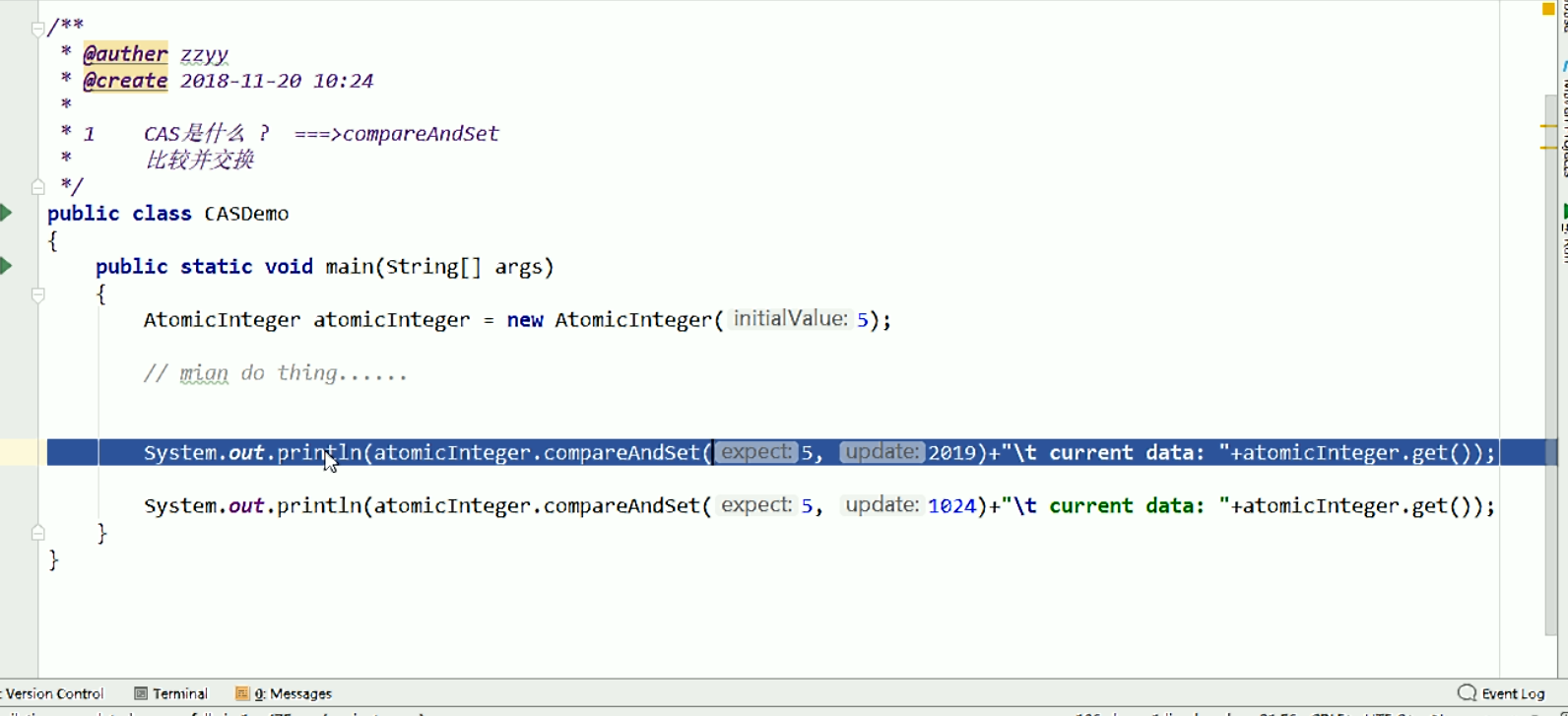
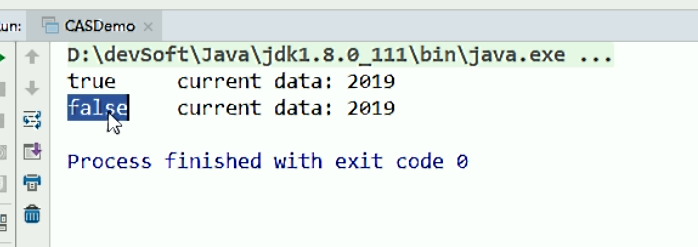
AtomicInteger调用里面的compareAndSet(期望值 , 修改值) 方法



