



河海大学

计算机与信息学院

第三章 习题



3 题

- 有两个优先级相同的进程 P1和P2，各自执行的操作如下。

信号量 S1 和 S2 的初值均为 0。试问 P1、P2 并发执行后，x、y、z 的值各为多少？

```
P1() {  
  y = 1;  
  y = y + 3;  
  V(S1);  
  z = y + 1;  
  P(S2);  
  y = z + y;  
}
```

```
P2() {  
  x = 1;  
  x = x + 5;  
  P(S1);  
  x = x + y;  
  V(S2);  
  z = z + x;  
}
```



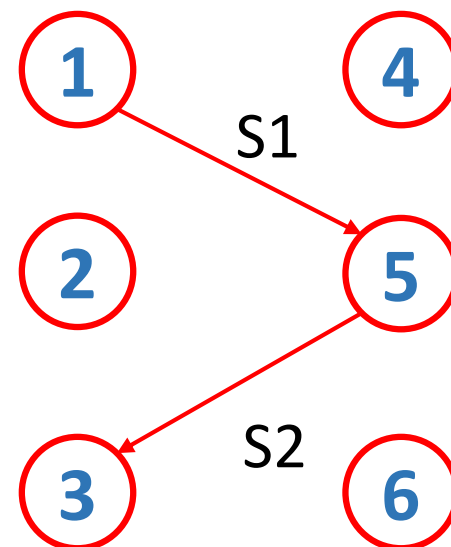
3 题

- 有两个优先级相同的进程 P1和P2，各自执行的操作如下。

信号量 S1 和 S2 的初值均为 0。试问 P1、P2 并发执行后，x、y、z 的值各为多少？

```
P1() {  
  y = 1; 1  
  y = y + 3;  
  V(S1);  
  z = y + 1; 2  
  P(S2);  
  y = z + y; 3  
}
```

```
P2() {  
  x = 1; 4  
  x = x + 5;  
  P(S1);  
  x = x + y; 5  
  V(S2);  
  z = z + x; 6  
}
```





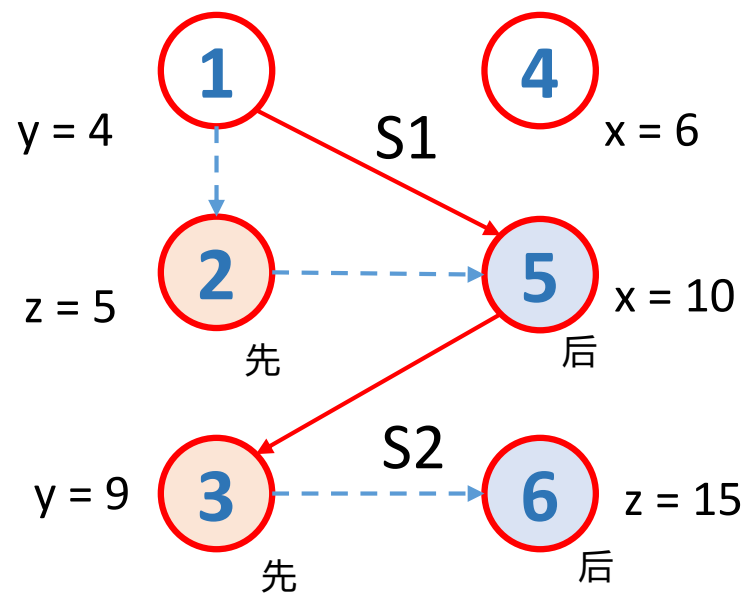
3 题

- 有两个优先级相同的进程 P1和P2，各自执行的操作如下。

信号量 S1 和 S2 的初值均为 0。试问 P1、P2 并发执行后，x、y、z 的值各为多少？

```
P1() {  
  y = 1; 1  
  y = y + 3;  
  V(S1);  
  z = y + 1; 2  
  P(S2);  
  y = z + y; 3  
}
```

```
P2() {  
  x = 1; 4  
  x = x + 5;  
  P(S1);  
  x = x + y; 5  
  V(S2);  
  z = z + x; 6  
}
```





