

第三章(1) 应用题参考答案

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 初值均为 0。试问 P1、P2 并发执行后，x、y、z 的值各为多少？

P1 () {	P2 () {
y=1;	x=1;
y=y+3;	x=x+5;
V(S1);	P(S1);
z=y+1;	x=x+y;
P(S2);	V(S2);
y=z+y;	z=z+x;
}	}

答：现对进程语句进行编号，以方便描述。

y=1;	①	x=1;	⑤
y=y+3;	②	x=x+5;	⑥
V(S1);		P(S1);	
z=y+1;	③	x=x+y;	⑦
P(S2);		V(S2);	
y=z+y;	④	z=z+x;	⑧

①、②、⑤和⑥是不相交语句，可以任何次序交错执行，而结果是唯一的。接着无论系统如何调度进程并发执行，当执行到语句⑦时，可以得到 x=10，y=4。按 Bernstein 条件，语句③的执行结果不受语句⑦的影响，故语句③执行后得到 z=5。最后，语句④和⑧并发执行，这时得到了两种结果为：

语句④先执行：x=10，y=9，z=15。

语句⑧先执行：x=10，y=19，z=15。

此外，还有第三种情况，语句③被推迟，直至语句⑧后再执行，于是依次执行以下三个语句：

z=z+x;
z=y+1;
y=z+y;

这时 z 的值只可能是 y+1=5，故 y=z+y=5+4=9，而 x=10。

第三种情况为：x=10，y=9，z=5。

9、一个快餐厅有 4 类职员：(1) 领班：接受顾客点菜；(2) 厨师：准备顾客的饭菜；(3) 打包工：将做好的饭菜打包；(4) 出纳员：收款并提交食品。每个职员可被看作一个进程，试用一种同步机制写出能让四类职员正确并发运行的程序。

答：典型的进程同步问题，可设四个信号量 S1、S2、S3 和 S4 来协调进程工作。

semaphore S1,S2,S3,S4;

S1=1;S2=S3=S4=0;

cobegin

process P1() {

process P3() {

<pre> while(true) { {有顾客到来}; P(S1); {接受顾客点菜}; V(S2); } </pre>	<pre> while(true) { P(S3); {将做好的饭菜打包}; V(S4); } </pre>
<pre> process P2() { while(true) { P(S2); {准备顾客的饭菜}; V(S3); } } </pre>	<pre> process P4() { while(true) { P(S4); {收款并提交食品}; V(S1); } } </pre>

coend

11、(总共 18 分)

(1)两个并发进程并发执行，其中，A、B、C、D 是原语，试给出可能的并发执行路径。

```

process P( ) {
    A;
    B;
    C;
}
process Q( ) {
    D;
    E;
}

```

(2) 两个并发进程 P1 和 P2 并发执行，它们的程序分别如下：

```

P1( ) {
    While(true) {
        k=k*2;
        k=k+1;
    }
}
P2( ) {
    while(true) {
        print k;
        k=0;
    }
}

```

若令 k 的初值为 5，让 P1 先执行两个循环，然后，P1 和 P2 又并发执行了一个循环，写出可能的打印值，指出与时间有关的错误。

答：

(1) 共有 10 种交错执行的路径：

A、B、C、D、E； A、B、D、E、C； A、B、D、C、E；
A、D、B、E、C； A、D、B、C、E； A、D、E、B、C；
D、A、B、E、C； D、A、B、C、E； D、A、E、B、C； D、E、A、B、C。

(2) 把语句编号，以便于描述：

```

k=k*2;    ①      print k;    ③
k=k+1;    ②      k=0;        ④

```

a) K 的初值为 5，故 P1 执行两个循环后，K=23。

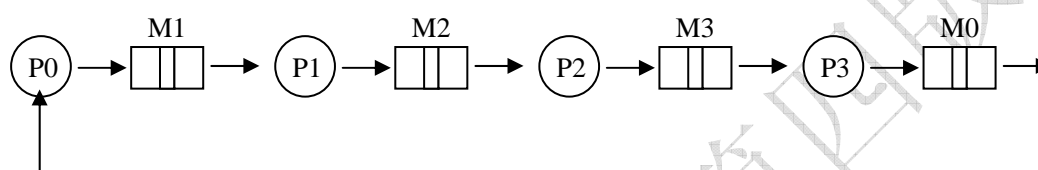
b) 语句并发执行有以下情况：

①、②、③、④，这时的打印值为：47

- ③、④、①、②，这时的打印值为：23
- ①、③、②、④，这时的打印值为：46
- ①、③、④、②，这时的打印值为：46
- ③、①、②、④，这时的打印值为：23
- ③、①、④、②，这时的打印值为：23

