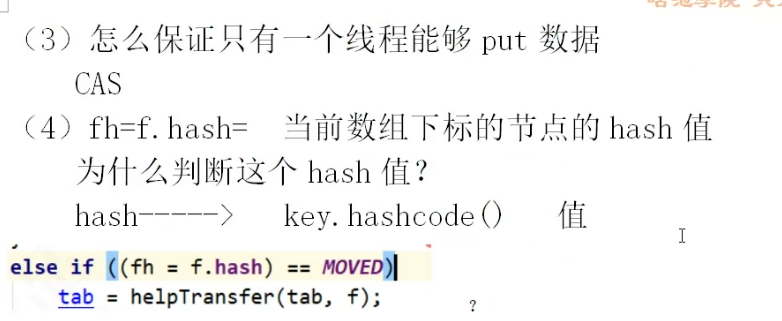


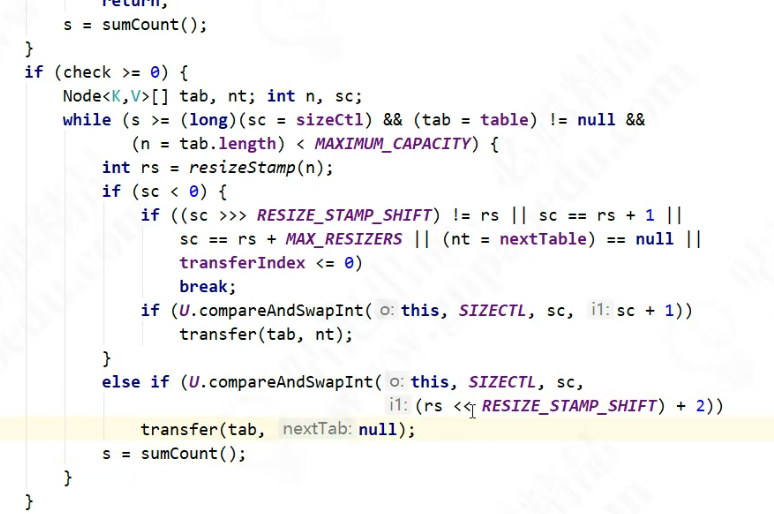
Next时候不需要再CAS，因为该链表 已经是加锁了。



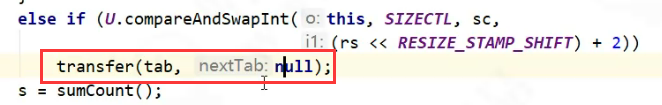
扩容：

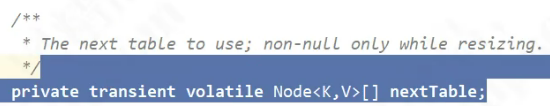






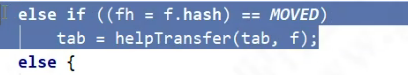
Transfer 扩容。



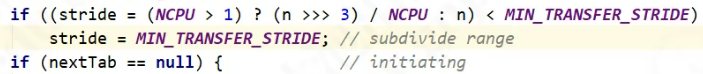


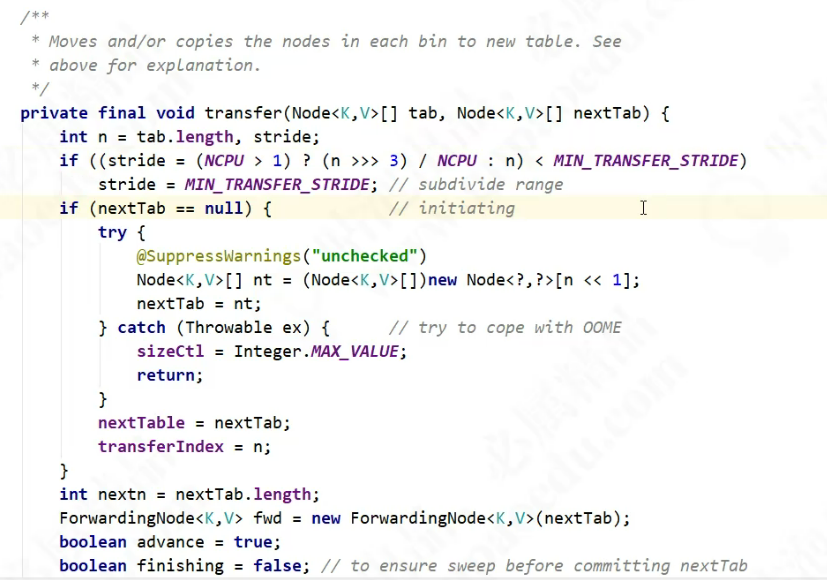


等于 **负一** ：

