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;
}
```

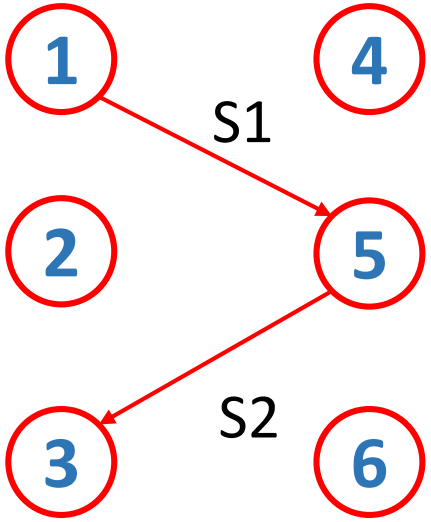
```
P2() {
  x = 1;
  x = x + 5;
  P(S1);
  x = x + y;
  V(S2);
  z = z + x;
}
```

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

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

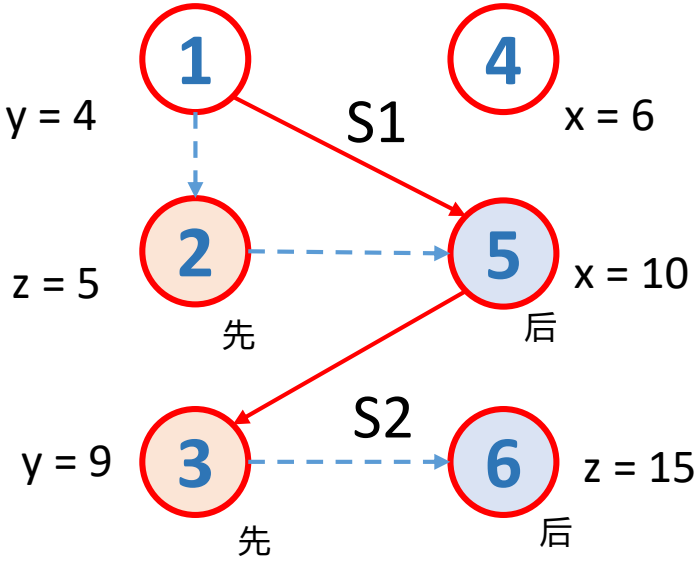


3 题

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

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



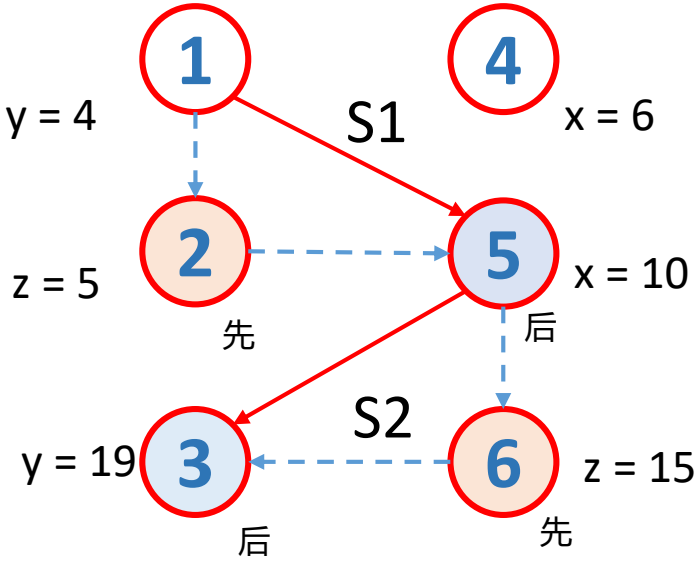
x = 10 ; y = 9 ; z = 15

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

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



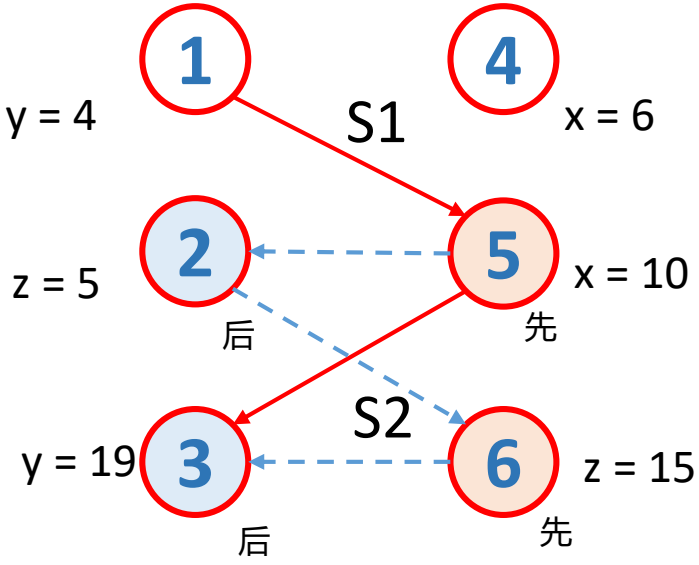
x = 10 ; y = 19 ; z = 15

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

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



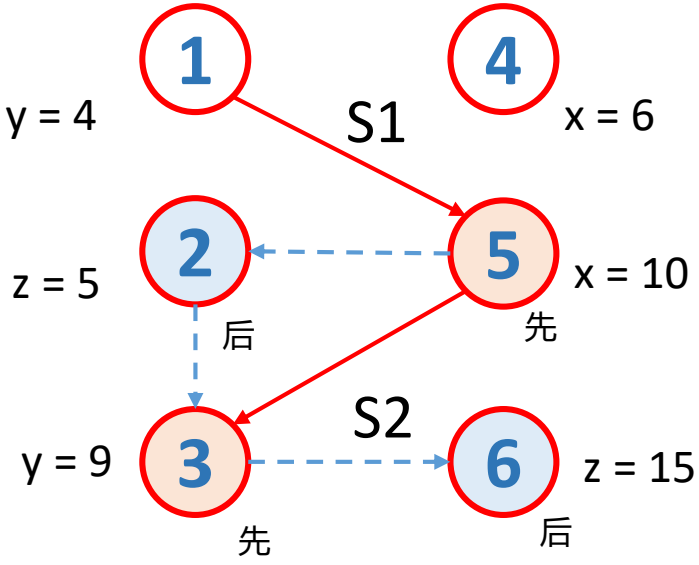
x = 10 ; y = 19 ; z = 15

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

