



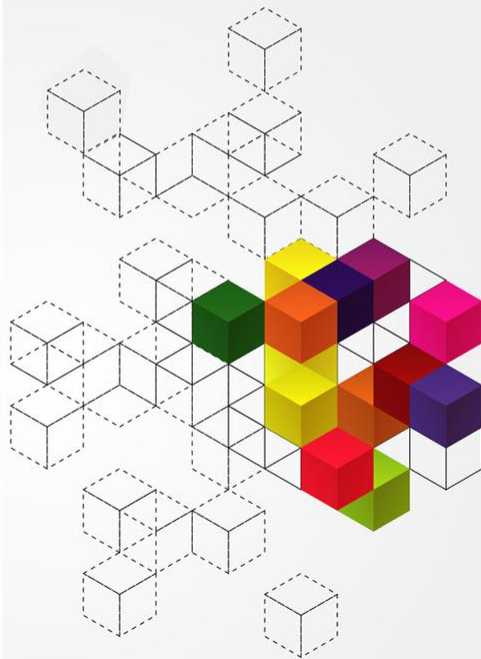
# 操作系统

Operating system

孔维强

大连理工大学

- 一、处理进程间接协作
- 二、处理进程直接协作
- 三、信号量值区间分析



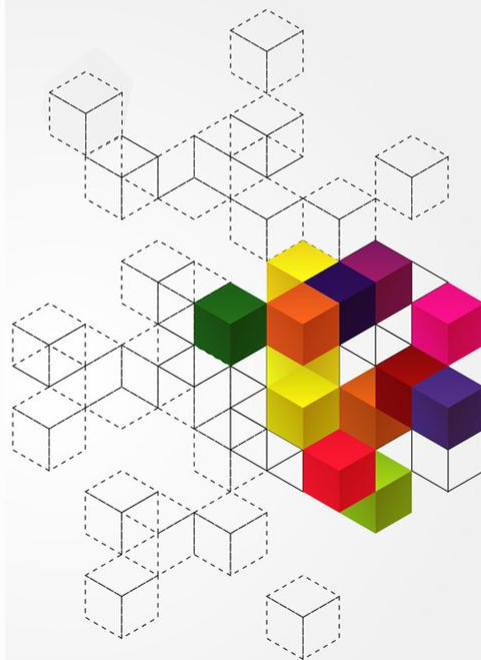
# 一、处理进程间接协作

## 1.用在解决临界区问题:

信号量mutex,初值=1

```
do{  
    wait(mutex);  
    临界区;  
    signal(mutex);  
    退出区;  
}while(1);
```

