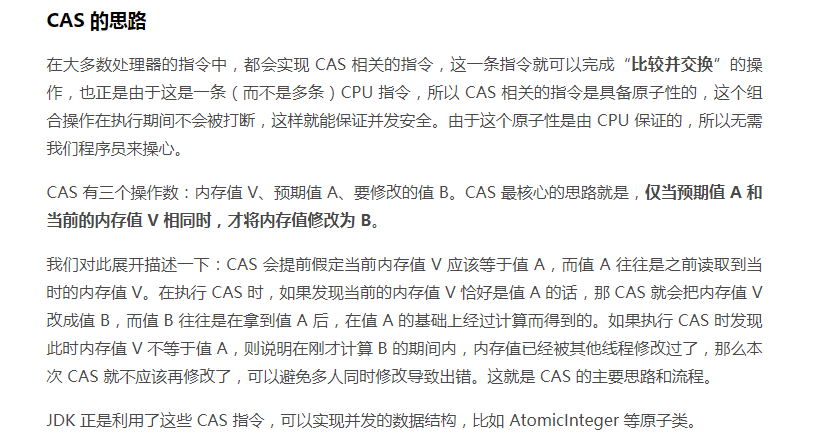
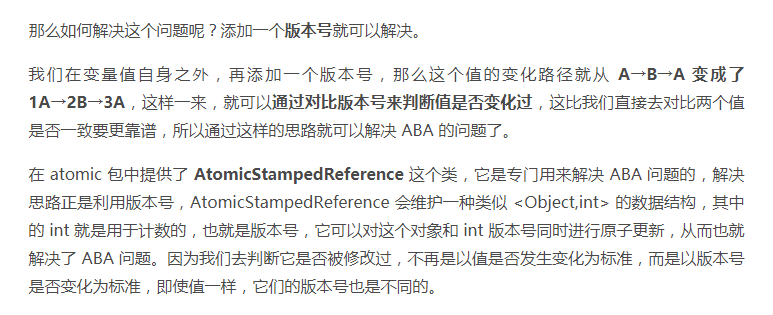
1. **CAS简介**



