**飞机票**

Var A : ARRAY[1..m] of integer;

mutex : semaphore

;

mutex:= 1;

cobegin

process Pi

var Xi:integer;

begin

L1:

按旅客定票要求找到A[j];

P(mutex);

Xi := A[j];

if Xi>=1 then

begin

Xi:=Xi

-1;A[j]:=Xi;

V(mutex);

{输出一张票

}

；

end;

else begin

V(mutex);

{输出

“票已售完

”

}

；

end;

goto L1;

end;

coend

**哲学家进餐问题**

semaphore fork[5];

for (int i=0;i<5;i++)

fork[i]= 1;

semaphore room=4; //增加一个侍者

cobegin

process philosopher\_i( ){/\*i=0,1,2,3 \*/

while(true) {

think( );

P(room); //控制最多允许4为哲学家取叉子

P(fork[i];

P(fork[(i+1)%5] ) ;

eat( );

V(fork[i]);

V(fork([i+ 1] % 5);

V(room);

}

}

Coend

**生产者/消费者问题**

B : ARRAY[0..k-1] of integer;

sput: semaphore； /\* 可以使用的空缓冲区数 \*/

sget: semaphore; /\* 缓冲区内可以使用的产品数 \*/

sput := k; /\* 缓冲区内允许放入k件产品 \*/

sget := 0; /\* 缓冲区内没有产品 \*/

putptr, getptr : integer; putptr:=0; getptr:=0;

process producer

begin

L1: produce a product；

P(sput);

B[putptr] := product；

putptr :=(putptr+1) mod k;

V(sget);

goto L1;

end;

process consumer

begin

L2: P(sget);

Product:= B[getptr]；

getptr:=(getptr+1) mod k;

V(sput);

consume a product；

goto L2;

End

**苹果-桔子**

plate : integer;

sp:semaphore； /\* 盘子里可以放几个水果 \*/

sg1:semaphore; /\* 盘子里有桔子 \*/

sg2:semaphore; /\* 盘子里有苹果 \*/

sp := 1; /\* 盘子里允许放入一个水果\*/

sg1 := 0; /\* 盘子里没有桔子 \*/

sg2 := 0; /\* 盘子里没有苹果\*/

process father

begin

L1: 削一个苹果；

P(sp);

把苹果放入plate；

V(sg2);

goto L1;

end;

process mother

begin

L2: 剥一个桔子；

P(sp);

把桔子放入plate；

V(sg1);

goto L2;

end;

process son

begin

L3: P(sg1);

从plate中取桔子；

V(sp);

吃桔子；

goto L3;

end;

process daughter

begin

L4: P(sg2);

从plate中取苹果；

V(sp);

吃苹果；

goto L4;

End

**读者写者写者优先**

semaphore rmutex，wmutex，S；

rmutex=1; wmutex=1; S=1; //增加互斥信号量S

int readcount=0; //读进程计数

process reader\_i( ) {

while (true) {

P(S);

P(rmutex);

if (readcount==0) P(wmutex);

readcount++;

V(rmutex);

V(S);

读文件；

P(rmutex);

readcount--;

if(readcount==0) V(wmutex);

V(rmutex);

}

}

process writer\_i( ) {

while(true) {

P(S);

P(wmutex);

写文件；

V(wmutex);

V(S);

}

}

**睡眠的理发师**

int waiting=0; //等候理发顾客坐的椅子数

int CHAIRS=N; //为顾客准备的椅子数

semaphore customers, barbers, mutex;

customers=0; barbers=0; mutex=1;

process barber( ) {

while(true) {

*P(customers);*

//有顾客吗?若无顾客,理发师睡眠

P(mutex);

//若有顾客时，进入临界区

waiting--; //等候顾客数少一个

V(barbers); //理发师准备为顾客理发

V(mutex); //退出临界区

cut\_hair();

//理发师正在理发(非临界区)

}

}

process customer\_i( ) {

P(mutex); //进入临界区

if(waiting<CHAIRS) {

//有空椅子吗

waiting++; //等候顾客数加1

V(customers); //唤醒理发师

V(mutex); //退出临界区

P(barbers);

//理发师忙，顾客坐下等待

get\_haircut(); //否则顾客坐下理发

}

else V(mutex); //人满了,走吧！

}

1. **农夫猎人**

semaphore Scage=1;

semaphore Stiger=0;

semaphore Ssheep=0;

void hunter()

{

while (true) {

………

P(Scage);

将虎放入笼中；

V(Stiger);

}

}

void peasant()

{

while (true) {

………

P(Scage);

将羊放入笼中；

V(Ssheep);

}

}

void hotel()

{

while (true) {

P(Ssheep);

将羊取出笼中；

V(Scage);

……….

}

}

void zoo()

{

while (true){

P(Stiger);

将虎取出笼中；

V(Scage);

……….

}

}

void main()

{

parbegin(hunter, peasant, hotel, zoo);

}

**售票问题**

Var run1, run2, stop1, stop2: semaphore;

run1:=0; run2:=0; stop1:=0; stop2:=0;

void Driver() {

while (true)

{

P(run1);

P(run2);

开车;

停车;

V(stop1);

V(stop2);

}

}

void Seller1() {

while (true) {

上乘客;

关车门;

V(run1);

售车票;

P(stop1);

开车门;

下乘客;

}

}

void Seller2() {

while (true) {

上乘客;

关车门;

V(run2);

售车票;

P(stop2);

开车门;

下乘客;

}

}

void main() {

parbegin(Driver; Seller1; Seller2);

}

**吸烟者问题**

semaphor:s0,s1,s2,s3;

S0=1;S1=0;S2=0;S3=0;

Process businessman {

//供应者进程

L1：i:=RAND( ) mod 3;

j:=RAND( ) mod 3;

If (i=j) then goto L1;

P(S0);

Put\_items [i]\_on\_table;

Put\_items [j]\_on\_table;

if (i=0 and j=1) or (i=1 and j=0) V(S[3]);

if (i=1 and j=2) or (i=2and j=1) V(S[1]);

if (i=0 and j=2) or (i=2 and j=0)

V(S[2]);

goto L1;

}

Process consumer (k) {

//吸烟者进程，k=1,2,3

L1：

P(S[k]);

take\_one\_item\_from\_table;

take\_one\_item\_from\_table;

V(S0);

make\_cigarette\_and\_smokeing

goto L1;

}

**缓冲区管理**

var mutex,empty,full:semaphore;

count,in:integer

buffer:array[0..79] of char;

mutex=1;empty=80;full=0;

count=0;in=0;

process Pi(i=1,...,n))

begin

L: 读入一字符到x;

P(empty);

P(mutex);

Buffer[in]=x;

in=(in+1) % 80;

count++;

if (count==80)

{count=0; V(mutex); V(full); }

else V(mutex);

goto L;

end;

process Q

begin

while(true) {

P(full);

P(mutex);

for(int j=0; j< 80;j++)

read buffer[j];

in:=0;

V(mutex);

for (int j=0; j< 80;j++)

V(empty);

}

end;

**银行业务问题**

var customer\_count, server\_count, mutex: semaphore;

customer\_count:=0; server\_count:=n;

mutex:=1;

process customeri(i=1,2,….)

begin

take a number;

P(mutex);

等待区找到空沙发坐下;

V(mutex);

V(customer\_count);

P(server\_count);

end;

Process servers j(j=1,2,3,…)

Begin

L: P(customer\_count);

P(mutex);

被呼号顾客离开沙发走出等待区;

V(mutex);

为该号客人服务;

客人离开;

V(server\_count);

go to L;

end;