由于进程 P1 和 P2 并发执行，共享了变量 K，故产生了‘结果不唯一’。

19、四个进程 P_i ($i=0\cdots 3$) 和四个信箱 M_j ($j=0\cdots 3$)，进程间借助相邻信箱传递消息，即 P_i 每次从 M_i 中取一条消息，经加工后送入 $M_{(i+1)\bmod 4}$ ，其中 M_0 、 M_1 、 M_2 、 M_3 分别可存放 3、3、2、2 个消息。初始状态下， M_0 装了三条消息，其余为空。试以 P、V 操作为工具，写出 P_i ($i=0\cdots 3$) 的同步工作算法。



答：

```

semaphore mutex1,mutex2,mutex3,mutex0;
mutex1=mutex2=mutex3=mutex0=1;
semaphore empty0,empty1,empty2,empty3;
empty0=0;empty1=3;empty2=2;empty3=2;
semaphore full0,full1,full2,full3;
full0=3;full1=full2=full3=0;
int in0,in1,in2,in3,out0,out1,out2,out3;
in0=in1=in2=in3=out0=out1=out2=out3=0;
cobegin
process P0() {
    while(true) {
        P(full0);
        P(mutex0);
        {从 M0[out0]取一条消息};
        out0=(out0+1) % 3;
        V(mutex0);
        V(empty0);
        {加工消息};
        P(empty1);
        P(mutex1);
        {消息存 M1[in1]};
        in1=(in1+1) % 3;
        V(mutex1);
        V(full1);
    }
}
  
```

```
process P1() {
    while(true) {
        P(full1);
        P(mutex1);
        {从 M1[out1]取一条消息};
        out1=(out1+1) % 3;
        V(mutex1);
        V(empty1);
        {加工消息};
        P(empty2);
        P(mutex2);
        {消息存 M2[in2]};
        in2=(in2+1) % 2;
        V(mutex2);
        V(full2);
    }
}
```

```
process P2() {
    while(true) {
        P(full2);
        P(mutex2);
        {从 M2[out2]取一条消息};
        out2=(out2+1) % 2;
        V(mutex2);
        V(empty2);
        加工消息;
        P(empty3);
        P(mutex3);
        {消息存 M3[in3]};
        in3=(in3+1) % 2;
        V(mutex3);
        V(full3);
    }
}
```

```
process P3() {
    while(true) {
        P(full3);
        P(mutex3);
        {从 M3[out3]取一条消息};
        out3=(out3+1) % 2;
        V(mutex3);
        V(empty3);
        {加工消息};
        P(empty0);
    }
}
```

```

        P(mutex0);
        {消息存 M0[in0]};
        in0=(in0+1) % 3;
        V(mutex0);
        V(full0);
    }
}
coend

```

39、独木桥问题 4：在独木桥问题 1 中，要求各方向的汽车串行过桥，但当另一方提出过桥时，应能阻止对方未上桥的后继车辆，待桥面上的汽车过完桥后，另一方的汽车开始过桥。试用信号量和 P、V 操作写出汽车过独木桥问题的同步算法。

解：stop 用于当另一方提出过桥时，应阻止对方未上桥的后继车辆。

```

semaphore stop,wait,mutex1,mutex2;
stop=mutex1=mutex2=1;wait=1;
int counter1,counter2; counter1=0;counter2=0;
cobegin
    process P 东() {
        P(stop);
        P(mutex1);
        count1++;
        if (count1==1) P(wait);
        V(mutex1);
        V(stop);
        {过桥};
        P(mutex1);
        Count1--;
        if (count1==0) V(wait);
        V(mutex1);
    }
    process P 西() {
        P(stop);
        P(mutex2);
        count2++;
        if (count2==1) P(wait);
        V(mutex2);
        V(stop);
        {过桥};
        P(mutex2);
        count2--;
        if (count2==0) V(wait);
        V(mutex2);
    }
coend

```

63、某高校开设网络课程并安排上机实习，如果机房共有 2m 台机器，有 2n 个学生选课，规定：(1)每两个学生分成一组，并占用一台机器，协同完成上机实习；(2)仅当一组两个学生到齐，并且机房机器有空闲时，该组学生才能进机房；(3)上机实习由一名教师检查，检查完毕，一组学生同时离开机房。试用信号量和 P、V 操作模拟上机实习过程。

答：

```

semaphore mutex,enter;
mutex=1;enter=0;
int finish,test,rc,computercounter;
finish=test=rc=0;computercounter=2m;
cobegin
    process studenti() { //i=1, 2, ...
        P(computercounter); /*申请计算机
        P(mutex);
        rc=rc+1; /*学生互斥计数
    }
coend

```

```
    if (rc==1)    {V(mutex);P(enter);} /*若只来一个学生，则在 enter 上等待
                  else {rc=0; V(mutex);V(enter);} /*到达一组中第二个学生，rc 清 0 是为下一组计数用
{学生进入机房,上机实习};
V(finish);          /*告诉老师，实习结束
P(test);            /*等待老师检查实习结果
V(computercounter); /*归还计算机
}
process teacher( ) {
    P(finish);      /*等第一个学生实习结束
    P(finish);      /*等第二个学生实习结束
    {检查实习结果};
    V(test);        /*第一个学生检查完成
    V(test);        /*第二个学生检查完成
}
coend.
```