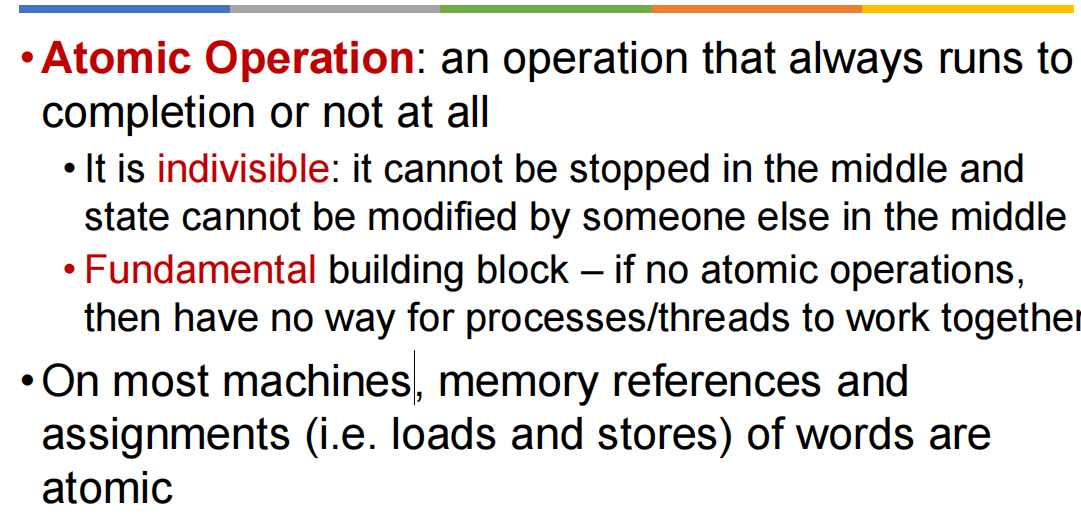
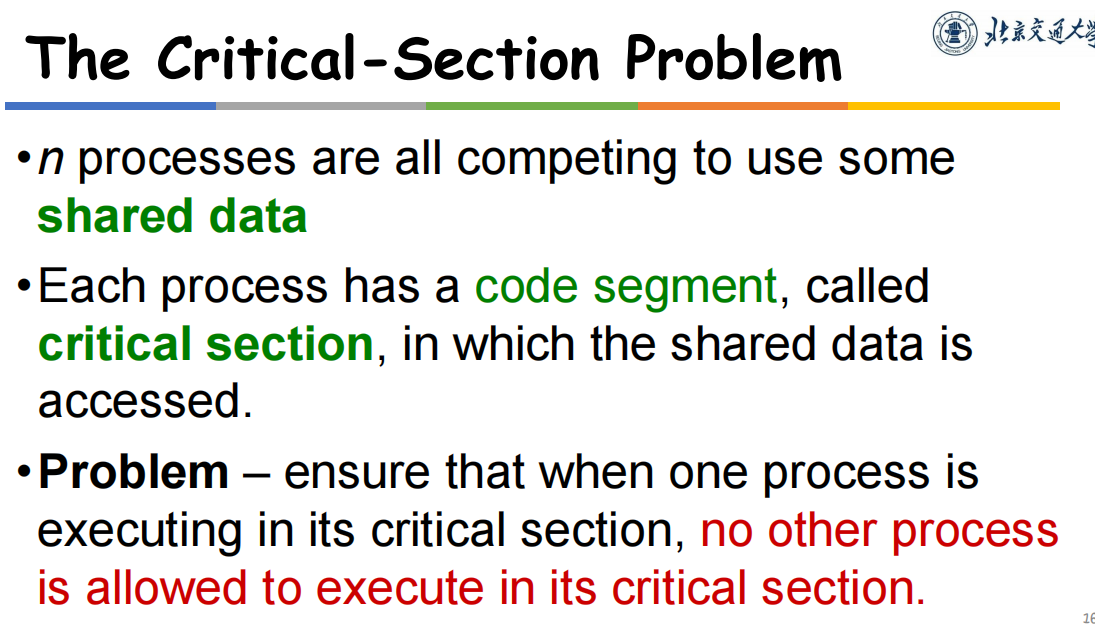


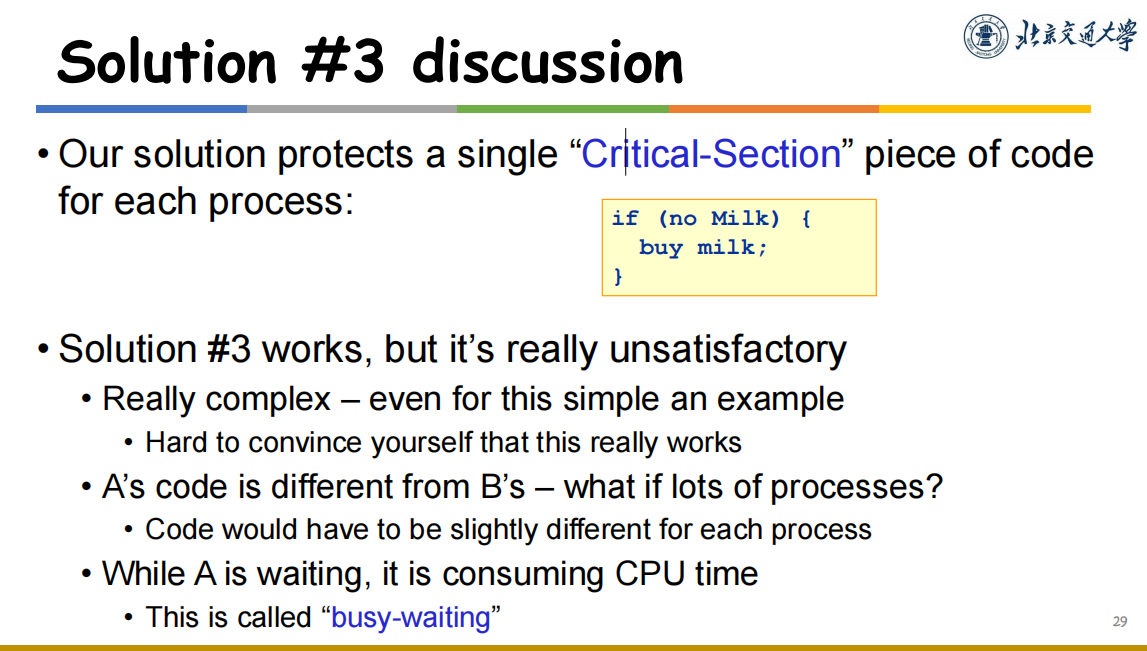