结果为



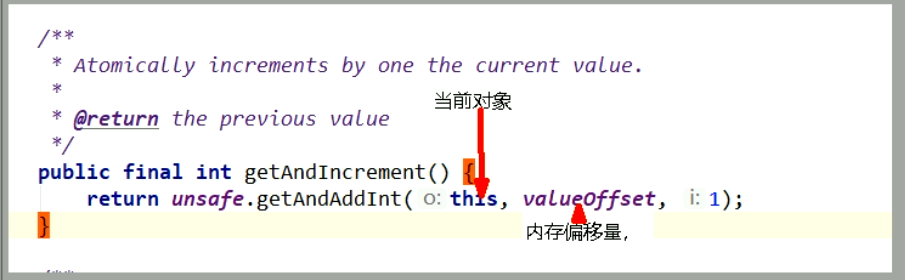
CompareAndSet方法的两个参数(期望值,更新值)

期望值必须和主内存里面的值相同,才可以更改写回到主内存

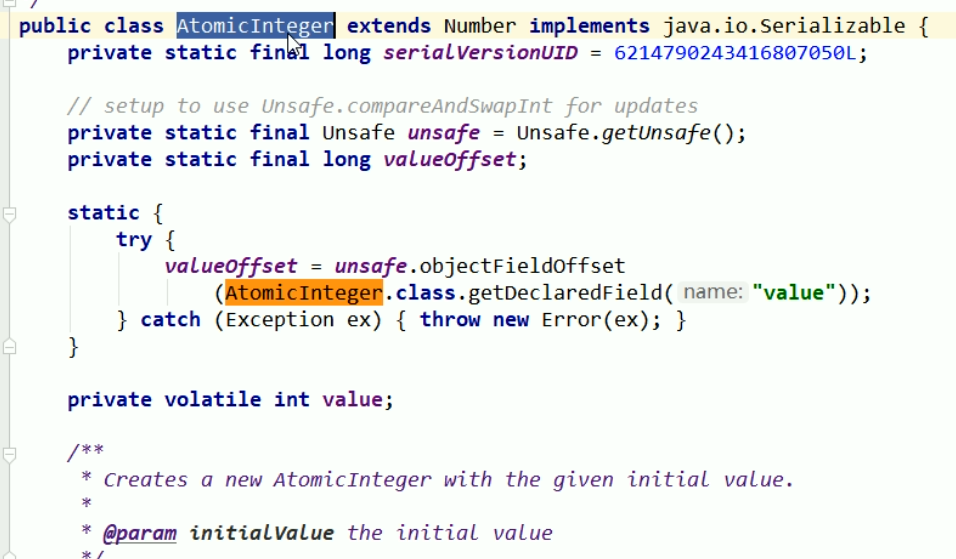
如果线程的期望值和物理内存的真实值一样,返回true,修改物理内存的值

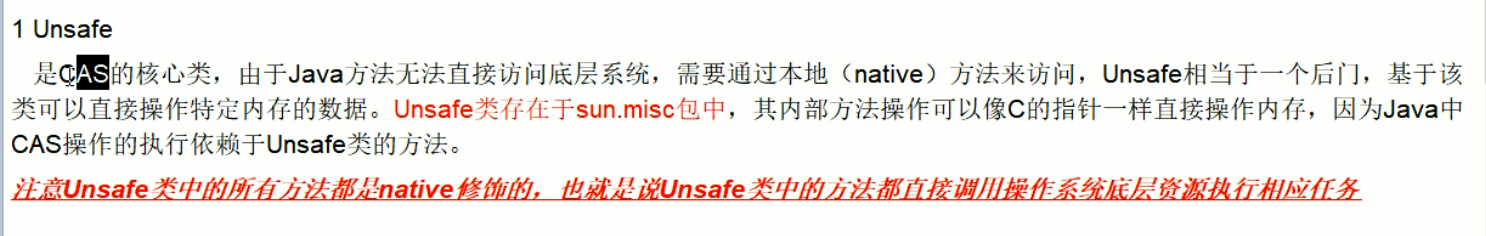
如果线程的期望值和物理内存的真实值不一样,返回false,不可以修改物理内存的值