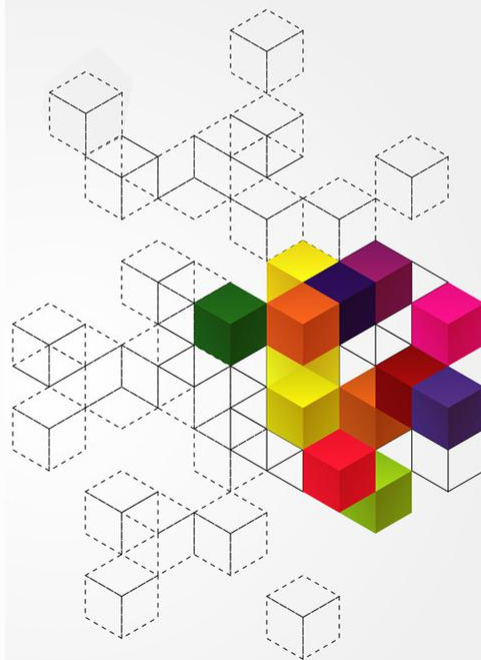
使用信号量的互斥实现



# 一、处理进程间接协作

## ● 用在解决临界区问题：

- 解决多资源竞争
- 例如：拥有固定数量座位的自习室自习问题
- 进入自习室，登记
- 退出时，撤销登记
- 资源：自习室座位



## 二、处理进程直接协作

### 2.用在解决进程直接同步问题:

比如, 我们做一个软件project, 分成几个部分  
 $m_1, m_2, \dots, m_k$

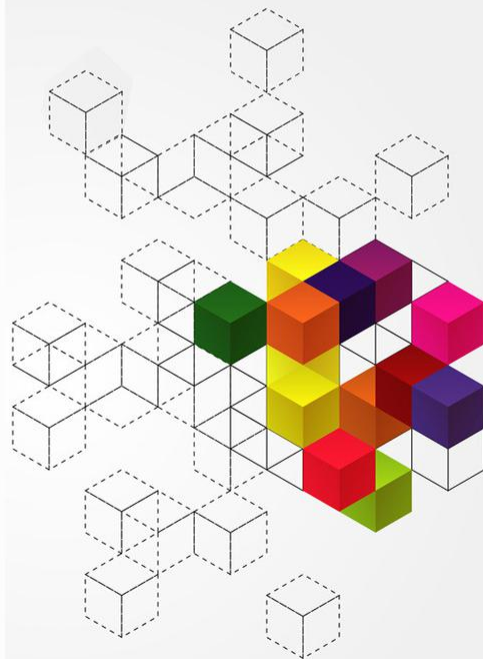
- 必须做完 $m_i$ 之后, 才能开始做 $m_{i+1}$

可以借助信号量来实现这 $k$ 个进程的先后协同

```
do{  
    实施 $m_i$   
    signal( $sem_i$ );  
}while(1);
```

```
do{  
    wait( $sem_i$ );  
    实施 $m_{i+1}$   
}while(1);
```

信号量初值设定必须正确,  $sem_i$ 的初值应为?



## 二、处理进程直接协作

### 2.用在解决进程直接同步问题:

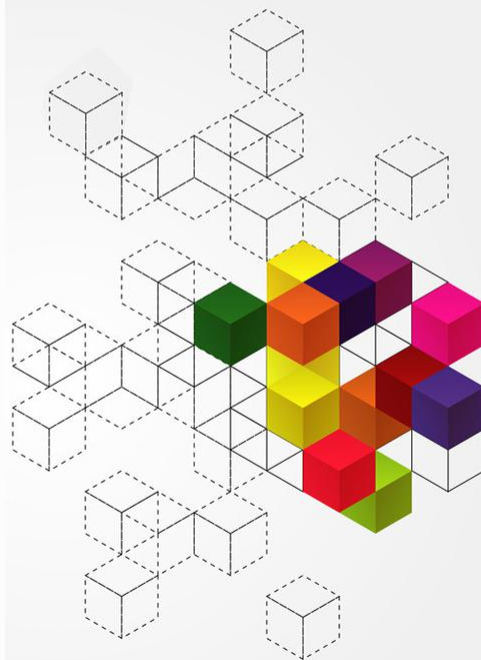
- 实际例子:
- 公交司机, 售票员 协作问题

**司机  $P_1$**

```
While(true)
{
    启动车辆;
    正常运行;
    到站停车;
}
```

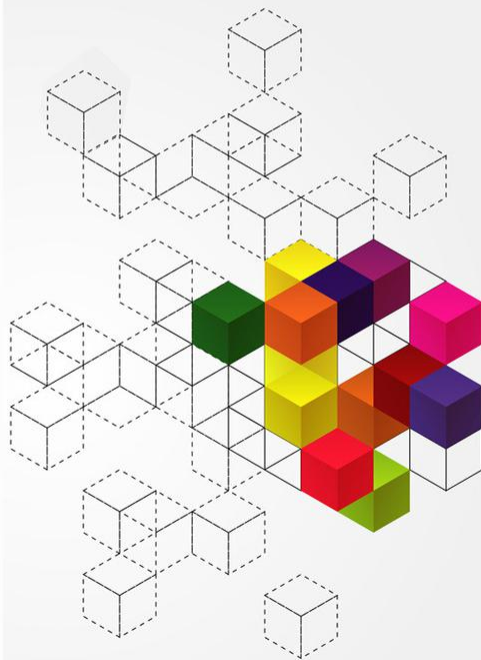
**售票员  $P_2$**

```
While(true)
{
    关门;
    售票;
    开门;
}
```



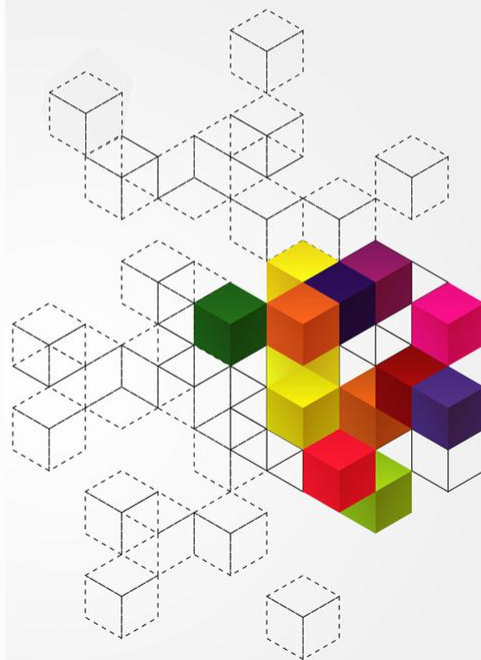
### 三、信号量值区间分析

- ①  $n$  个并发进程，信号量初始值为 1，当  $n$  个进程都执行 P 操作后，信号量的值为多少？
- ② 信号量初值为 4，多次 PV 操作后变为 -2，那么获得资源的进程数目是多少？
- ③ 5 个并发进程，信号量初始值为 3，那么信号量取值范围是多少？



