

CAS:compare and swap

比较并操作，cas的语义是“我认为V（内存中）的值应该为A（原值），如果是，那么将V的值更新为B（线程需要更新A为B），否则不修改并告诉V的值实际是多少”这句话的意思就是如果内存中的值和原值不一样，则不对其进行修改，CAS是一个乐观锁的技术，当多个线程尝试使用CAS同时更新同一个变量时，只有其中一个线程能更新变量的值，而其他线程都失败，失败的线程并不会挂起，而是被告知这次竞争失败，并可以再次尝试。CAS有3个操作数，内存值V、旧的预期值A、要修改的新值B。当且仅当预期值A和内存值V相同时，将内存值V修改为B，否则什么都不做。

CAS包含三个参数（V, E, N）

V表示要更新的变量，E表示预期值，N表示新值，如果V值等于E值，则将V的值设为N。若V值和E值不同，则说明已经有了其他线程做了更新，则当前线程什么都不做，通俗的理解就是CAS操作需要我们提供一个期望值，当前期望值与当前线程的变量相同时，说明还没线程修改改值，当前线程可以进行修改，也就是执行CAS操作，但如果期望值与当前线程不符，则说明改值已被其他线程修改，此时不执行更新操作，但可以选择重新读取该变量再尝试修改变量，也可以放弃操作。

独占锁是一种悲观锁，synchronized就是一种独占锁，它假设最坏的情况，并且只有在确保其他线程不会造成干扰的情况下执行，会导致其他所有需要锁的线程挂起，等待持有锁的线程释放锁。而另一种更加有效的锁就是乐观锁。所谓乐观锁就是，每次不加锁而是假设没有冲突而去完成某项操作，如果有冲突就重试，直到成功为止。