



河海大学

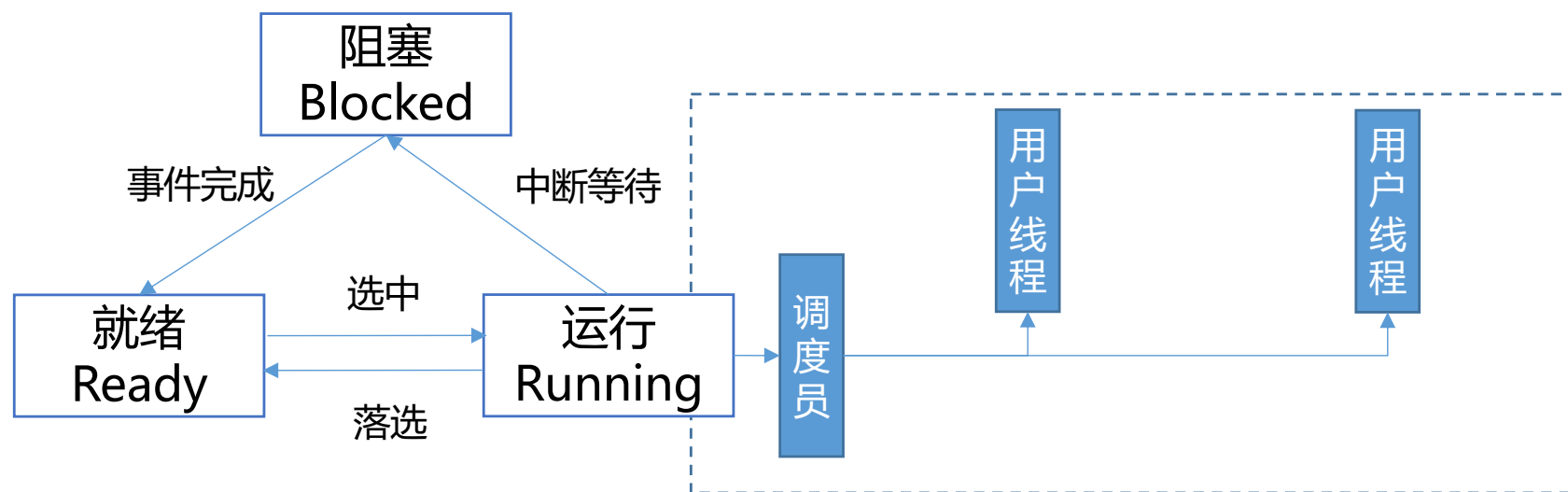
计算机与信息学院

# 习题讲解



## 第二章 3

- 某操作系统不支持多线程机制，高级程序设计语言提供了用户级多线程库，请画出用户级线程与操作系统进程之间的状态转换图。





## 第二章 6

- 假定执行作业 Job1~Job5，作业号的数字下标为其到达顺序，即依次按照序号 1、2、3、4、5 进入单处理器系统，各作业的执行时间和优先数如下表所示。

作业号	执行时间	优先权
Job1	10	3
Job2	1	1
Job3	2	3
Job4	1	4
Job5	5	2

- 分别给出先来先服务调度算法、时间片轮转算法（时间片长度为 1ms）、短作业优先算法及非抢占式的优先数调度算法（优先数越小则优先级越高）下各作业的执行次序；
- 计算每种情况下作业的平均周转时间。



## 第二章 6

先来先服务调度算法

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1										2	3		4	5					

$$T_{FCFS} = \frac{(10 + 11 + 13 + 14 + 19)}{5} = 13.4 \text{ ms}$$

时间片轮转算法 ( 时间片长度为 1ms )

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	1	3	5	1	5	1	5	1	5	1	1	1	1	1	1

$$T_{RR} = \frac{(19 + 2 + 7 + 4 + 14)}{5} = 9.2 \text{ ms}$$



## 第二章 6

短作业优先算法

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2	4	3	5					1											

$$T_{SJF} = \frac{(19 + 1 + 4 + 2 + 9)}{5} = 7 \text{ ms}$$

非抢占式的优先数调度算法（优先数越小则优先级越高）

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2	5				1											3	4		

$$T_{PS} = \frac{(16 + 1 + 18 + 19 + 6)}{5} = 12 \text{ ms}$$



## 第二章 7

- 有 5 个批处理作业 A - E 均已到达计算中心。对于时间片轮转算法 RR，优先级调度算法 PS，短作业优先算法 SJF，先来先服务算法 FCFS，在忽略进程切换时间的前提下，计算平均作业周转时间。其中 5 是最高级。
  - 时间片为 2min，其它算法单道运行，直到结束（非抢占式）