# 本讲小结

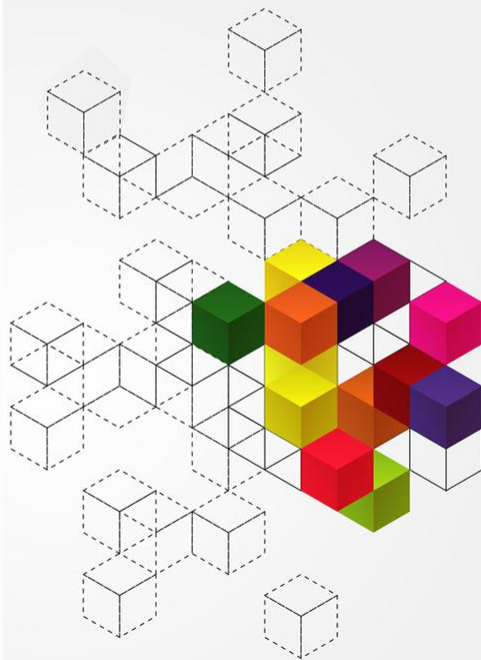
- 处理进程间接协作
- 处理进程直接协作
- 信号量值区间分析





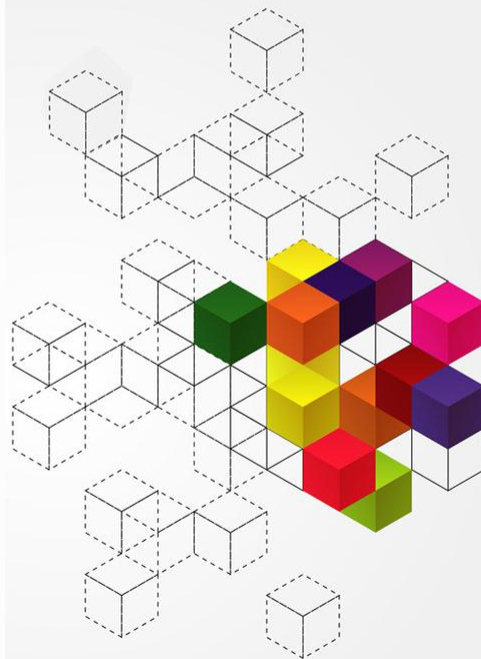
# 疑惑

- 信号量到底怎么用？初值怎么设？
  - 同步信号量：同步时使用，这些信号量只与制约进程和被制约进程有关而不是与整组并发进程有关。
  - 互斥信号量：互斥时使用。



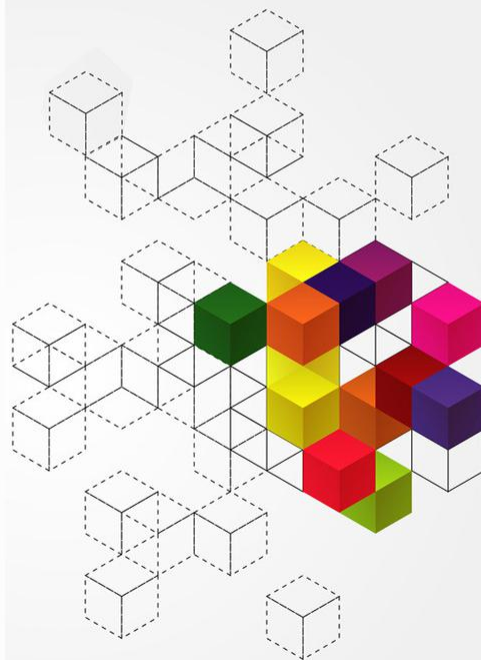
# P、V操作讨论

- 信号量的物理意义：
  - $S > 0$  表示有  $S$  个资源可用
  - $S = 0$  表示无资源可用
  - $S < 0$ ,  $|S|$  表示  $S$  等待队列中的进程个数
- $\text{wait}(S)$ : 申请一个资源
- $\text{signal}(S)$ : 释放一个资源



