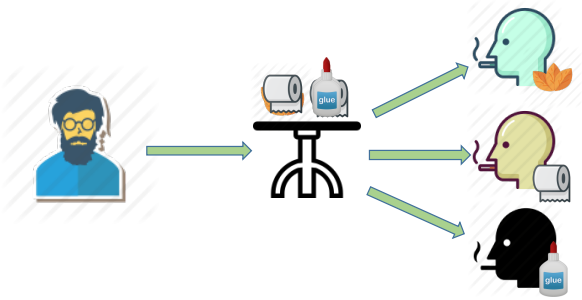
本节内容

吸烟者问题

问题描述

假设一个系统有三个抽烟者进程和一个供应者进程。每个抽烟者不停地卷烟并抽掉它，但是要卷起并抽掉一支烟，抽烟者需要有三种材料：烟草、纸和胶水。三个抽烟者中，第一个拥有烟草、第二个拥有纸、第三个拥有胶水。供应者进程无限地提供三种材料，供应者每次将两种材料放桌子上，拥有剩下那种材料的抽烟者卷一根烟并抽掉它，并给供应者进程一个信号告诉完成了，供应者就会放另外两种材料再桌上，这个过程一直重复（让三个抽烟者轮流地抽烟）



问题分析

假设一个系统有三个抽烟者进程和一个供应者进程。每个抽烟者不停地卷烟并抽掉它，但是要卷起并抽掉一支烟，抽烟者需要有三种材料：烟草、纸和胶水。三个抽烟者中，第一个拥有烟草、第二个拥有纸、第三个拥有胶水。供应者进程无限地提供三种材料，供应者每次将两种材料放桌子上，拥有剩下那种材料的抽烟者卷一根烟并抽掉它，并给供应者进程一个信号告诉完成了，供应者就会放另外两种材料再桌上，这个过程一直重复（让三个抽烟者轮流地抽烟）

本质上这题也属于“生产者-消费者”问题，更详细的说应该是“可生产多种产品的单生产者-多消费者”。

1. 关系分析。找出题目中描述的各个进程，分析它们之间的同步、互斥关系。
2. 整理思路。根据各进程的操作流程确定P、V操作的大致顺序
3. 设置信号量。设置需要的信号量，并根据题目条件确定信号量初值。（互斥信号量初值一般为1，同步信号量的初始值要看对应资源的初始值是多少）

桌子可以抽象为容量为1的缓冲区，要互斥访问

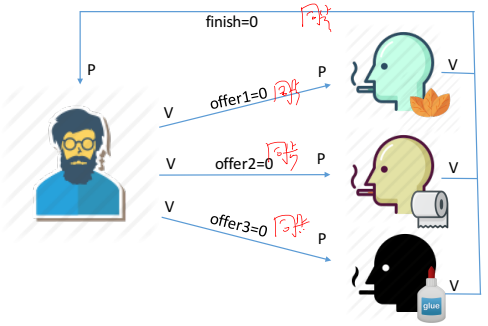
组合一：纸+胶水
组合二：烟草+胶水
组合三：烟草+纸

同步关系（从事件的角度来分析）：
桌上有组合一 → 第一个抽烟者取走东西
桌上有组合二 → 第二个抽烟者取走东西
桌上有组合三 → 第三个抽烟者取走东西
发出完成信号 → 供应者将下一个组合放到桌上

PV操作顺序：“前V后P”

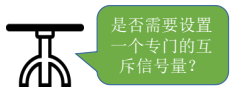
问题分析

假设一个系统有三个抽烟者进程和一个供应者进程。每个抽烟者不停地卷烟并抽掉它，但是要卷起并抽掉一支烟，抽烟者需要有三种材料：烟草、纸和胶水。三个抽烟者中，第一个拥有烟草、第二个拥有纸、第三个拥有胶水。供应者进程无限地提供三种材料，供应者每次将两种材料放桌子上，拥有剩下那种材料的抽烟者卷一根烟并抽掉它，并给供应者进程一个信号告诉完成了，供应者就会放另外两种材料再桌上，这个过程一直重复（让三个抽烟者轮流地抽烟）



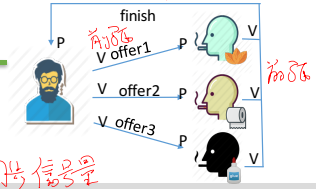
桌上有组合一 → 第一个抽烟者取走东西
桌上有组合二 → 第二个抽烟者取走东西
桌上有组合三 → 第三个抽烟者取走东西
发出完成信号 → 供应者将下一个组合放到桌上

如何实现



是否需要设置一个专门的互斥信号量？

缓冲区大小为1，同一时刻，四个同步信号量中至多有一个的值为1




```
provider  (){
    while(1){
        if(i==0)  {
            将组合一放桌上;
            V(offer1);
        } else  if(i==1){
            将组合二放桌上;
            V(offer2);
        } else  if(i==2){
            将组合三放桌上;
            V(offer3);
        }
        i = (i+1)%3;
        P(finish);
    }
}
```

```
semaphore  offer1  = 0;      //桌上组合一的数量
semaphore  offer2  = 0;      //桌上组合二的数量
semaphore  offer3  = 0;      //桌上组合三的数量
semaphore  finish  = 0;      //抽烟是否完成
int  i  = 0;                 //用于实现"三个抽烟者轮流抽烟"
```

```
smoker1  (){
    while(1){
        P(offer1);
        从桌上拿走组合一；卷烟；抽掉；
        V(finish);
    }
}
```

```
smoker2  (){
    while(1){
        P(offer2);
        从桌上拿走组合二；卷烟；抽掉；
        V(finish);
    }
}
```

```
smoker3  (){
    while(1){
        P(offer3);
        从桌上拿走组合三；卷烟；抽掉；
        V(finish);
    }
}
```


知识回顾与重要考点

吸烟者问题可以为我们解决“可以生产多个产品的单生产者”问题提供一个思路。
值得吸取的精华是：“轮流让各个吸烟者吸烟”必然需要“轮流的在桌上放上组合一、二、三”，注意体会我们是如何用一个整型变量 i 实现这个“轮流”过程的。
如果题目改为“每次随机地让一个吸烟者吸烟”，我们有应该如何用代码写出这个逻辑呢？

若一个生产者要生产多种产品（或者说会引发多种前驱事件），那么各个V操作应该放在各自对应的“事件”发生之后的位置。