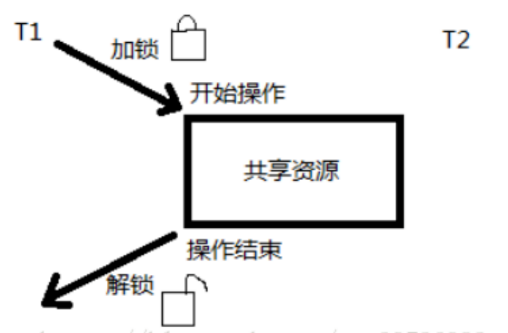
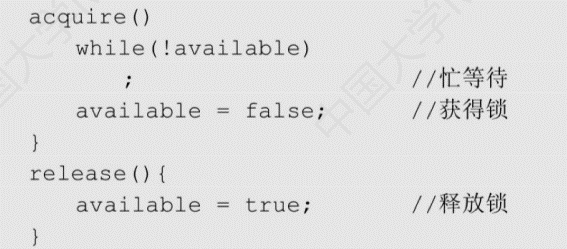
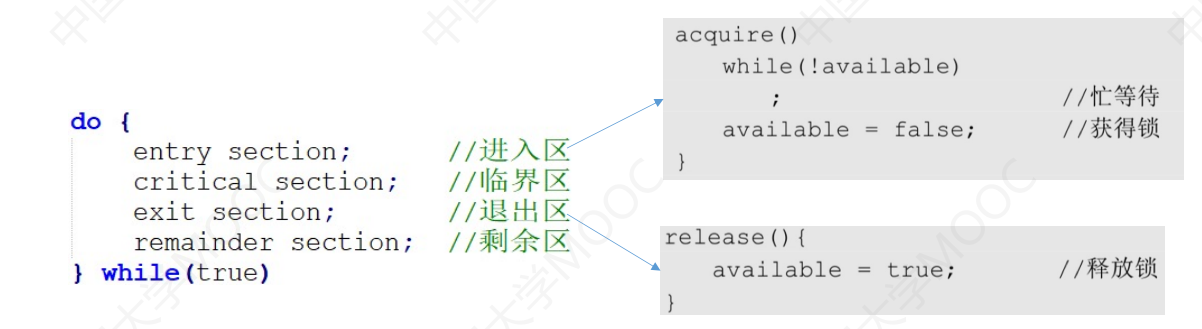
## 对与互斥锁问题