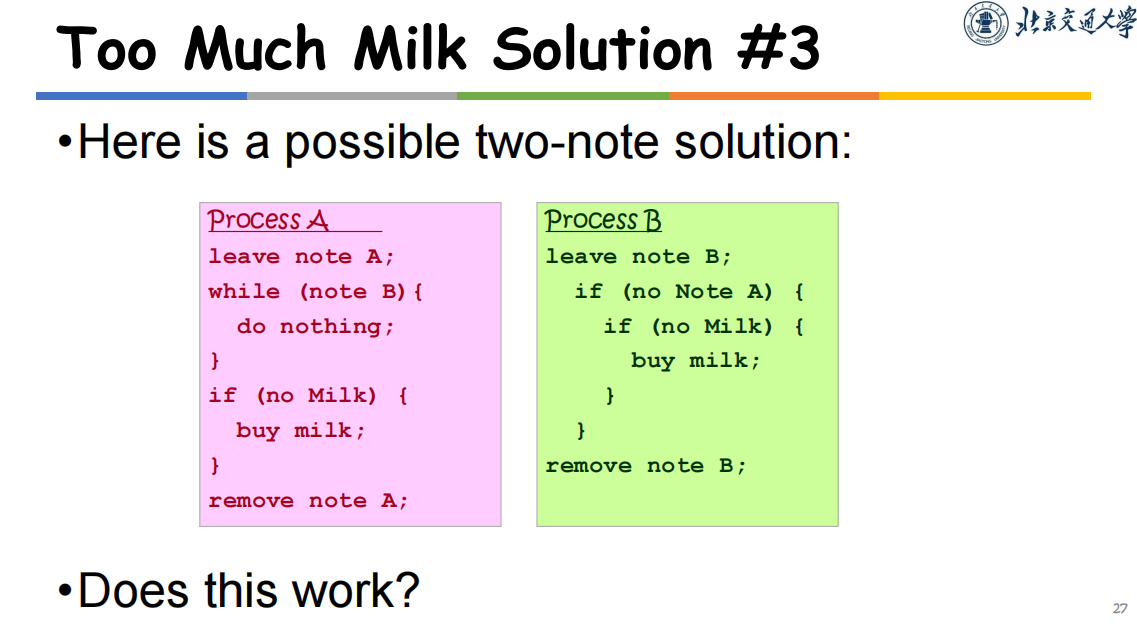
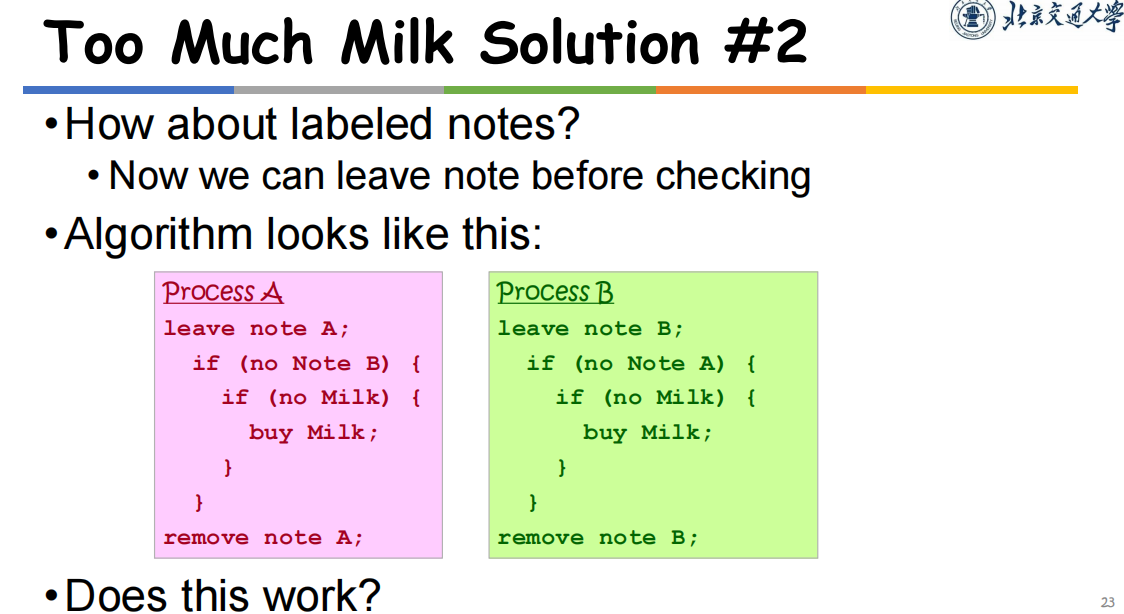
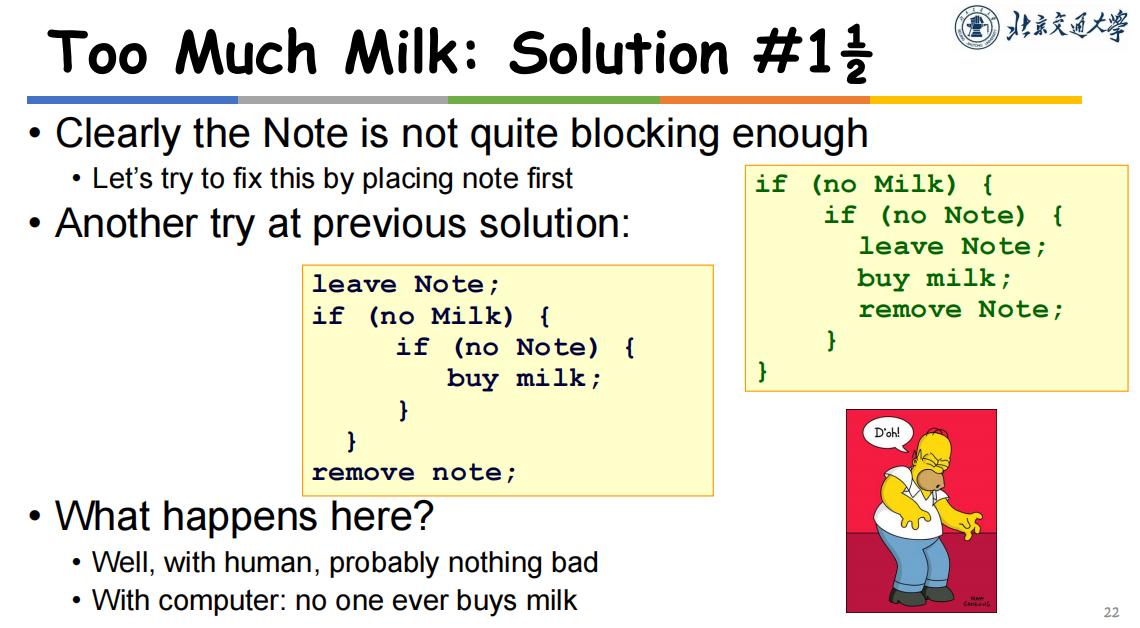