# P、V (wait、singal) 操作讨论

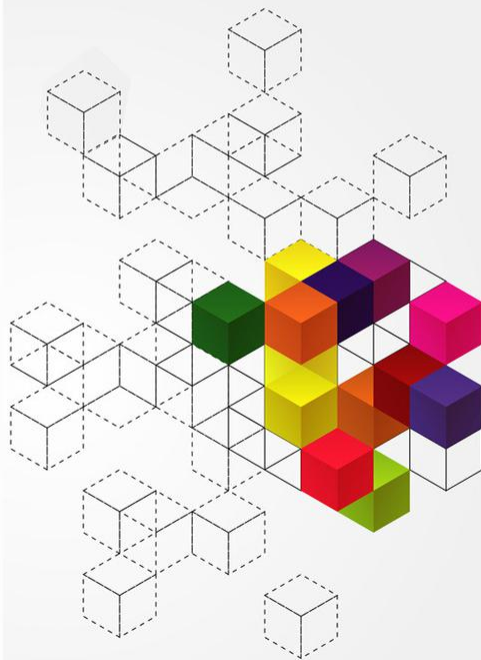
- 互斥信号量
  - 必定成对出现：进入临界区——临界区——退出临界区
- 同步信号量
  - 未必成对出现，依赖于同步关系的性质
- 当为互斥操作时，它们同处于同一进程
- 当为同步操作时，则不在同一进程中出现
- 如果P(S1)和P(S2)两个操作在一起，那么P操作的顺序至关重要，一个同步P操作与一个互斥P操作在一起时，同步P操作在互斥P操作前（互斥信号量永远紧邻临界区）
- 两个V操作的顺序无关紧要



# Exercise 1

例：有一个充分大的池子，两个人分别向池内投球。甲投红球，乙投蓝球，一次投一个。开始时池子中有红蓝球各一个。

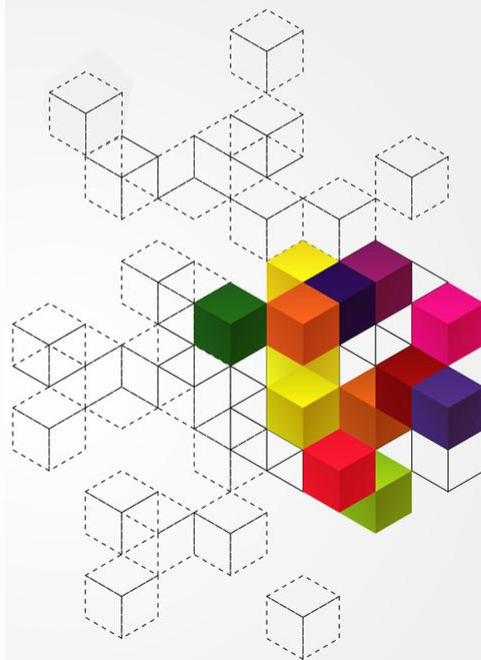
要求池中红蓝球满足：蓝球数 $\leq$ 红球数 $\leq$ 2蓝球数。用P、V操作描述两个进程。



甲  
while(true){  
  P(r);  
  扔一个红球;  
  V(b);  
}

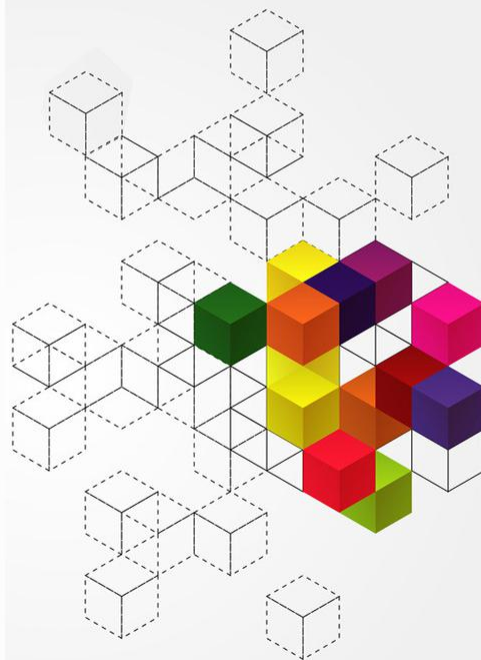
乙  
while(true){  
  P(b);  
  扔一个蓝球;  
  V(r);  
  V(r);  
}

r初值为1, b初值为0



## Exercise 2: 理发师问题

- 理发店有一位理发师、一把理发椅和 $n$ 把供等候理发的顾客坐的椅子
- 如果没有顾客，理发师便在理发椅上睡觉
- 一个顾客到来时，他/她必须叫醒理发师
- 如果理发师正在理发时又有顾客来到，则如果有空椅子可坐，就坐下来等待，否则就离开