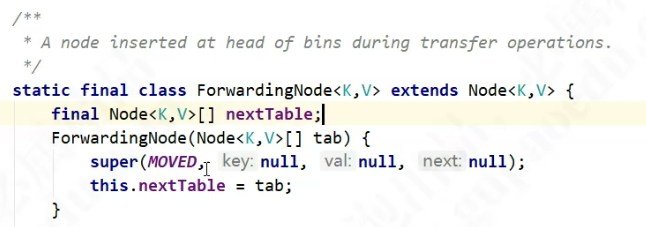


根据CPU的数目进行：



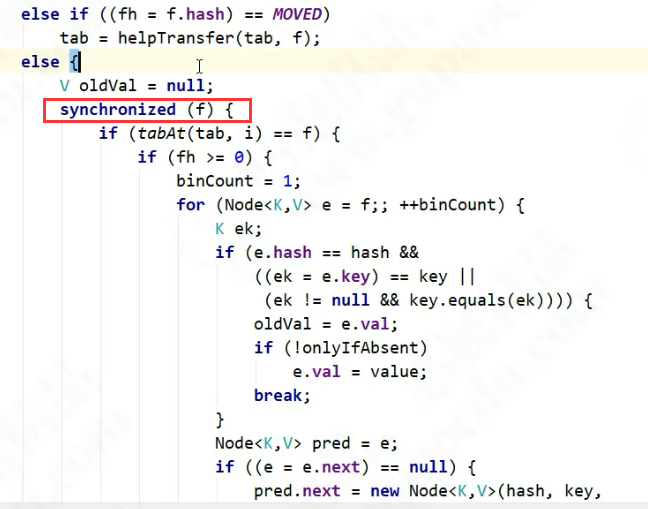


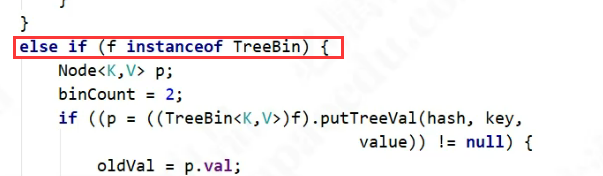




再拷贝时，需要synchronized加锁。

分成两种情况：链表复制和红黑树复制。

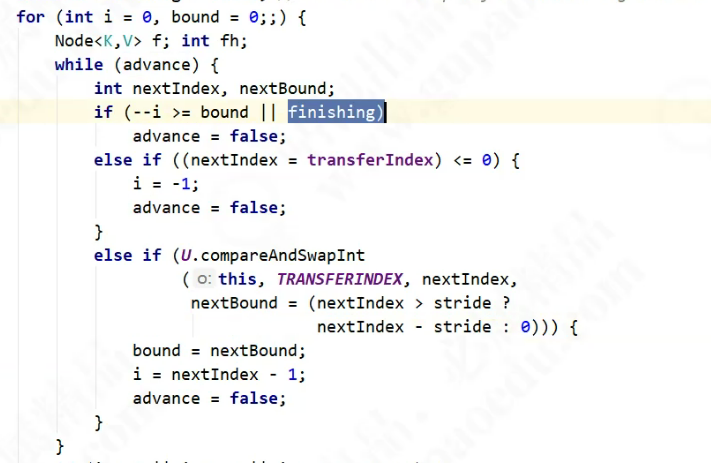


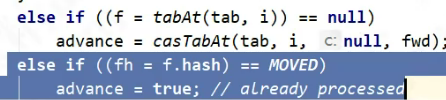


最难理解的还是扩容：

让多个线程一起参与扩容。

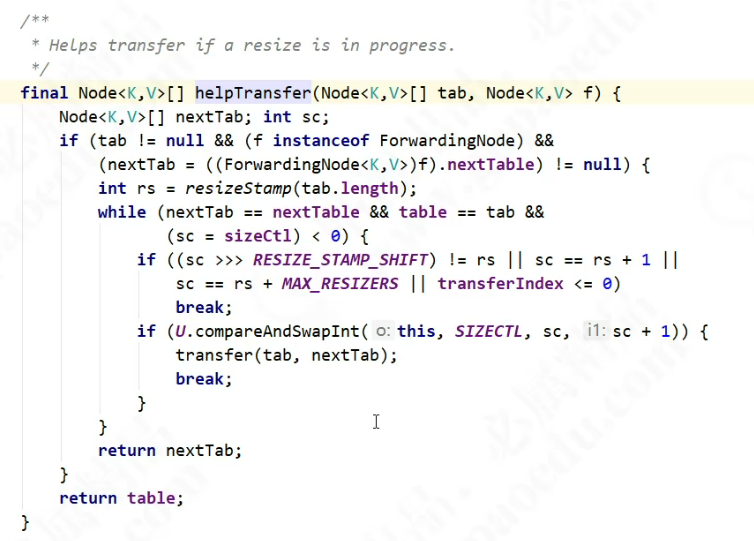
对i不断进行减减操作：如从15逐渐减减至头节点 。



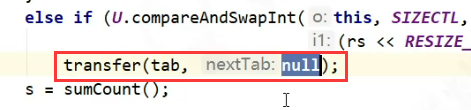