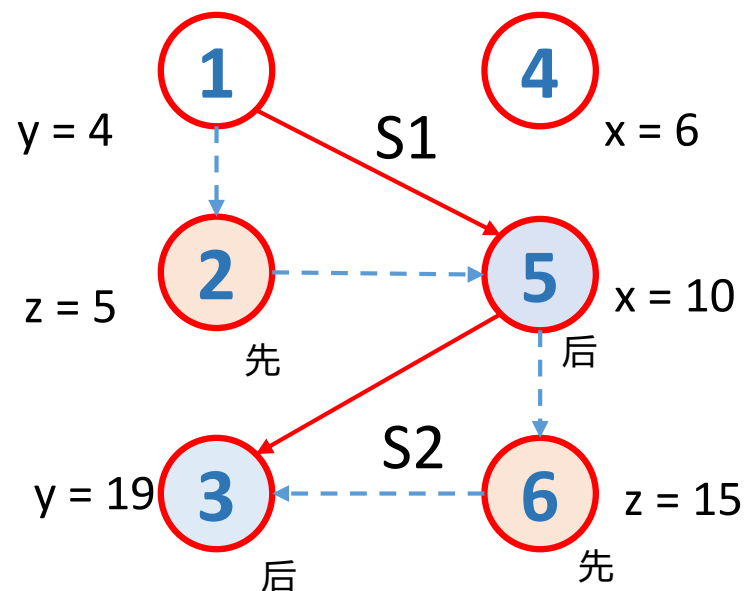
3 题

- 有两个优先级相同的进程 P1和P2，各自执行的操作如下。

信号量 S1 和 S2 的初值均为 0。试问 P1、P2 并发执行后，x、y、z 的值各为多少？

```
P1() {  
  y = 1; 1  
  y = y + 3;  
  V(S1);  
  z = y + 1; 2  
  P(S2);  
  y = z + y; 3  
}
```

```
P2() {  
  x = 1; 4  
  x = x + 5;  
  P(S1);  
  x = x + y; 5  
  V(S2);  
  z = z + x; 6  
}
```



x = 10 ; y = 19 ; z = 15



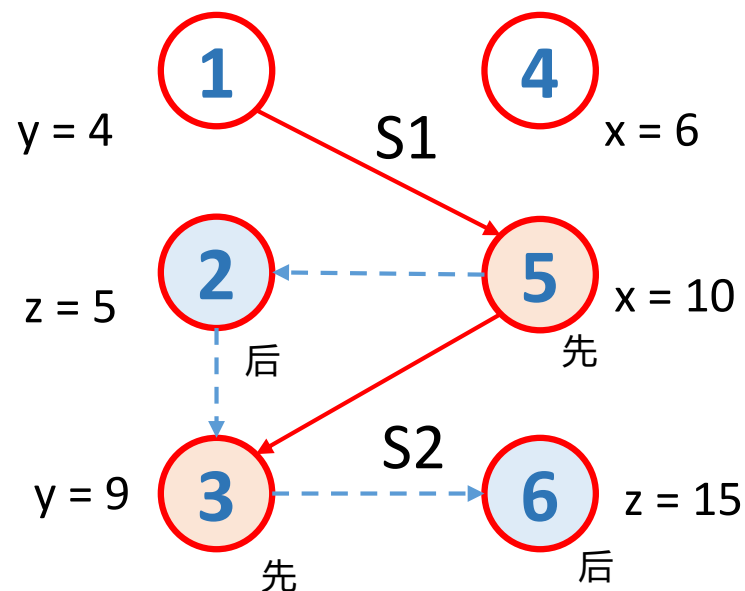
3 题

- 有两个优先级相同的进程 P1和P2，各自执行的操作如下。

信号量 S1 和 S2 的初值均为 0。试问 P1、P2 并发执行后，x、y、z 的值各为多少？

```
P1() {  
  y = 1; 1  
  y = y + 3;  
  V(S1);  
  z = y + 1; 2  
  P(S2);  
  y = z + y; 3  
}
```

```
P2() {  
  x = 1; 4  
  x = x + 5;  
  P(S1);  
  x = x + y; 5  
  V(S2);  
  z = z + x; 6  
}
```