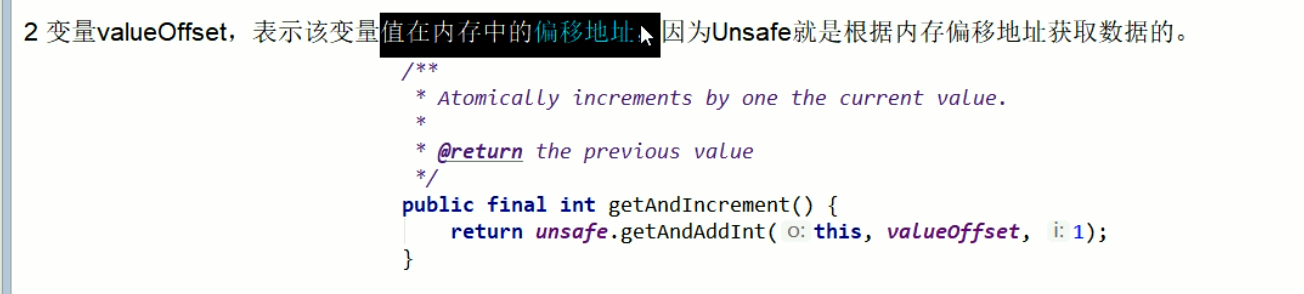
方法的底层是这样的



内存偏移量 通过 这个 unsafe.objectFieldoffset

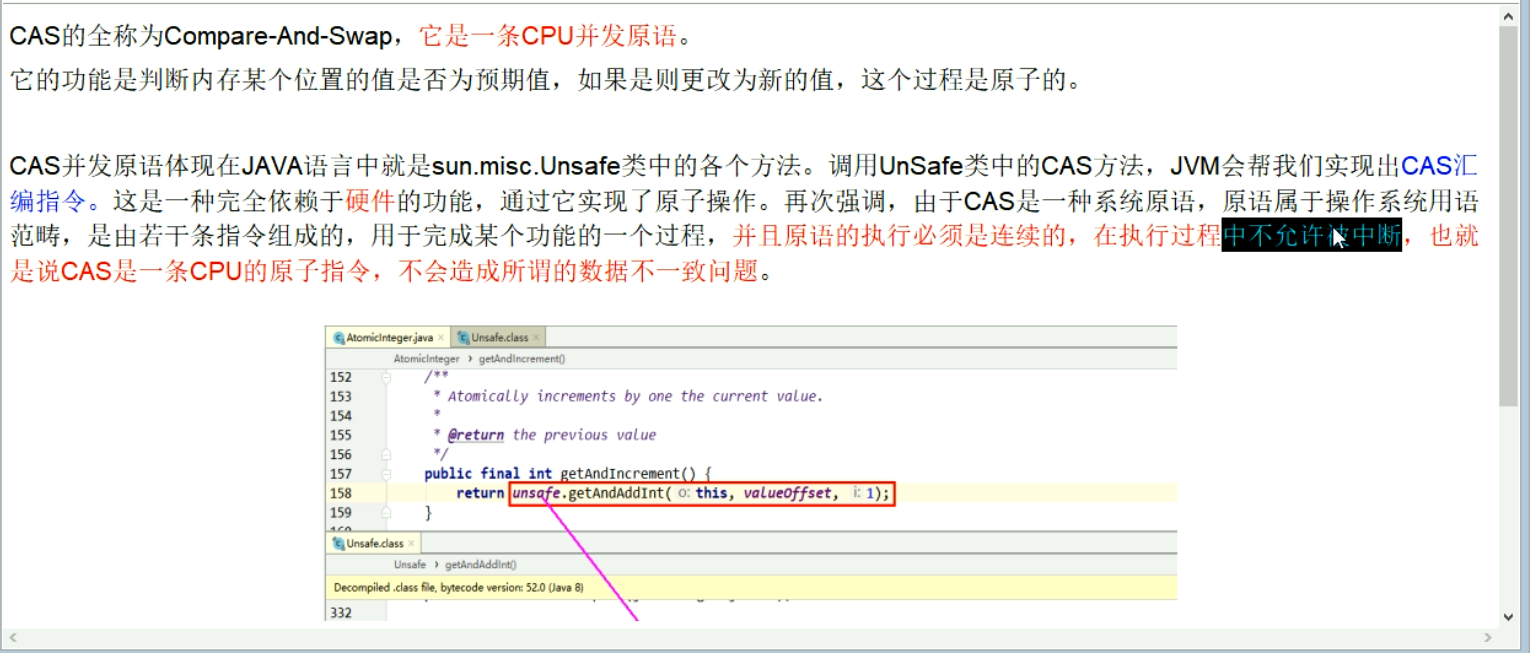