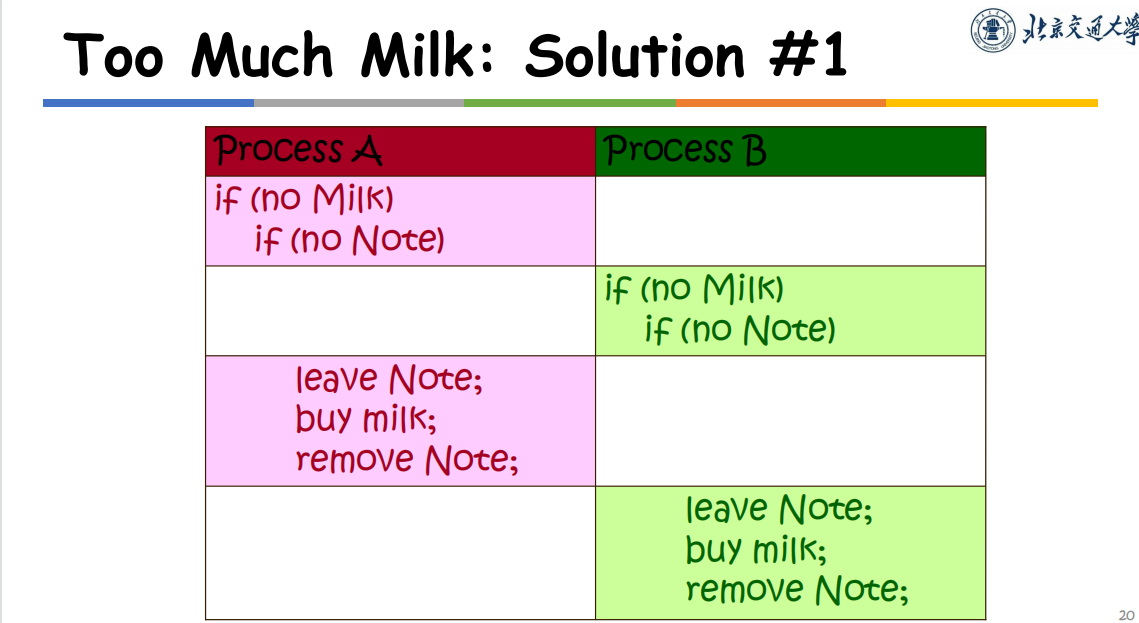
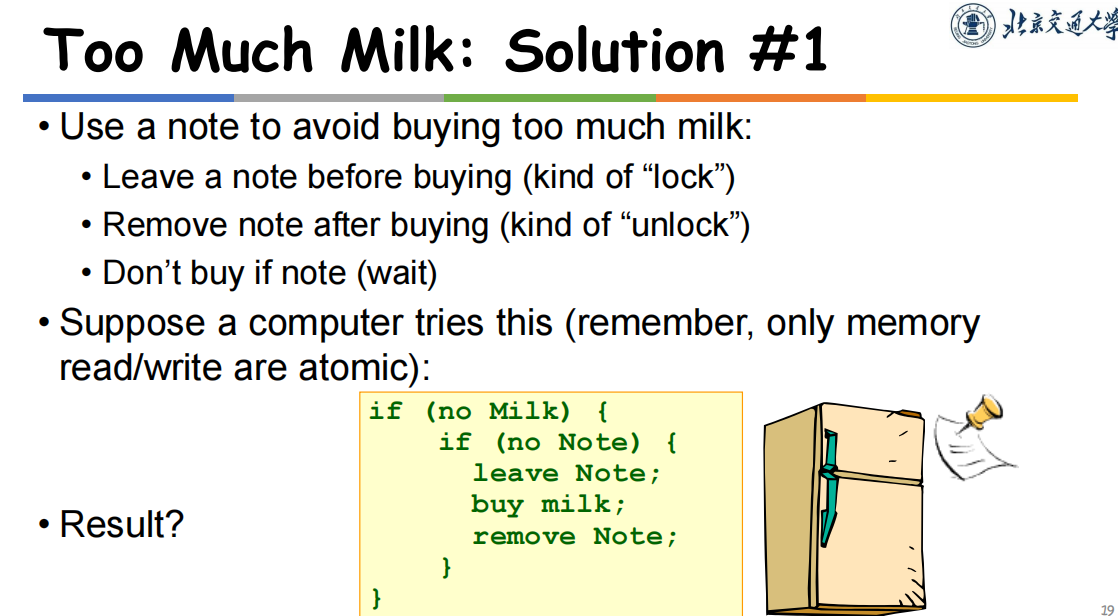
原子操作



互斥性 + 关键部分

将其定义为关键部分，并且划分。



