第三章(2) 应用题参考答案

5 有一阅览室，读者进入时必须先在一张登记表上登记，该表为每一座位列出一个表目，包括座号、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 P、V 操作；(2）管程，来实现用户进程的同步算法。

答：(1) 使用信号量和 P、V 操作：

```
struct {char name[10] ;
        int number;
    }A[100];
    semaphore mutex,seatcount;
    int i;mutex=1;seatcount=100;
  for(int i=0;i<100;i++ )
    {A[i].number=i;A[i].name=null;}
cobegin
process readeri(char readername[ ]) {                    //(i=1,2,…)
        P(seatcount);
        P(mutex);
          for (int i=1; i< 100 ;i++)
           if (A[i].name==null ) A[i].name=readername;
           reader get the seat number =i;                /*A[i].number*/
        V(mutex)
        {进入阅览室，座位号 i，座下读书};
        P(mutex);
          A[i].name=null;
        V(mutex);
        V(seatcount);
        离开阅览室;
       }
coend.
```

(2) 使用管程实现：

```
type readbook=MONTOR {
  semaphore R;
   int R_count, i, seatcount;
   char name[100] ;
   seatcount=100;
  InterfaceModule IM;
  DEFINE readbook( ),readerleave( );
  USE enter(), leave(),wait( ), signal( );
void readercome(char readername[ ]) {
      enter (IM);
        if (seatcount>=100) wait(R,R_count,IM);
        seatcount=seatcount+1;
```

```
            for (int i=0;i<100;i++) {
             if (name[i]==null ) name[i]=readername;
            }
            get the seat number=i;
          leave(IM );
    }
  void readerleave(char readername) {
        enter(IM);
            seatcount--;
            for(int i=0;i<100;i++)
                  if (name[i]==readername) name[i]=null;
            signal(R,R_count,IM);
        leave(IM);
  }
     cobegin
     process reader i ( ) {            //i=1,2.…
            readbook.readercome(readername);
            read the book;
            readbook.readerleave(readername);
            leave the readroom;
       }
coend
```

45　桌上有一只盘子，最多可以容纳两个水果，每次仅能放入或取出一个水果。爸爸向盘子中放苹果(apple)，妈妈向盘子中放桔子(orange)，两个儿子专等吃盘子中的桔子，两个女儿专等吃盘子中的苹果。试用：(1)信号量和 P、V 操作，(2)管程，来实现爸爸、妈妈、儿子、女儿间的同步与互斥关系。

答：(1)用信号量和 P、V 操作。

类似于课文中的答案，扩充如下：1) 同步信号量初值为 2；2) 要引进一个互斥信号量 mutex，用于对盘子进行互斥；3)盘子中每一项用橘子、苹果 2 个枚举值。

```
          enum {apple, orange} plate [2];
          boolean flag0,flag1;
          semaphore mutex;
          semaphore sp;                  /* 盘子里可以放几个水果 */
          semaphore sg1，sg2;            /* 盘子里有桔子，有苹果?*/
          sp=2;                         /* 盘子里允许放入二个水果*/
          sg1=sg2=0;                     /* 盘子里没有桔子，没有苹果 */
          flag0=flag1=false;mutex=1;
     cobegin
     process father（）{                    process son（）{
          while(true) {                        while(true) {
          削一个苹果;                           P(sg1);
          P(sp);                               P(mutex);
```

```
        P(mutex);
        if (flag0==false)
          {plate [0]=苹果; flag0=true;}
        else { plate[1]=苹果; flag1=true; }
        V(mutex);
        V(sg2);
        }
    }
    process mother () {
        while(true) {
        剥一个桔子;
        P(sp);
        P(mutex);
        if (flag0==false)
         {plate [0]=桔子; flag0=true;}
       else {plate[1]=桔子; flag1=true; }
        V(mutex);
        V(sg1);
        }
    }
    coend
```

