第十五章 原子变量与非阻塞同步机制

1. 非阻塞算法

用底层的原子机器指令代替锁来确保数据在并发访问中的一致性。

2. 优缺点

优点：有巨大的可伸缩性和活跃性优势

缺点：设计和实现都复杂的多

#### 一．锁的劣势

1. 在挂起和恢复线程等过程中存在着很大的开销，并且通常存在着较长的时间中断。

2. 激烈竞争的锁，调度开销与工作开销的比值会非常高。

3. volatile虽然提供了与锁相似的可见性但是并没有提供原子性。

4. 当一个线程在等待锁的时候不能做任何事情。

5. 死锁，优先级反转。

#### 二．硬件对并发的支持

1. 比较并交换（CAS）

原值：V

期望比较值：A

写入值：B

（1）判断：如果V和A相等是否相等

（2）如果相等则把V替换为B，否则不进行任何操作

（3）返回V原有的值

2. 特点

（1）底层CPU支持比较并交换（Compare And Swap）指令，可以原子性地进行执行取值-判断-写入，保证同时执行的线程只有一个线程能够成功，其余的失败。

（2）JVM将CAS操作直接编译成机器指令，提高运行速度

（3）一次加锁机制比一次CAS执行的指令更多，且可能发生线程调度，开销更大，所以在竞争不激烈的情况下，CAS比加锁开销更小

（4）CAS将对资源的竞争处理交由调用者来处理，所以使用起来更复杂。

#### 三．原子变量类

1. 共有12个原子变量类，可分为4组：标量类（Scalar）、更新器类、数组类、以及复合变量类。

2. 最常用的标量类：AtomicInteger、AtomicLong、AtomicBoolean、AtomicReference

3. 原子变量是一种“更好的volatile”

4. 性能比较：锁与原子变量

（1）在高度竞争情况下，锁的性能更好；在更真实的情况下，原子变量的性能更好。

（2）实际情况中，原子变量在可伸缩性上要高于锁，因为常见的竞争程度原子变量的效率更高。

#### 四．非阻塞算法

1. 非阻塞算法：

一个线程的失败或挂起不会导致其他线程也失败或挂起。

2. 无锁算法：

在算法的每个步骤中都存在某个线程能够执行下去。

3. 非阻塞栈

4. 非阻塞链表

5. ABA问题

A变为B，然后B又变为A，有可能误认为没有发生改变。

解决办法：增加版本号。