设置整型变量 waiting  
CHAIRS

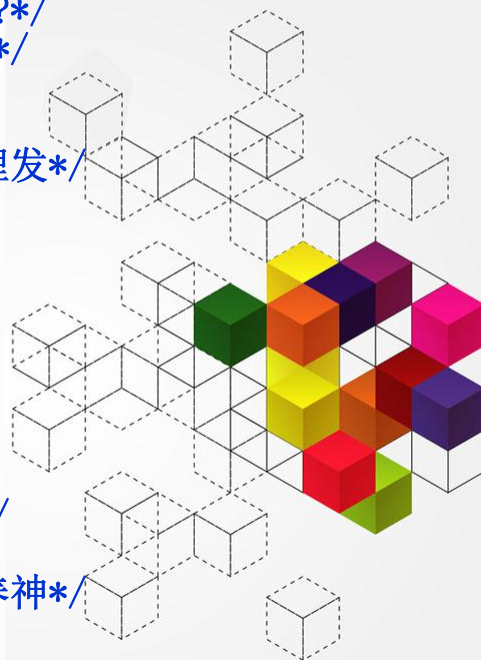
/\*等候理发的顾客数\*/  
/\*为顾客准备的椅子数\*/

信号量 customers顾客数, barbers空着的理发师数, mutex  
customers = 0; barbers = 0; waiting = 0; mutex = 1;

```
Procedure barber;  
while(TRUE) {  
    P(cutomers);  
    P(mutex);  
    waiting = waiting-1;  
    V(barbers);  
    V(mutex);  
    cut-hair( );  
}  
procedure customer  
{  
    P(mutex);  
    if waiting<CHAIRS{  
        waiting = waiting+1;  
        V(customers);  
        V(mutex);  
        P(barbers);  
        get-haircut( );  
    }  
    else V(mutex);  
}
```

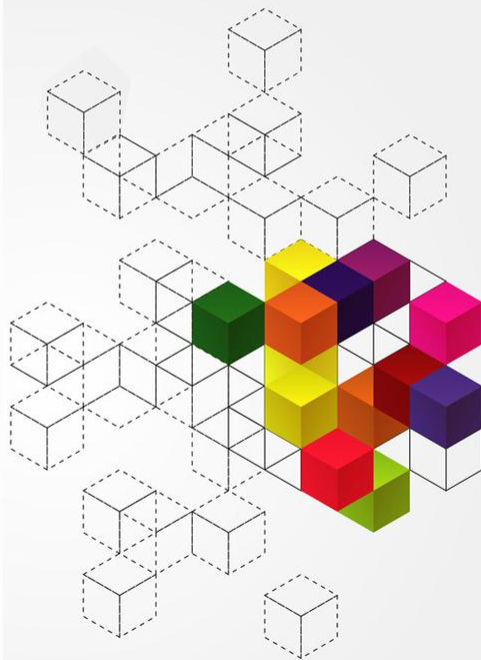
```
/*理完一人, 还有顾客吗?*/  
/*若无顾客, 理发师睡眠*/  
/*进程互斥*/  
/*等候顾客数少一个*/  
/*理发师去为一个顾客理发*/  
/*开放临界区*/  
/*正在理发*/  
  
/*进程互斥*/  
/*看看有没有空椅子*/  
/*等候顾客数加1*/  
/*必要的话唤醒理发师*/  
/*开放临界区*/  
/*无理发师, 顾客坐着养神*/  
/*理发*/
```

```
/*人满了, 走吧!*/
```



## Exercise3: 苹果桔子问题

桌上有一只盘子，每次只能放入一只水果；  
爸爸专向盘子中放苹果(apple)，妈妈专向  
盘子中放桔子(orange)，儿子专等吃盘子  
中的桔子，女儿专等吃盘子里的苹果





```
sp          /* 盘子里可以放几个水果 */
sg1         /* 盘子里有桔子 */
sg2         /* 盘子里有苹果 */
设初值      sp := 1; sg1 := 0; sg2 := 0;
```

```
process father
While(true) {
    削一个苹果;
    P(sp);
    把苹果放入plate;
    V(sg2);
}

process mother
While(true) {
    剥一个桔子;
    P(sp);
    把桔子放入plate;
    V(sg1);
}
```

```
process son
While(true) {
    P(sg1);
    从plate中取桔子;
    V(sp);
    吃桔子;
}

process daughter
While(true) {
    P(sg2);
    从plate中取苹果;
    V(sp);
    吃苹果;
}
```