(2)用管程.苹果桔子。

```
TYPE    FMSD = MONITOR
        enum {apple, orange} plate[2];
        int count; bool flag0,flag1;
        semaphore SP, SS, SD ;
        int SP_count,SS_count,SD_count;
        count=0;flag0=false;flag1=false;
        plate[0]=plate[1]=null;
InterfaceModule IM;
DEFINE put, get;
USE wait, signal, enter, leave;
void put(FRUIT fruit) {
    enter(IM);
    if(count==2)    wait(SP,SP_count,IM);
    else {if(flag0==false)
                {plate[0]=fruit;flag0=true;}
                  else {plate[1]=fruit; flag1=true; }
                  count++;
                 if(fruit==orange)    signal(SS,SS_count,IM);
                 else signal(SD,SD_count,IM);
                }
      leave(IM);
  }
```

```
        if (flag0&&plate[0]= =桔子)
          { x = plate[0];flag0=false;}
        else { x= plate[1]; flag1=false;}
        V(mutex);
        V(sp);
        吃桔子;
        }
    }
process daughter () {
        while(true) {
        P(sg2);
        P(mutex);
        if (flag0&&plate[0]==苹果)
          { x= plate[0];flag0=false;}
        else { x= plate[1]; flag1=false;}
        V(mutex);
        V(sp);
        吃苹果;
        }
    }
```

```
void get(FRUIT fruit, FRUIT &x) {
      enter(IM);
       if ((count==0) ‖ plate!=fruit)
            if (fruit==orange)    wait(SS,SS_count,IM);
           else    wait(SD,SD_count,IM);
           count--;
          if (flag0&&plate[0]= =fruit)
                {x=plate[0];flag0=false;}
          else {x=plate[1]; flag1=false;}
      signal(SP, SP_count,IM);
      leave(IM);
}
 cobegin
    process father( ) {
       while (true) {
          {准备好苹果};
          FMSD.put(apple);
        }
     }
    process mother( ) {
       while (true) {
          {准备好桔子};
          FMSD.put(orange);
        }
    }
process son( ) {
   while (true) {
       FMSD.get(orange,x);
       {吃取到的桔子};
    }
   }
process daughter( ) {
    while (true) {
        FMSD.get(apple,x);
        {吃取到的苹果};
      }
   }
coend
```

46　一组生产者进程和一组消费者进程共享九个缓冲区，每个缓冲区可以存放一个整数。生产者进程每次一次性向 3 个缓冲区写入整数，消费者进程每次从缓冲区取出一个整数。请用：(1)信号量和 P、V 操作，(2)管程，写出能够正确执行的程序。

答：(1)信号量和 P、V 操作。

```
var    int buf[9];
        int count,getptr,putptr;
        count=0;getpt=0;putpt=0;
        semaphoreS1,S2,SPUT,SGET;
        S1=1;S2=1;SPUT=3;SGET=0;
   main( ) {
      cobegin
        producer-i();consumer-j();
      coend
  }
process producer-i( ) {
         while(true) {
          {生产 3 个整数};
            P(SPUT);
            P(S1);
            buf[putptr]=整数 1;
            putptr=(putptr+1) % 9;
            buf[putptr]=整数 2;
            putptr=(putptr+1) % 9;
            buf[putptr]=整数 3;
            putptr=(putptr+1) % 9;
            V(SGET);
            V(SGET);
            V(SGET):
            V(S1);
        }
   }
process consumer-j( ) {
          int y;
          while(true) {
            P(SGET);
            P(S2);
            y=buf[getptr];
            getptr=(getptr+1) % 9;
            count++;
            if (count==3)
                 {count=0;V(SPUT);}
            V(S2);
            {consume the  整数 y};
            }
        }
```

(2) 管程/生产者消费者。

```
TYPE    get_put = MONITOR
          int buf [9];
```

```
        int count,getptr,putptr;
        semaphore SP,SG;
        int SP_count,SG_count;
       count:=0;getptr:=0;putptr:=0;
InterfaceModule IM;
DEFINE put, get;
USE wait, signal, enter, leave;
void put(int a1,int a2,int a3) {
     enter(IM);
   if (count>6)   wait(SP,SP_count,IM);
   count=count+3;
   buf[putptr]=a1;
   putptr=(putptr+1) % 9;
   buf[putptr]=a2;
   putptr=(putptr+1) % 9;
   buf[putptr]=a3;
   putptr=(putptr+1) % 9;
  signal(SG,SG_count,IM);
  leave(IM);
}
void get(int b) {
     enter(IM);
      if (count= =0)   wait(SG,SG_count,IM);
      b=buf[getptr];
      getptr=(getptr+1) % 9;
      count=count++;
      if (count < 7)   signal(SP,SP_count, IM);
      else if (count > 0)   signal(SG,SG_count,IM);
      leave(IM);
  }
  cobegin
    process producer-i( ) {
      while(true) {
      {生产 3 个整数};
      get-put.put(a1,a2,a3);
    }
  }
  process consumer-j( ) {
       while(true) {
       get-put.get(b)
       {consume the  整数 b};
       }
    }
  coend
```

53　现有三个生产者 P1、P2、P3，他们都要生产桔子水，每个生产者都已分别购得两种不同原料，待购得第三种原料后就可配制成桔子水，装瓶出售。有一供应商能源源不断地供应糖、水、桔子精，但每次只拿出一种原料放入容器中供给生产者。当容器中有原料时需要该原料的生产者可取走，当容器空时供应商又可放入一种原料。假定：