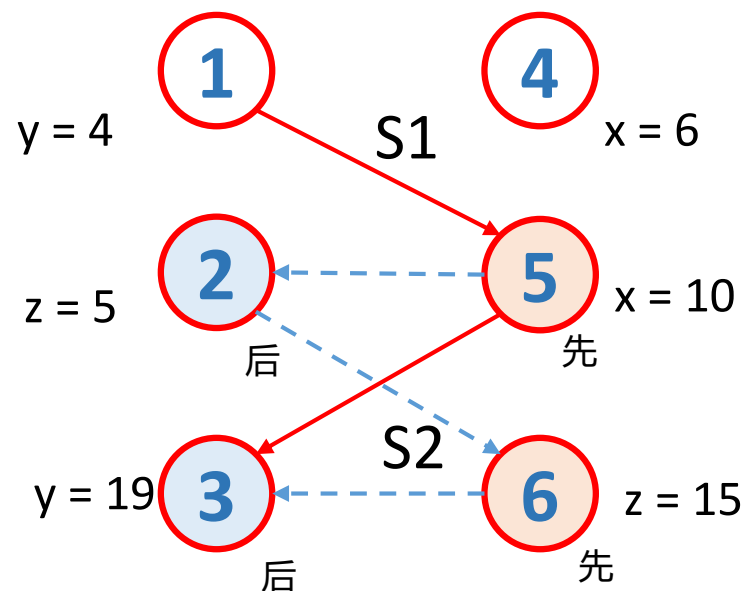
3 题

- 有两个优先级相同的进程 P1和P2，各自执行的操作如下。

信号量 S1 和 S2 的初值均为 0。试问 P1、P2 并发执行后，x、y、z 的值各为多少？

```
P1() {  
  y = 1; 1  
  y = y + 3;  
  V(S1);  
  z = y + 1; 2  
  P(S2);  
  y = z + y; 3  
}
```

```
P2() {  
  x = 1; 4  
  x = x + 5;  
  P(S1);  
  x = x + y; 5  
  V(S2);  
  z = z + x; 6  
}
```



x = 10 ; y = 19 ; z = 15



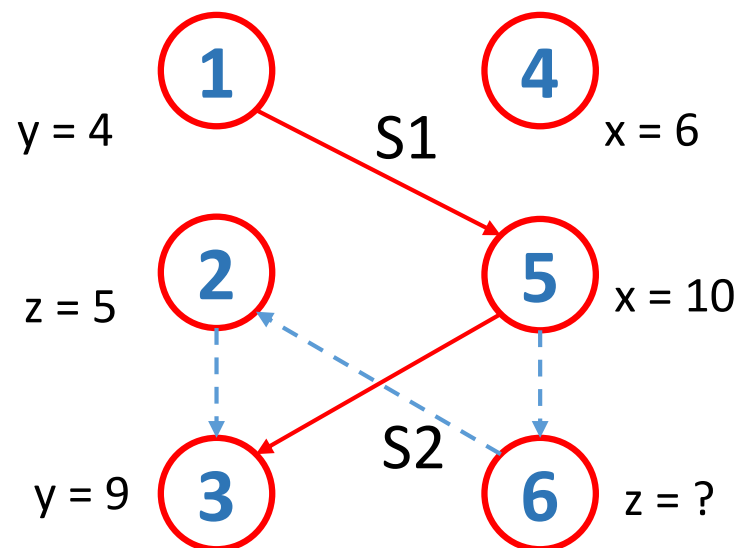
3 题

- 有两个优先级相同的进程 P1和P2，各自执行的操作如下。

信号量 S1 和 S2 的初值均为 0。试问 P1、P2 并发执行后，x、y、z 的值各为多少？

```
P1() {  
  y = 1; 1  
  y = y + 3;  
  V(S1);  
  z = y + 1; 2  
  P(S2);  
  y = z + y; 3  
}
```

```
P2() {  
  x = 1; 4  
  x = x + 5;  
  P(S1);  
  x = x + y; 5  
  V(S2);  
  z = z + x; 6  
}
```



