

CAS机制当中使用了3个基本操作数：内存地址V，旧的预期值A，要修改的新值B。

举例：

1. 在内存地址V当中，存储着值为10的变量。



内存地址V

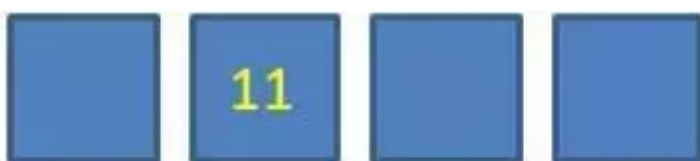
1. 此时线程1想要把变量的值增加1。对线程1来说，旧的预期值A=10，要修改的新值B=11。



内存地址V

线程1: A = 10 B = 11

1. 在线程1要提交更新之前，另一个线程2抢先一步，把内存地址V中的变量值率先更新成了11。

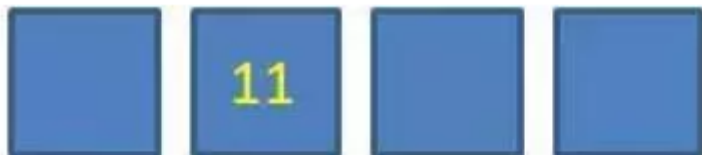


内存地址V

线程1: A = 10 B = 11

线程2: 把变量值更新为11

1. 线程1开始提交更新，首先进行A和地址V的实际值比较（Compare），发现A不等于V的实际值，提交失败。



内存地址V

线程1: $A = 10$ $B = 11$

$A \neq V$ 的值 ($10 \neq 11$)

提交失败!

线程2: 把变量值更新为11

1. 线程1重新获取内存地址V的当前值, 并重新计算想要修改的新值。此时对线程1来说, $A=11$, $B=12$ 。这个重新尝试的过程被称为自旋。



内存地址V

线程1: $A = 11$ $B = 12$

1. 这一次比较幸运, 没有其他线程改变地址V的值。线程1进行Compare, 发现A和地址V的实际值是相等的。



内存地址V

线程1: $A = 11$ $B = 12$

$A == V$ 的值 ($11 == 11$)

1. 线程1进行SWAP, 把地址V的值替换为B, 也就是12。



内存地址V

线程1: A = 11 B = 12
A == V的值 (11 == 11)
地址V的值更新为12