生产者 P1 已购得糖和水；

生产者 P2 已购得水和桔子精；

生产者 P3 已购得糖和桔子精；

试用：1)管程，2)信号量与 P、V 操作，写出供应商和三个生产者之间能正确同步的程序。

答：1)管程。

```
TYPE makedrink=monitor
        semaphore S,S1,S2,S3;
        int S_count,S1_count,S2_count,S3_count;
        enum {糖,水,桔子精} container;
InterfaceModule IM;
DEFINE give，produce1，produce2，produce3;
USE enter,wait,signal,leave;
void give( ) {
          enter(IM);
          take material;
          if (container!=null)    wait(S,S_count,IM);
          else container=material;
          if(container==桔子精)    signal(S1,S1_count,IM);
             else if(container==糖)    signal(S2,S2_count,IM);
                    else signal(S3,S3_count,IM);
          leave(IM);
     }
void produce1( ) {
           enter(IM);
           if (container!=桔子精)    wait(S1,S1_count,IM);
            else {take  桔子精  from container;  做桔子水;}
           signal(S,S_count,IM);
           leave(IM);
     }
void produce2( ) {
           enter(IM);
           if (container!=糖)    wait(S2,S2_count,IM);
            else { take  糖 from container;做桔子水;}
           signal(S,S_count,IM);
           leave(IM);
      }
void produce3( )    {
           enter(IM);
           if (container!=水)    wait(S3,S3_count,IM);
```

```
            else { take  水 from container;  做桔子水;}
            signal(S,S_count,IM);
            leave(IM);
        }
    cobegin
    process 供应商( ) {
      while(true) {
        makedrink.give();
        }
     }
    Process P1( ) {
      while(true) {
          makedrink.produce1();
        }
     }
    Process P2( ) {
      while(true) {
          makedrink.produce2();
          }
     }
    Process P3( ) {
      while(true) {
          makedrink.produce3();
        }
    }
    coend.
```

2) 信号量与 P、V 操作

```
    semaphore S,S1,S2,S3;
    S=1;S1=S2=S3=0;
    enum container{糖,水,桔子精} ;
    cobegin
     process  供应商( ) {
            while(true) {
              P(S);
              {take material into container};
              if(container==桔子精)    V(S1);
              else    if(container==糖)    V(S2);
                      else    V(S3);
              }
        }
process P1( ) {
            while(true) {
              P(S1);
```

```
            {take 桔子精 from container};
            V(S);
            {做桔子水};
            }
      }
process P2( ) {
            while(true) {
            P(S2);
            {take 糖 from container};
            V(S);
            {做桔子水};
           }
      }
process P3( ) {
           while(true) {
            P(S3);
            {take 水 from container};
            V(S);
            {做桔子水};
            }
      }
coend.
```

54 有一材料保管员，他保管纸和笔若干。有 A、B 两组学生，A 组学生每人都备有纸，B 组学生每人都备有笔。任一学生只要能得到其他一种材料就可以写信。有一个可以放一张纸或一支笔的小盒，当小盒中无物品时，保管员就可任意放一张纸或一支笔供学生取用，每次允许一个学生从中取出自己所需的材料，当学生从盒中取走材料后允许保管员再存放一件材料，请用：1)信号量与 P、V 操作，2)管程，写出他们并发执行时能正确工作的程序。

答：1)信号量与 P、V 操作。

```
var   semaphore S,Sa,Sb,mutexa,mutexb;
      S=mutexa=mutexb=1;Sa=Sb=0;
      enum {paper,pen} box;
   cobegin
      process 保管员( ) {
          while(true) {
            P(S);
            {take a material into box};
            If((box)==paper)   V(Sa);
            else   V(Sb);
           }
          }
     process A 组学生( ) {
          while(true) {
               P(Sa);
```

```
            P(mutexa);
            {take the pen from box};
            V(mutexa);
            V(S);
            {write a letter};
            }
        }
    process B 组学生( ) {
        while(true) {
            P(Sb);
            P(mutexb);
            {take the paper from box};
            V(mutexb);
            V(S);
            {write a letter};
          }
        }
  coend
```

2) 管程.材料保管员。

```
TYPE   paper&pen=monitor
       semaphore S,S1,S2;
       int S_count，S1_count，S2_count;
      enum box {paper,pen}; box=null;
InterfaceModule IM;
DEFINE   put，get1,get2;
USE   enter,wait,signal,leave;
void put( ) {
       enter(IM);
       {take a material};
       if((box)!=null)   wait(S,S_count,IM);
       else   box=material;
       if ((box)==pen)   signal(S1,S1_count,IM);
         else signal(S2,S2_count,IM);
       leave(IM);
     }
 void get1( ) {
        enter(IM);
        if ((box)==null)  ‖  (box)!=pen)   wait(S1,S1_count,IM);
          else { take pen from box};
        signal(S,S_count,IM);
        leave(IM);
    }
void get2( ) {
```

```
            enter(IM);
            if((box)==null ‖ (box)!=paper)   wait(S2,S2_count,IM);
              else   { take the paper from box;}
            signal(S,S_count,IM);
            leave (IM);
      }
cobegin
process 保管员( ) {
      while(true)   paper&pen.put( );
   }
process A 组学生( ) {
      while(true) { paper&pen.get( );
             写信;
          }
      }
process B 组学生( ) {
      while(true) { paper&pen.get( )
             写信;
           }
      }
coend
```

