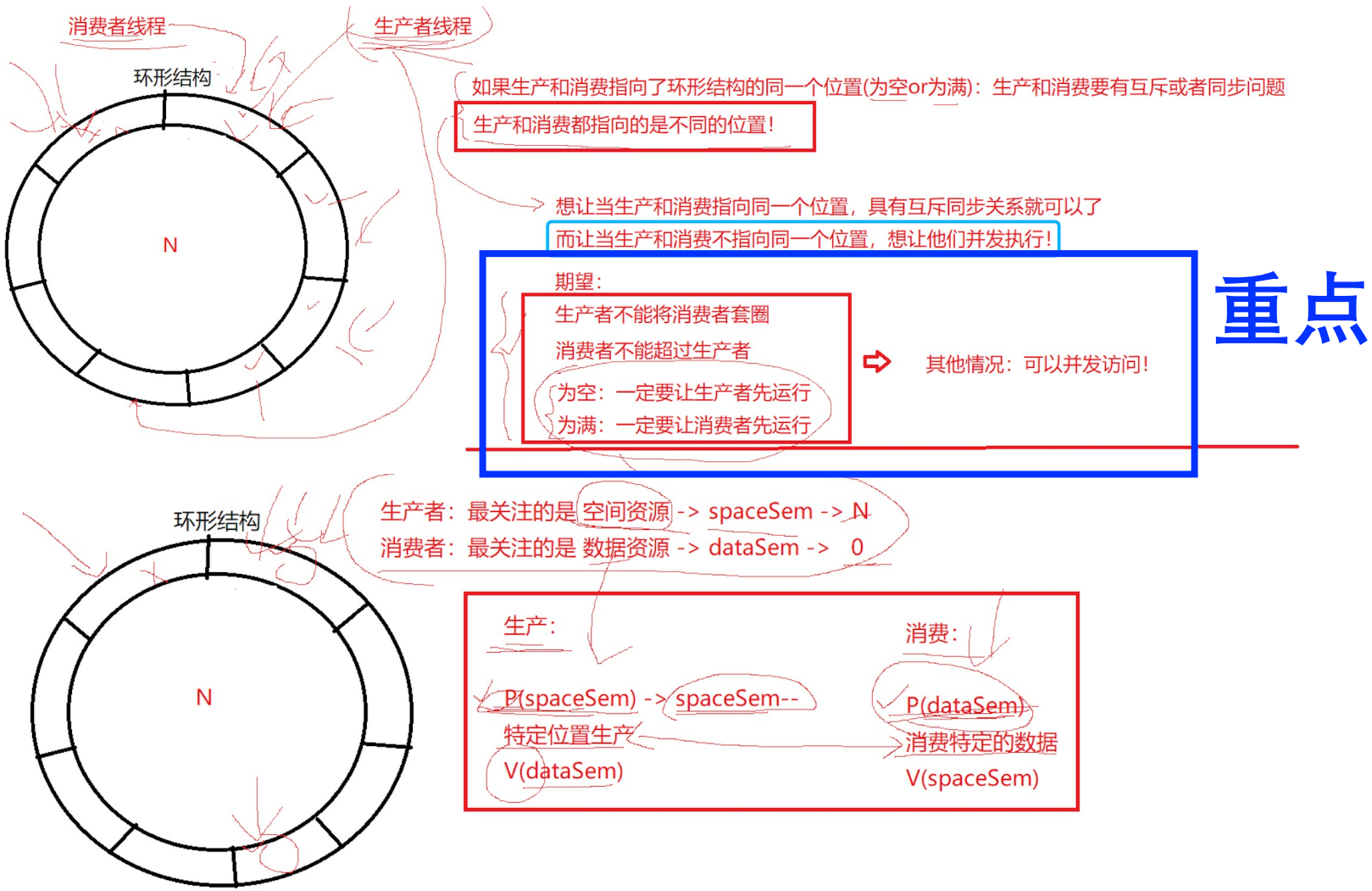
0208信号量+线程池

基于环形队列的生产消费模型

- 环形队列采用数组模拟，用模运算来模拟环状特性



当信号量是二元信号量的时候就相当于互斥锁

```
void *consumer(void *args)
{
    RingQueue<int> *rq = (RingQueue<int> *)args;
    while(true)
    {
        int x = 0;
        rq->pop(&x);
        std::cout << "consumer: " << x << std::endl;
        sleep(1);
    }
}
```

如果让消费者慢一点
我们最后看到的结果就应该是

一瞬间，队列被打满
然后生产一个消费一个

因为队列满了之后，生产者就会被挂起
直到有位置了，才能去生产

不得不考虑的一个问题：
先加锁还是先申请信号量？

先申请信号量效率更高！
这里看看视频的例子

例如：电影院入口只能通过一个人
如果我们先进入这个门口，再买票，效率就很低了！
所以，我们应该现在外面买了票，再来进这个门口

结合信号量的概念和锁的概念，先申请信号量的原因就很清晰了

1. 多生产多消费的意义在哪里？

2. 信号量本质是一把计数器 -> 计数器的意义是什么？

计数器的意义在于：
可以让我们在临界区外部就能了解到，临界区内部的情况！甚至可以减少临界区内部的判断

线程池本质就是一个
生产者消费者模型

需要一个任务队列
然后把任务派发给线程们就行了