我们的解决方案保护单个“关键部分”代码

对于每个过程：

解决方案3有效，但真的不令人满意

真的很复杂——即使是这个简单的例子

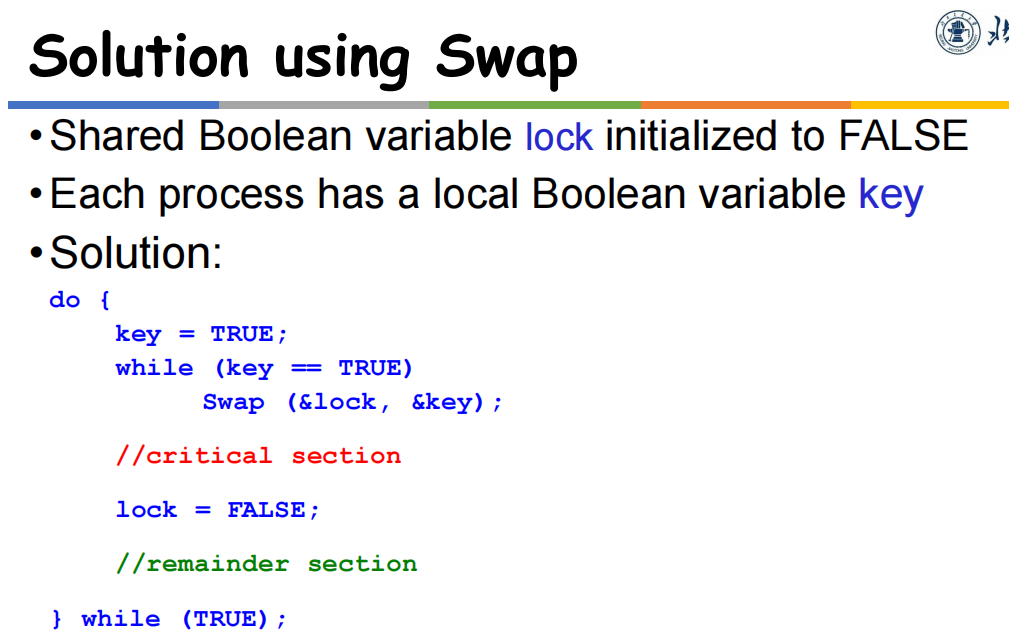