序号	作业名称	运行时间	到达次序	优先级
1	A	2 min	5	1
2	B	4 min	3	2
3	C	6 min	1	3
4	D	8 min	2	4
5	E	10 min	4	5 (最高)



# 11 题

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
C					D								B				E										A		

$$T_{FCFS} = \frac{(30 + 18 + 6 + 14 + 28)}{5} = 19.2 \text{ min}$$

序号	作业名称	运行时间	到达次序	优先级
1	A	2 min	5	1
2	B	4 min	3	2
3	C	6 min	1	3
4	D	8 min	2	4
5	E	10 min	4	5 (最高)



# 11 题

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

E D C B A

$$T_{PS} = \frac{(30 + 28 + 24 + 18 + 10)}{5} = 22 \text{ min}$$

序号	作业名称	运行时间	到达次序	优先级
1	A	2 min	5	1
2	B	4 min	3	2
3	C	6 min	1	3
4	D	8 min	2	4
5	E	10 min	4	5 (最高)





# 11 题

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
A	B	C	D	E	B	C	D	E	C	D	E	D	E	E															

$$T_{RR} = \frac{(2 + 12 + 20 + 26 + 30)}{5} = 18 \text{ min}$$

序号	作业名称	运行时间	到达次序	优先级
1	A	2 min	5	1
2	B	4 min	3	2
3	C	6 min	1	3
4	D	8 min	2	4
5	E	10 min	4	5 (最高)



# 11 题

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
A	B				C						D								E										

$$T_{SJF} = \frac{(2 + 6 + 12 + 20 + 30)}{5} = 14 \text{ min}$$

序号	作业名称	运行时间	到达次序	优先级
1	A	2 min	5	1
2	B	4 min	3	2
3	C	6 min	1	3
4	D	8 min	2	4
5	E	10 min	4	5 (最高)



## 第三章 2

- 已知Dijkstra 临界区软件算法的描述如下。试说明此算法满足临界区管理原则。

```
enum {idle,wantin,incs} flag[n];
int turn;
turn=0 or 1 ... or n-1;
process Pi() {      /*i=0,1,...,n-1 */
    int j;
    do {
        flag[i]=wantin;
        while(turn!=i)
            if(flag[turn]==idle) turn=i;
        flag[i]=incs;
        j=0;
        while(j<n&&(j==i||flag[j]!=incs))
            j++;
    } while(j<n);
    /*critical section*/;
    flag[i]=idle;
}
```

1. 空闲让进：对于进程  $i$ ，如其它进程不为incs 状态，则可以进入临界区；
2. 忙则等待：若存在进行  $j$  属于incs状态，则需要忙等等待；
3. 有限等待：进程完成临界区后，状态置为 idle
4. 让权等待：进程等待进入临界区时应该放弃CPU的使用。



## 第三章 3

- 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;  
}
```



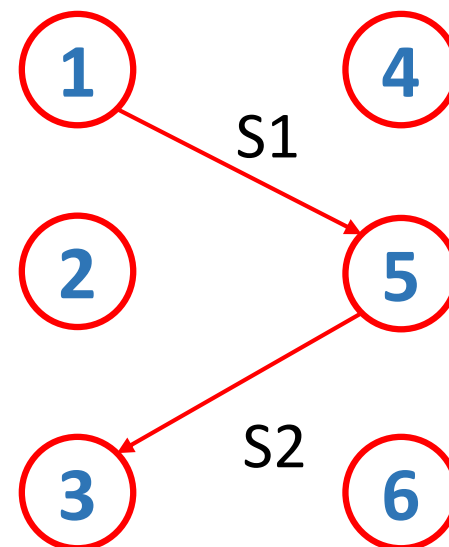
## 第三章 3

- 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  
}
```