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



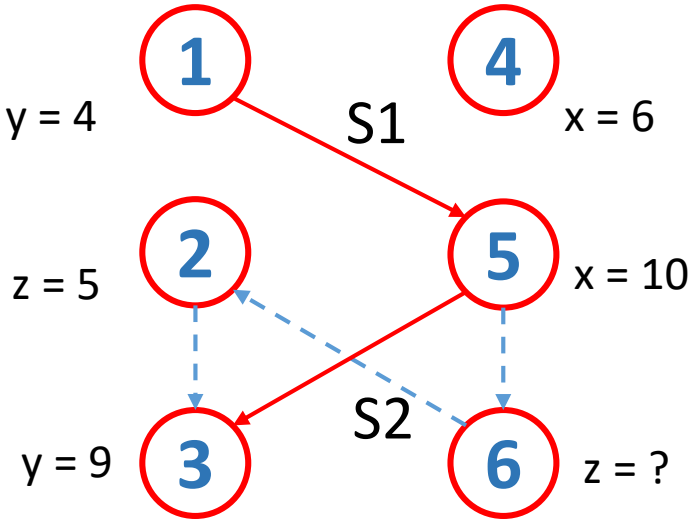
x = 10 ; y = 9 ; z = 15

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

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



x = 10 ; y = 9 ; z = 5

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 ≤ m ，设置信号量 empty1，full1；
- B 零件数量 ≤ m ，设置信号量 empty2，full2；
- A 零件数量 - B 零件数量 ≤ n ，设置信号量sa；
- B 零件数量 - A 零件数量 ≤ 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形队列，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);
  }
}
```

42题

- 举例说明不能解决互斥问题

```
bool blocked[2];
enum{0,1} turn;
blocked[0] = bolcked[1] = false;
turn = 1
```

	id = 0	id = 1
1	blocked[id] = true; //blcked[0] = true	
2	while (turn != id) { //turn = 1	
3	while (blocked[1-id]); //blocked[1] = false	
4		blocked[id] = true; //blocked[1] = true
5		while (turn != id) { //turn = 1
6		....
7		}
8	turn = id;	
9	}	
10		/* Critical Section*/
11	/* Critical Section*/	
12	...	...