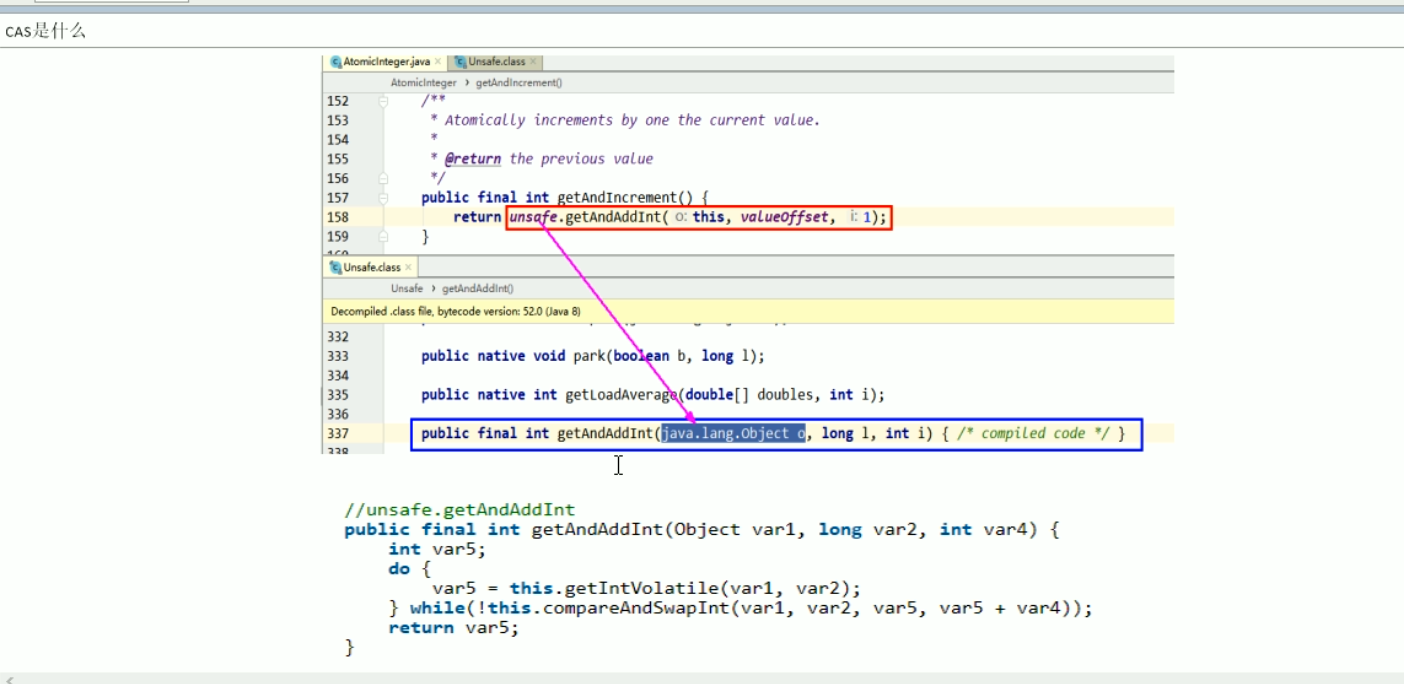




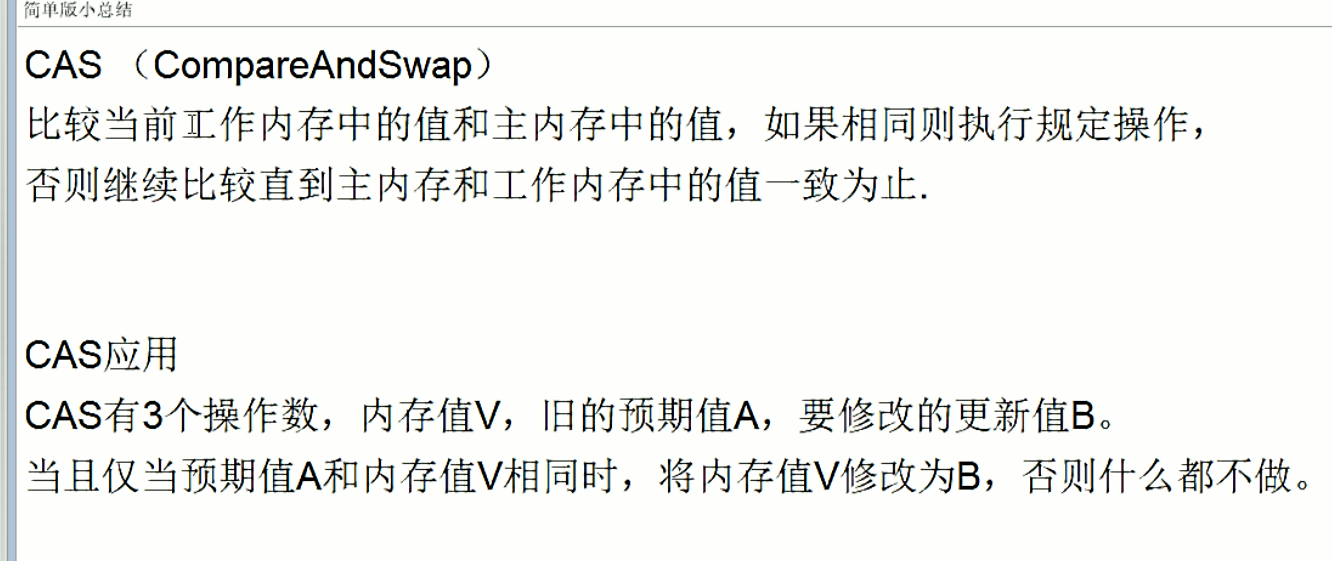
保证原子性靠的是底层unsafe类 ,unsafe来自jvm的rt.jar运行架包里面

Unsafe可以根据内存偏移地址获取数据