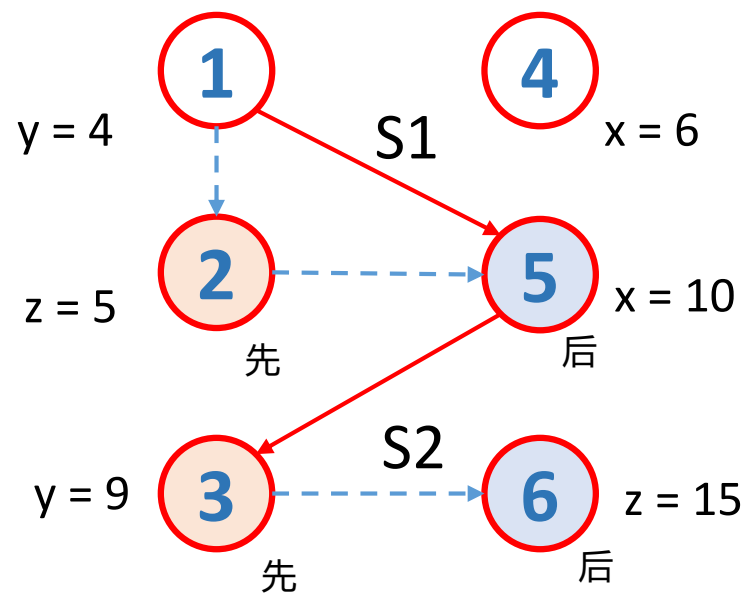
## 第三章 3

- 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 = 15**



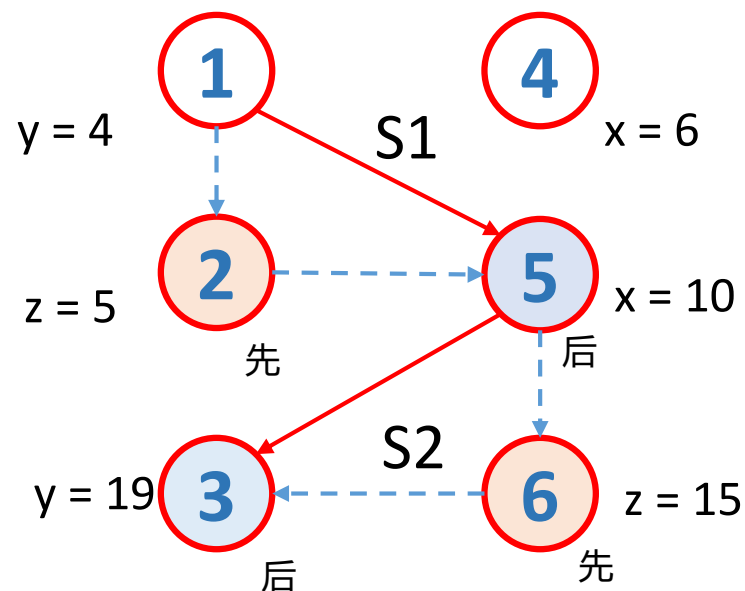
## 第三章 3

- 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**



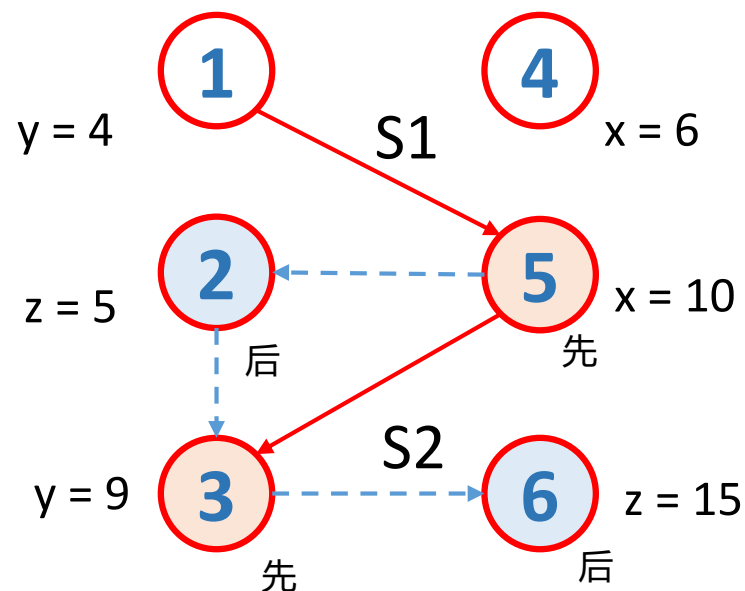
## 第三章 3

- 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 = 15**





## 第三章 3

- 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;