x = 10 ; y = 9 ; z = 5

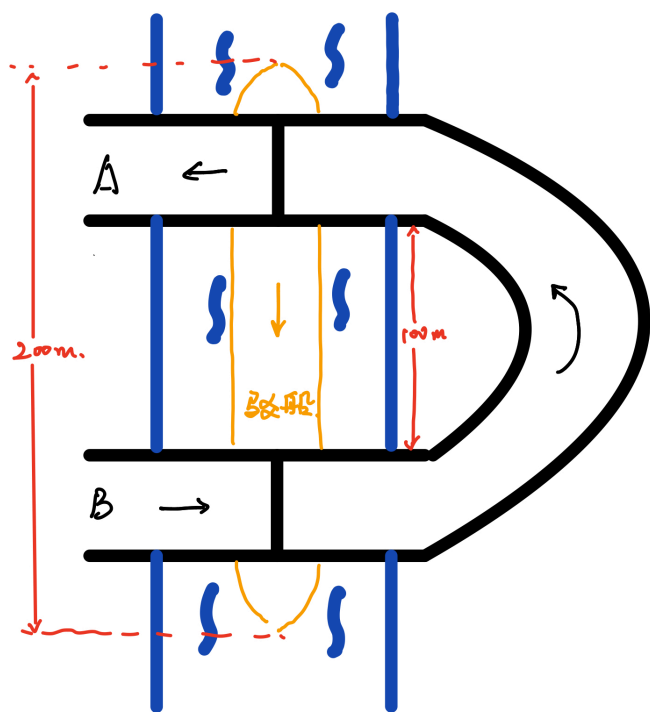


20 题

- 一条公路两次横跨运河，两个运河桥相距100m，均带有闸门，供船只通过运河桥，运河和公路的交通均是单向的。运河上的运输由驳船负责，在一艘驳船接近吊桥A时就拉汽笛警告，若桥上无车辆，吊桥就吊起，直到驳船尾通过此桥为止。对吊桥B也按照同样的次序进行处理。典型的驳船长度为200m，当它在河上航行时是否会产生死锁？若会，请说明理由，提出一种防止死锁的方法，并用信号量来实现驳船的同步。



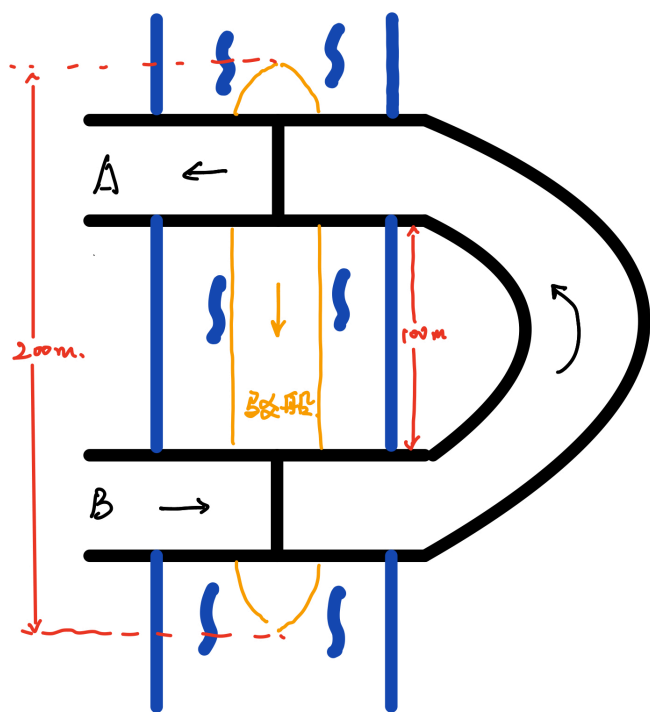
20 题



- 一条公路两次横跨运河，两个运河桥相距100m，均带有闸门，供船只通过运河桥，运河和公路的交通均是单向的。运河上的运输由驳船负责，在一艘驳船接近吊桥A时就拉汽笛警告，若桥上无车辆，吊桥就吊起，直到驳船尾通过此桥为止。对吊桥B也按照同样的次序进行处理。典型的驳船长度为200m，当它在河上航行时是否会产生死锁？若会，请说明理由，提出一种防止死锁的方法，并用信号量来实现驳船的同步。