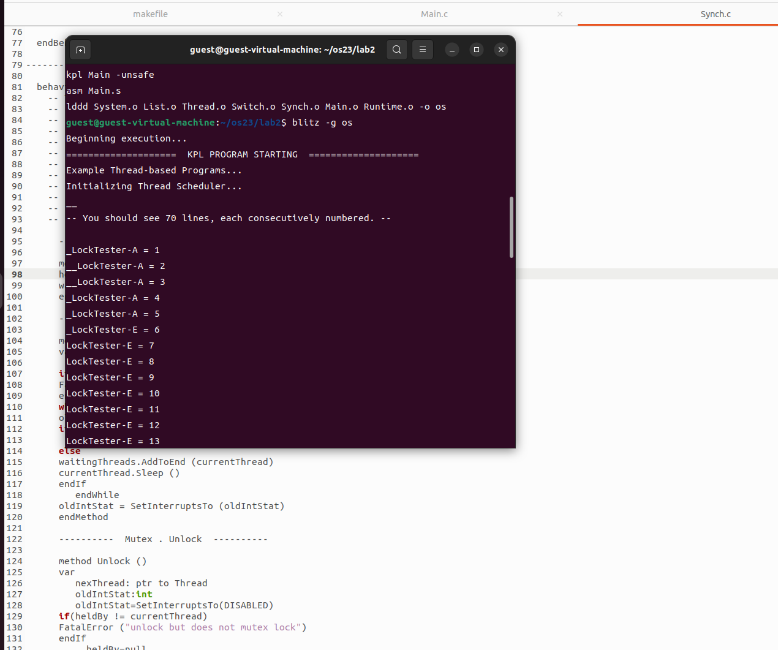
每个线程在对资源操作前都尝试先加锁，成功加锁才能操作，操作结束解锁。

但通过“锁”就将资源的访问变成互斥操作，而后与时间有关的错误也不会再产生了。

一些伪代码

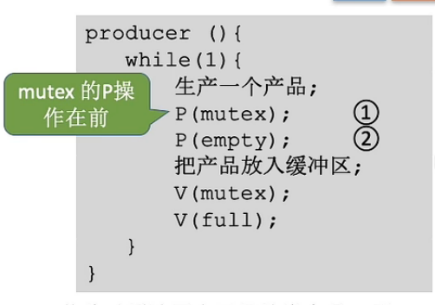


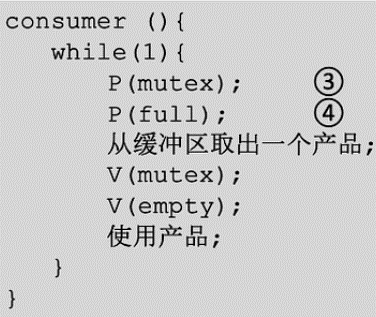


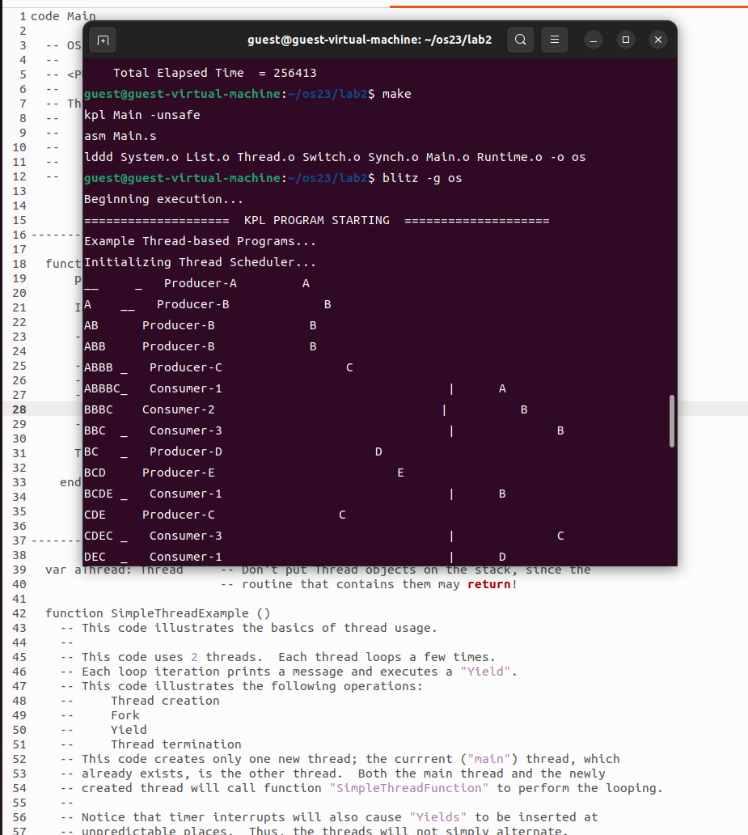


## 生产者与消费者问题

生产者消费者问题是一个多线程同步问题的经典案例。问题描述了共享固定大小缓冲区的两个线程——即所谓的“生产者”和“消费者”——在实际运行时会发生的问题。生产者的主要作用是生成一定量的数据放到缓冲区中，然后重复此过程。与此同时，消费者也在缓冲区消耗这些数据。该问题的关键就是要保证生产者不会在缓冲区满时加入数据，消费者也不会在缓冲区中空时消耗数据。







## 哲学家进晚餐问题

至多只允许四位哲学家同时去拿左筷子，最终能保证至少有一位哲学家能进餐，并在用完后释放两只筷子供他人使用。