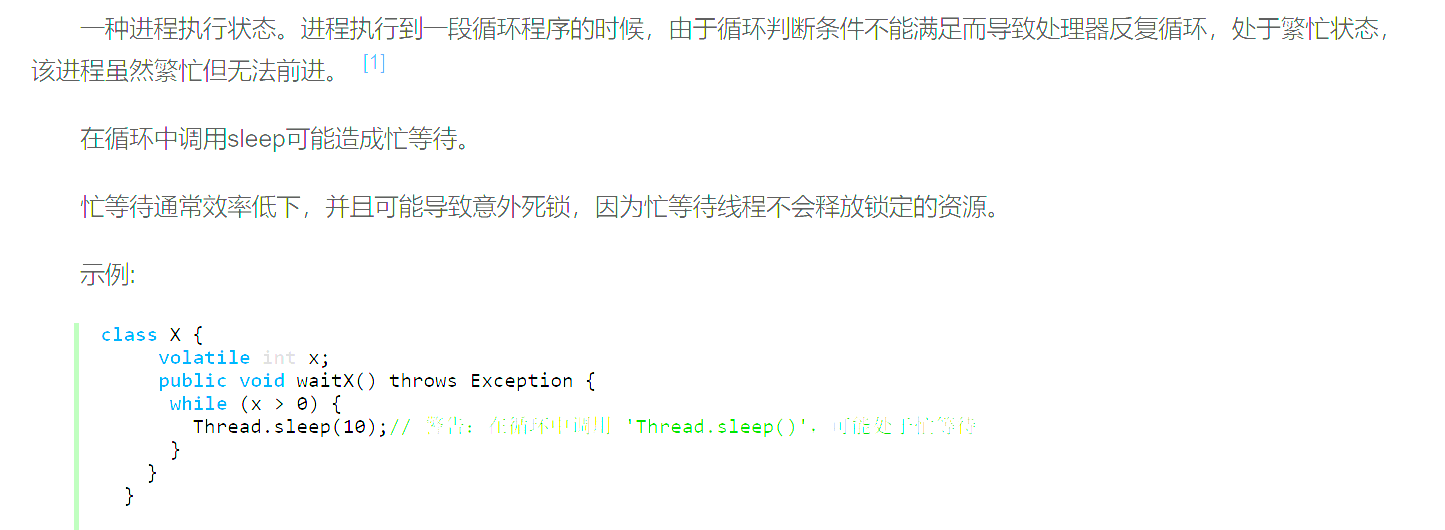
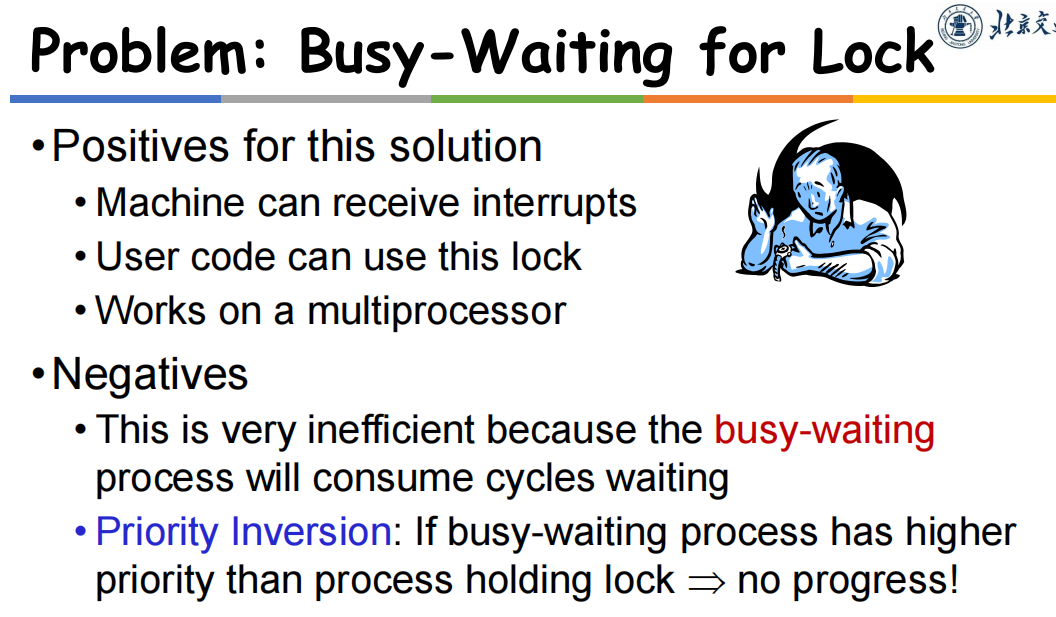
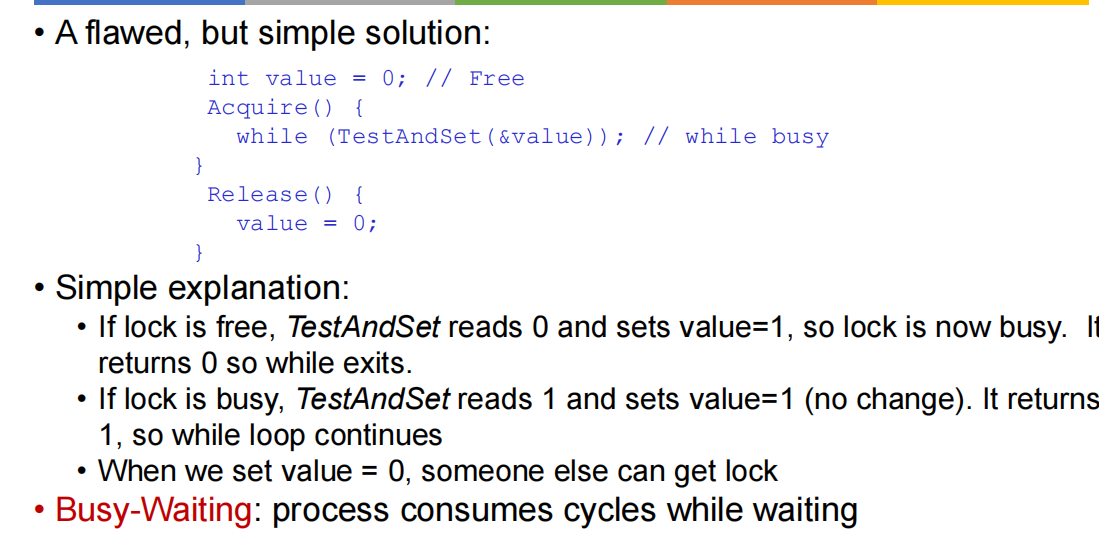