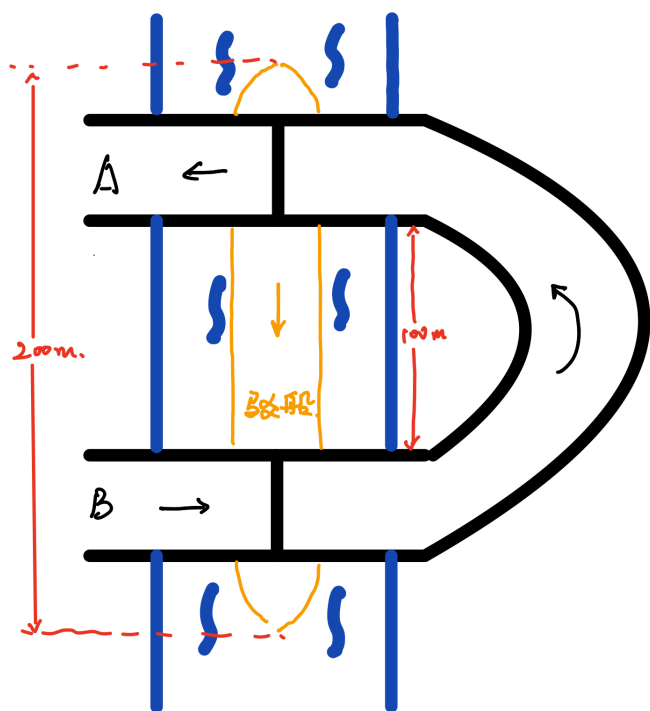
20 题



- 假设车辆先打开B，然后待通过A后，才可以改变B的状态
- 在此假设下，若有车辆先通过 B 闸口进入到公路上（非桥面），此时驳船到达；
- 由于驳船长于桥间距离，所以打开A闸口后，还需要打开B闸口才可以让驳船通过；
- 但由于车辆还在B、A之间，所以不能把B打开；由于船还没有通过，A也不能合上。处于死锁状态。



20 题



1. 可以破坏必要条件2，采用 AND 型信号量；
2. 可以破坏必要条件4，采用层次分配方法；

结果都是一样，有车过时，不准船走；有船过时，不准车走。



28题

- 有一个仓库可存放A 、 B 两种零件，最大库容量各为m 个。生产车间不断地取A 和B进行装配，每次各取一个。为避免零件锈蚀，按先入库者先出库的原则。有两组供应商分别不断地供应A 和B ，每次一个。为保证配套和合理库存，当某种零件比另一种零件超过n ($n < m$) 个时，暂停对数量大的零件的进货，集中补充数量少的零件。试用信号量与P 、 V 操作正确地实现它们之间的同步关系。



28题

- A 零件数量 $\leq m$, 设置信号量 empty1, full1;
- B 零件数量 $\leq m$, 设置信号量 empty2, full2;
- A 零件数量 - B 零件数量 $\leq n$, 设置信号量 sa;
- B 零件数量 - A 零件数量 $\leq n$, 设置信号量 sb;
- 由于A, B零件存储在同一仓库, 存取时只能取一, 设互斥信号量mutex;
- 为遵循先入库者先出库的原则, A 、 B 零件可以组织成两个循环队列, 并增加入库指针in1 、 in2 和出库指针out1 、 out2 来控制顺序。



28题

```
semaphore empty1 = m, empty2 = m  
semaphore full1 = 0, full2 = 0;  
semaphore sa = sb = n;  
semaphore mutex = 1;  
int in1 = in2 = out1 = out2 = 0;  
itemA[m] buffer1; itemB[m] buffer2;
```

```
process proA{  
    while (true){  
        P(empty1);  
        P(sa);  
        P(mutex);  
        buffer1[in1] = itemA;  
        in1 = (in1++)%m;  
        V(mutex);  
        V(sb);  
        V(full1);  
    }  
}
```

```
process proB{  
    while (true){  
        P(empty2);  
        P(sb);  
        P(mutex);  
        buffer2[in2] = itemB;  
        in2 = (in2++)%m;  
        V(mutex);  
        V(sa);  
        V(full2);  
    }  
}
```

```
process Consumer{  
    while(true){  
        P(full1);  
        P(full2);  
        P(mutex);  
        itemA = buffer1[out1];  
        itemB = buffer2[out2];  
        out1 = (out1++)%m;  
        out2 = (out2++)%m;  
        V(mutex);  
        V(empty1);  
        V(empty2);  
    }  
}
```