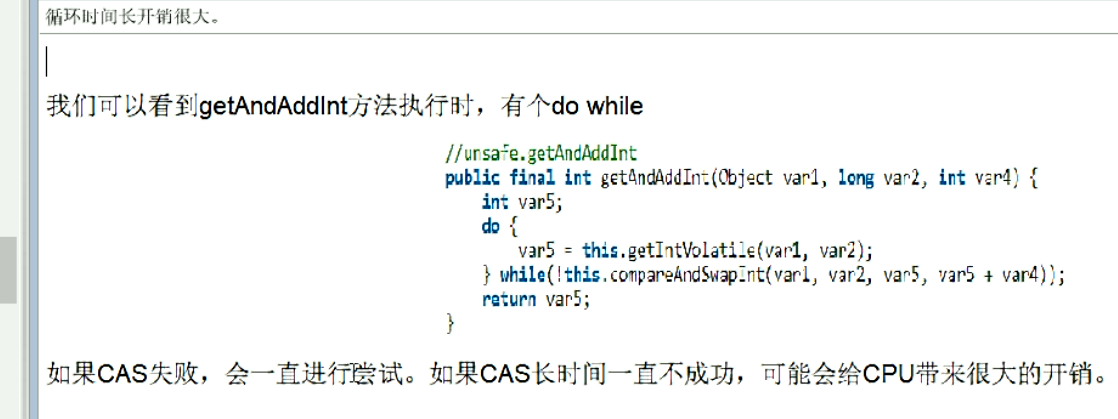




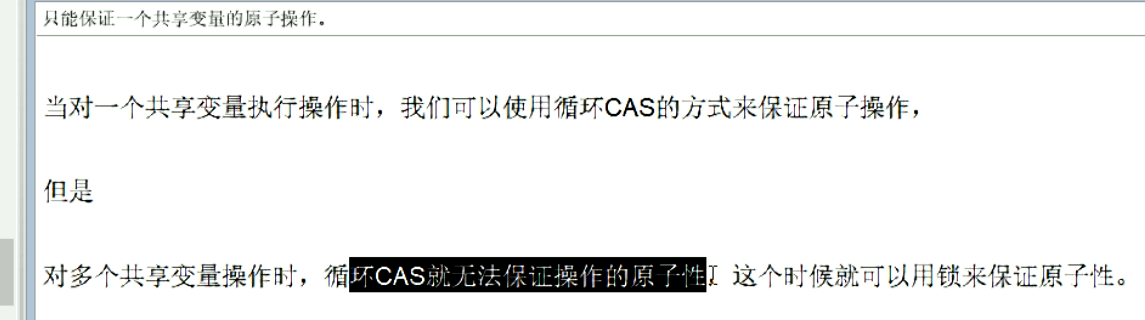
总结:cas自旋思想



Cas缺点

底层是一个do while.如果一直不同,会一直循环执行,开销大

第二点



第三点aba问题