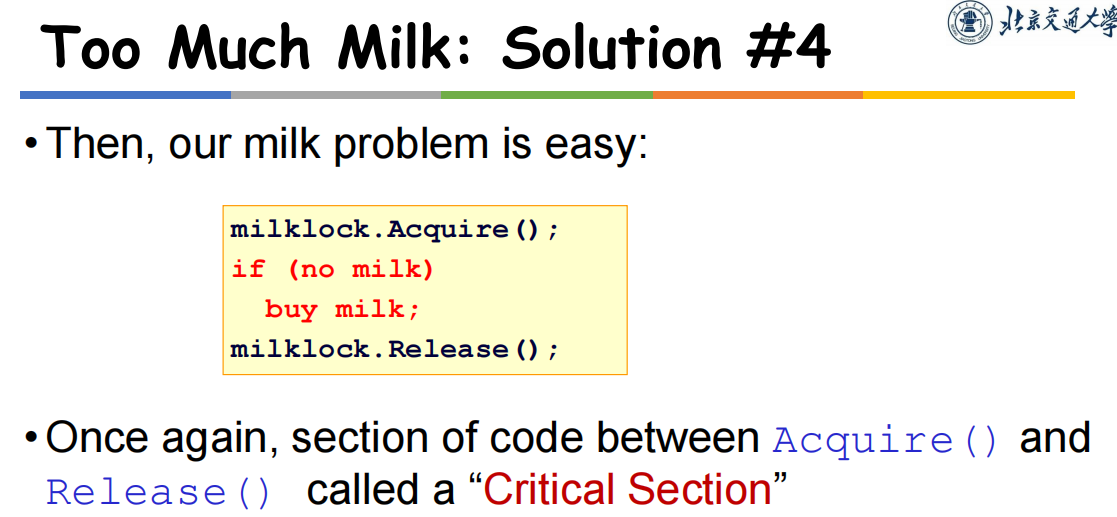
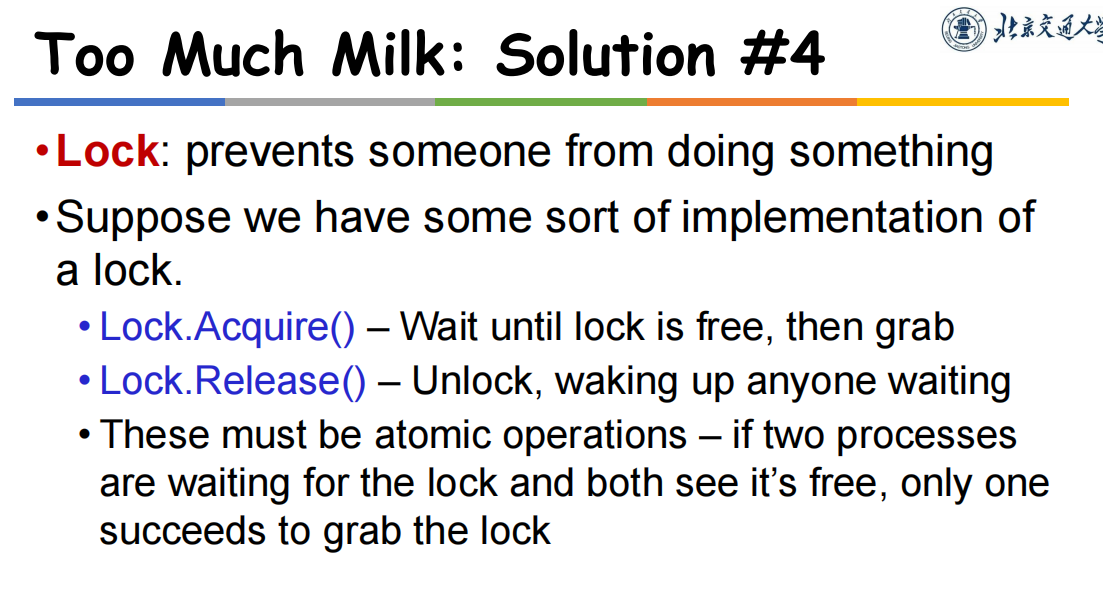
很难说服自己这真的有效

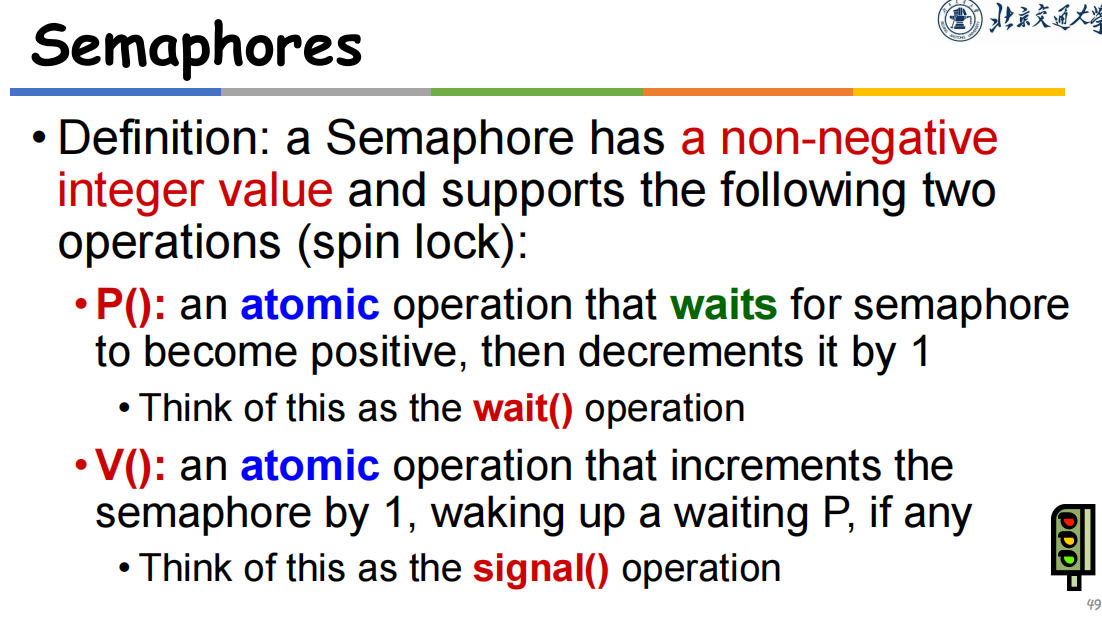