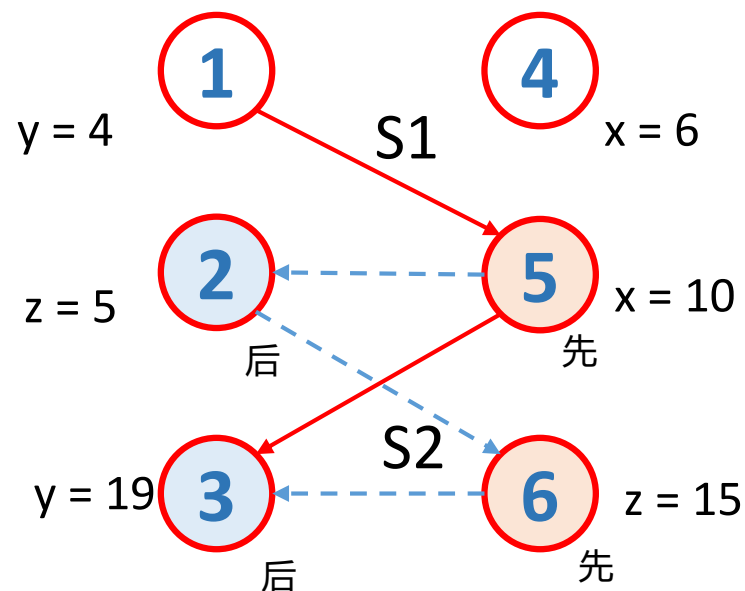
P(S1);

x = x + y;

V(S2);

z = z + x;

}



**x = 10 ; y = 19 ; z = 15**



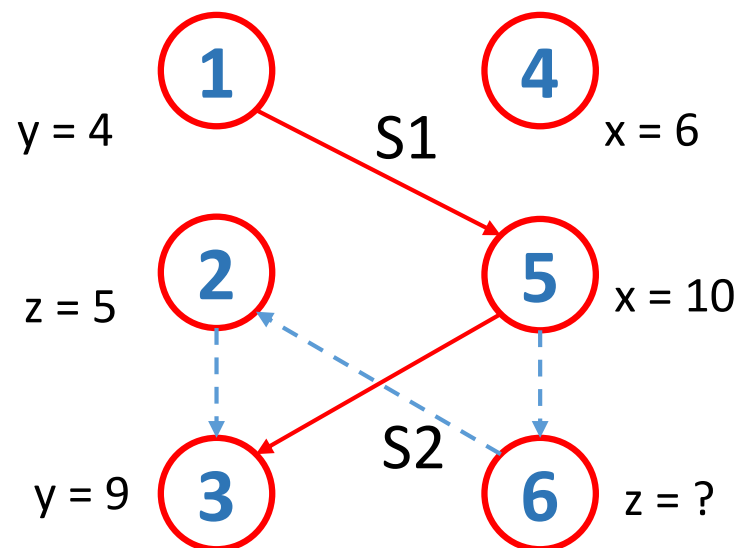
## 第三章 3

- 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**



## 第三章 5

- 有一个阅览室，读者进入时必须先在一张登记表上登记，此表为每个座位列出一个表目，包括座位号、姓名，读者离开时要注销登记信息；假如阅览室共有100 个座位。试用：
  - ①信号量和PV 操作；
  - ②管程，实现用户进程的同步算法。

```
struct {  
    char name[10];  
    int  number;  
}A[100];  
semaphore mutex = 1, seatCount = 100;  
int i = 0;
```

```
for (int i=0; i < 100; i++)  
    {A[i].number = i; A[i].name = null;}  
cobegin  
    process reader_i(char name[]){  
        P(seatCount);  
        P(mutex);  
        for (i=0;i<100;i++){  
            if (A[i].name == null) {  
                A[i].name = name;  
                break;  
            }  
        }  
        V(mutex);  
        {进入阅览室，座位号 i, 读书}  
        P(mutex);  
        A[i].name=null;  
        V(mutex);  
        V(seatCount);  
        {离开阅览室}  
    }  
coend;
```



## 第四章 7

- 在一个请求页式存储管理系统中，页表保存在寄存器中。已知处理一个未修改过的页面的缺页中断需要  $2\text{ms}$ ，处理一个已修改过的页面的缺页中断需要另加写盘时间  $8\text{ms}$ ，内存存取周期为  $1\mu\text{s}$ 。假定  $70\%$  被替换的页面曾被修改过，为了保证有效存取时间不超过  $2\mu\text{s}$ ，所允许的最大缺页中断率是多少？

设最大缺页中断率为  $x$ ，有：

$$(1 - x) * 1\mu\text{s} + (1 - 70\%) * x * (1\mu\text{s} + 2\text{ms}) + 70\% * x * (8\text{ms} + 1\mu\text{s}) = 2\mu\text{s}$$

$$x = 0.016\%$$

$$1\text{ms} = 1000\mu\text{s}$$