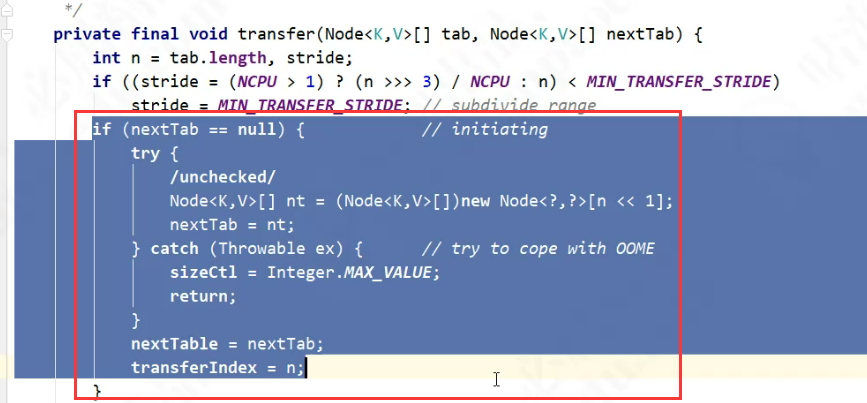
处理完毕。

HelpTransfer方法中也调用了transfer方法。



正常情况下：扩容传递的是null，进行2倍初始化。



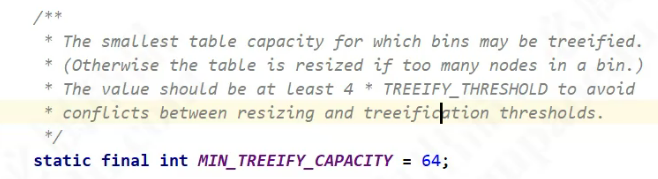


帮助扩容的时候：传递的不是null，不会执行上面的if语句:初始化。

**扩容的三种情况：**

1. **容量达到指定size**
2. **Put操作时，帮助扩容；helpTransfer**
3. **链表转红黑树的时候，当节点数量不超过64时，不能转成红黑树，而是要扩容。**

并不是当链表的长度超过8，都会转化红黑树，只有当总数大于64.



**仔细品味while循环和for循环的味道。**