A的代码与B的代码不同，如果有很多进程呢？

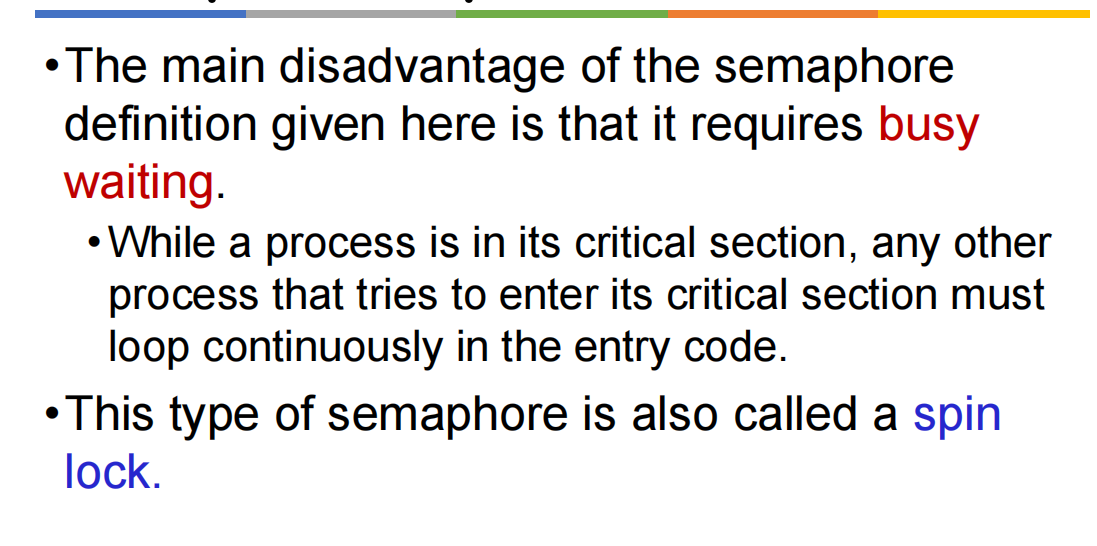
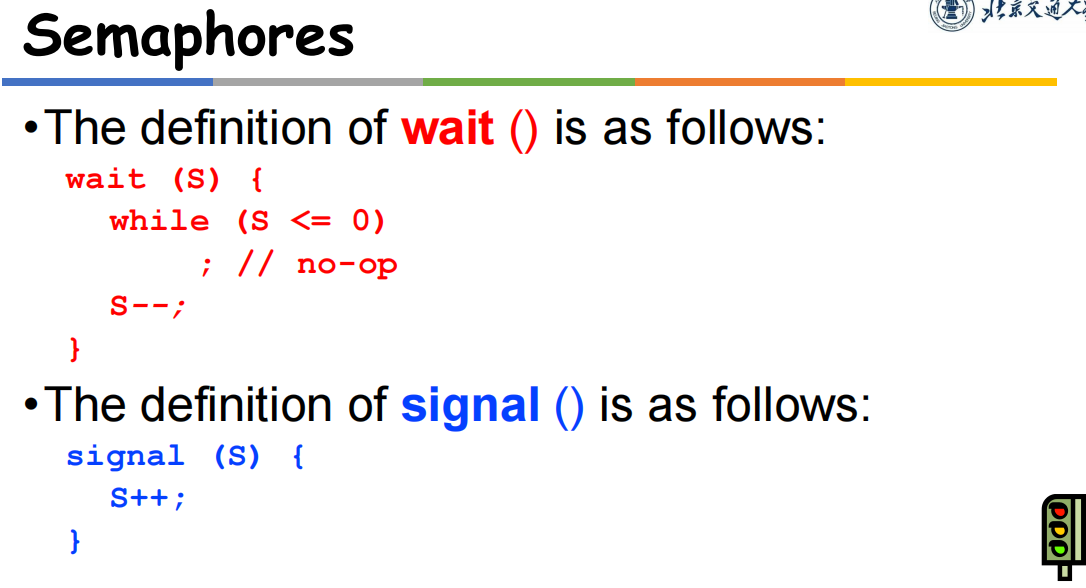
每个进程的代码必须略有不同