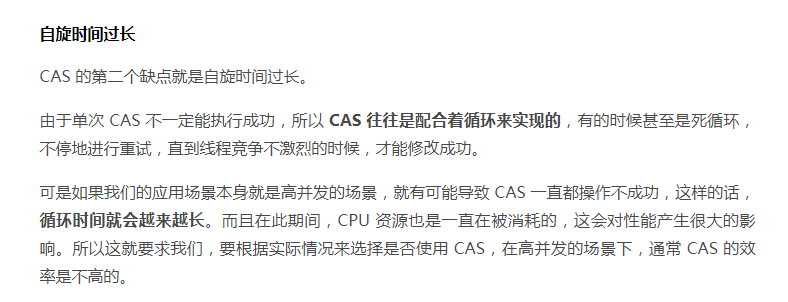
1. **CAS思路**



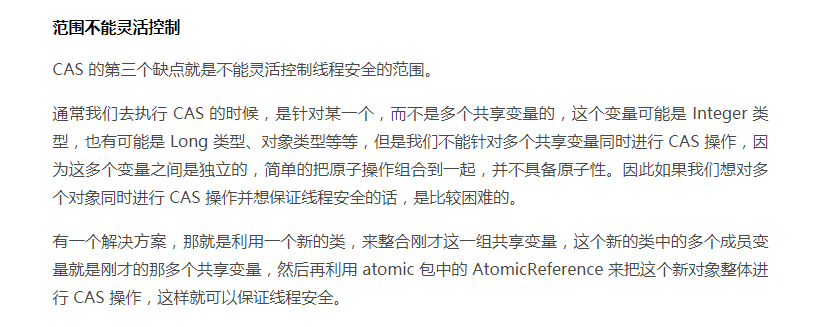
1. **CAS缺点**
2. ABA问题



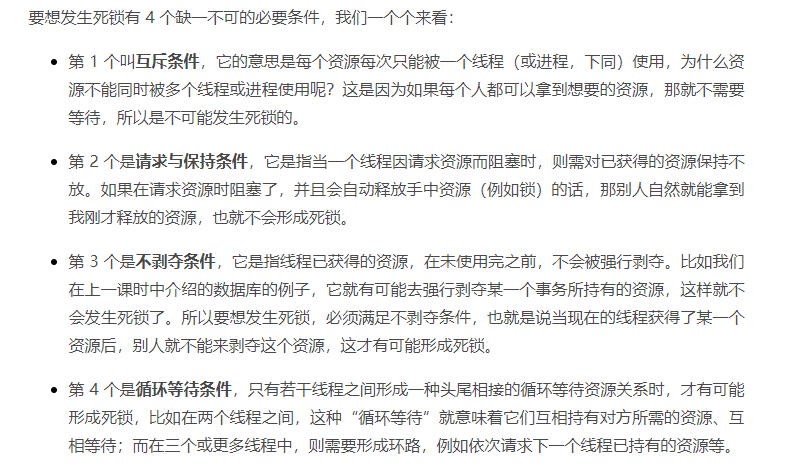
1. 自旋时间过长



1. 范围不能灵活控制



1. **发生死锁的 4 个必要条件**

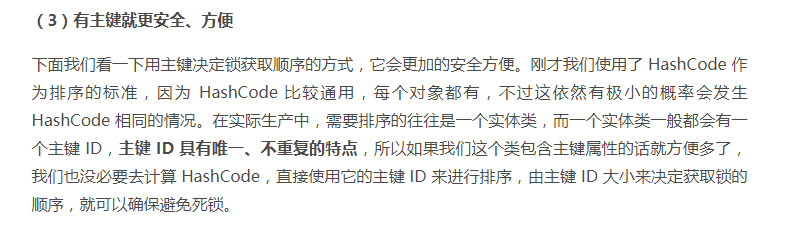


1. **避免死锁策略**

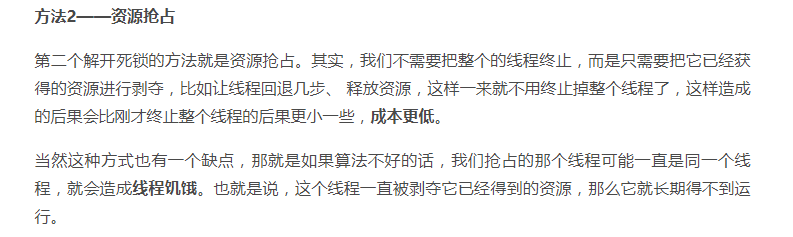
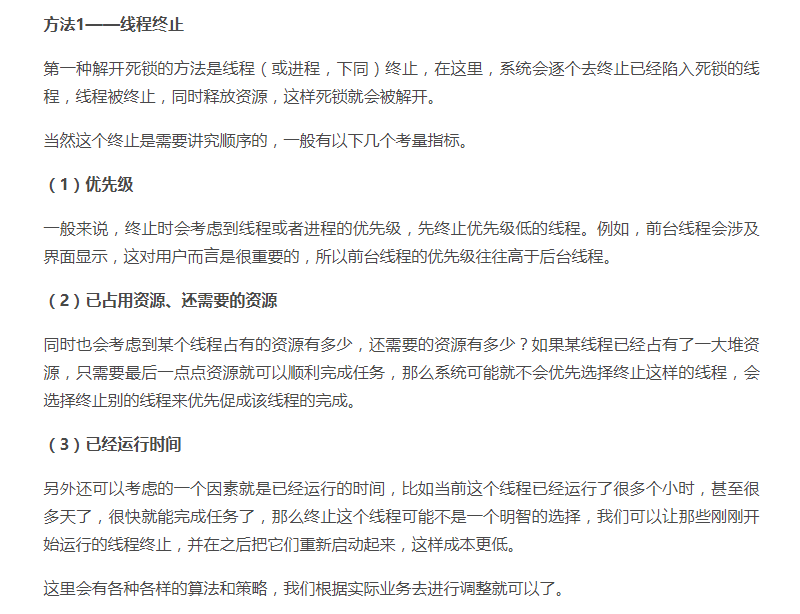
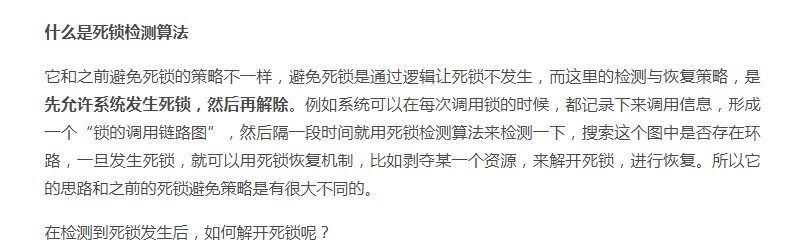
1、hashCode方式



2、主键方式



1. **检测与恢复死锁策略**



1. **哲学家就餐问题解决方案**

public static void main(String[] args) {

    Philosopher[] philosophers = new Philosopher[5];

    Object[] chopsticks = new Object[philosophers.length];

    for (int i = 0; i < chopsticks.length; i++) {

        chopsticks[i] = new Object();

    }

    for (int i = 0; i < philosophers.length; i++) {

        Object leftChopstick = chopsticks[i];

        Object rightChopstick = chopsticks[(i + 1) % chopsticks.length];

        if (i == philosophers.length - 1) {

            philosophers[i] = new Philosopher(rightChopstick, leftChopstick);

        } else {

            philosophers[i] = new Philosopher(leftChopstick, rightChopstick);

        }

        new Thread(philosophers[i], "哲学家" + (i + 1) + "号").start();

    }

}

1. **String 不可变性的好处**