59 设儿童小汽车生产线上有一只大的储存柜，其中有 N 个槽（N 为 5 的倍数且其值≥5），每个槽可存放一个车架或一个车轮。设有三组生产工人，其活动如下：

组 1 工人的活动	组 2 工人的活动	组 3 工人的活动
L1：加工一个车架； 车架放入柜的槽中。	L2：加工一个车轮； 车轮放入柜的槽中。	L3：在槽中取一个车架； 在槽中取四个车轮， 组装为一台小汽车。
goto L1:;	goto L2:;	goto L3:;

试用(1)信号量及 P，V 操作，(2)管程方法正确实现这三组工人的生产合作工作。

解：(1) 信号量及 P，V 操作

将柜子的 n 个槽口分为两部分：N/5 和 4N/5，分别装入车架和车轮

```
车架  box1[N/5];
车轮  box2[4*N/5] ;
semaphore mutex1,mutex2,S1,S2,S3,S4;
int counter,in1,in2,out1,out2;
S1=N/5;S2=4N/5;S3=S4=0;
counter=in1=in2=out1=out2=0;
mutex1=mutex2=1;
cobegin
process worker1( ) {
      while(true) {
```

```
            加工一个车架;
            P(S1);
            P(mutex1);
            {车架放入 box1(in1)};
            in1=(in1+1) % (N/5);
            V(mutex1);
            V(S3);
            }
        }
process worker2( ) {
            while(true) {
            加工一个车轮;
            P(S2);
            P(mutex2);
            {车轮放入 box2(in2)};
            in2=(in2+1) % (4*N/5);
            counter=counter+1;
            if(counter==4)    {counter=0;V(S4);}
            V(mutex2);
            }
        }
process worker3( ) {
            while(true) {
            P(S3);
            P(mutex1);
            取车架从 box1(out1);
            out1=(out1+1) % (N/5);
            V(mutex1)
            V(S1);
            P(S4);
            P(mutex2);
            取车轮从 box2(out2);
            out2=(out2+1) % (4*N/5);
            取车轮从 box2(out2);
            out2=(out2+1) % (4*N/5);
            取车轮从 box2(out2);
            out2=(out2+1) % (4*N/5);
            取车轮从 box2(out2);
            out2=(out2+1) % (4*N/5);
            V(S2);
            V(mutex2);
            装配车子;
            }
        }
```

coend

(2)用管程方法.汽车厂。

```
type    produce_toy_car=monitor
          车架 box1[N/5];
          车轮 box2[4*N/5];
          semaphore S1,S2,S3,S4;
          int S1_count,S2_count,S3_count,S4_count;
          int counter1,counter2,count,in1,in2,out1,out2;
          counter1=counter2=count=in1=in2=out1=out2=0;
InterfaceModule IM;
DEFINE put1,put2,take;
USE wait,signal,enter,leave;
   void put1( ) {
      enter(IM);
      if(counter1==N/5)    wait(S1,S1_count,IM);
      {车架放入 box1(in1)};
      in1=(in1+1) % (N/5);
      counter1=counter1+1;
      signal(S3,S3_count,IM);
      leave(IM);
    }
   void put2( ) {
         enter(IM);
         if(counter2==(4*N/5))    wait(S2,S2_count,IM);
         车轮放入 box2(in2);
         in2=(in2+1) % (4*N/5);
         counter2=counter2+1;
         count=count+1;
         if(count==4)    {count=0; signal(S4,S4_count,IM);}
          leave(IM);
      }
    void take( ) {
          enter(IM);
         if(counter1==0)    wait(S3,S3_count,IM);
         取车架从 box1(out1);
         out1=(out1+1) % (N/5);
         counter1=counter1-1;
         if (counter2<4)    wait(S4,S4_count,IM);
         取车轮从 box2(out2);
         out2=(out2+1) % (4*N/5);
         取车轮从 box2(out2);
         out2=(out2+1) % (4*N/5);
```

```
    取车轮从 box2(out2);
    out2=(out2+1) % (4*N/5);
    取车轮从 box2(out2);
    out2=(out2+1) % (4*N/5);
    counter2=counter2-4;
    signal(S1,S1_count,IM);
    signal(S2,S2_count,IM);
    leave(IM);
}
```