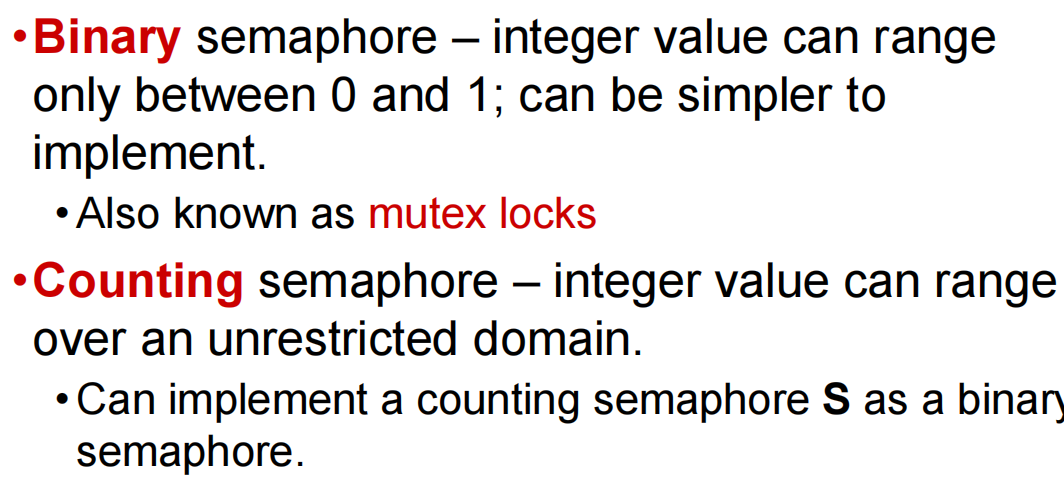
当A在等待时，它会消耗CPU时间

这被称为“忙碌的等待”



P操作 以及wait都是等到1的时候放行，





可以是简单的1-0 模式 也可以是数数模式

