1. 在公共汽车不断地到站、停车、行驶过程中，司机和售票员的活动分别如下：（1）司机：启动车辆；正常行车；到站停车；（2）售票员：关车门；售票；开车门。请用记录型信号量机制描述司机和售票员的同步关系。

答：

semaphore S1=0; //表示是否允许司机启动车辆，其初值为0；

semaphore S2=0; //表示是否允许售票员开车门，其初值为0。/\*定义信号量及初值\*/

driver(){ /\*司机描述\*/ Seller(){ /\*售票员描述\*/

while(1){ while(1){

wait(S1); 关车门;

启动车辆; signal(S1);

正常行车; 售票；

到站停车; wait(S2);

signal(S2); 开车门;

} }

} }

main(){

cobegin

Seller(); driver();

coend

}

1. “过独木桥”问题：同一方向的行人可连续过桥，当某一方向有人过桥时，另一方向的行人必须等待；当某一方向无人过桥时，另一方向的行人可以过桥。请用记录型信号量机制描述两个方向行人的同步关系。

答：

/\*定义信号量及初值\*/

//将独木桥的两个方向分别标记为A和B；

int countA=0,countB=0; //并用整形变量countA和countB分别表示A和B方向上已在独木桥上的行人数，它们的初值为0；

semaphore SA=1; //用信号量SA来实现对countA的互斥访问，其初值为1；

semaphore SB=1; //用信号量SB来实现对countB的互斥访问，其初值为1；

semaphore mutex=1; //用信号量mutex来实现两个方向的行人对独木桥的互斥使用。

A(){ /\*A方向描述\*/

while(1){

wait(SA); //等互斥资源countA可用

if(countA==0) wait(mutex); //如A方向无人过桥，则竞争桥方向

countA++; //人数+1

signal(SA); //释放互斥资源

通过独木桥;

wait(SA); //等互斥资源countA可用

countA--; //过完桥了，人数-1

if(countA==0) signal(mutex); //没人了，释放桥资源

signal(SA); //释放

}

}

B(){ /\*B方向描述\*/

while(1){

wait(SB);

if(countB==0) wait(mutex);

countB++;

signal(SB);

通过独木桥;

wait(SB);

countB--;

if(countB==0) signal(mutex);

signal(SB);

}

}

1. 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：信号量和P 、V 操作，来实现用户进程的同步算法。

答：/\*定义信号量及初值\*/

string A[100]; //登记表，座位号0-99，存储座位上读者的名字

semaphore mutexA=l; //mutexA座位数组A互斥信号量，初始值1代表可用

semaphore seatcount=100; //seatcount空余的座位数，初始值100

reader(string readername){ //读者进程，传递读者名

int readernumber; //读者座位号

wait(seatcount); //等有空座位，座位-1

wait(mutexA); //等座位数组A可用

for(i=0;i<100;i++) //找空座，并占位

if(A[i].name==NULL)

{A[i].name=readername; break;}

readernumber=i;

signal(mutexA); //置座位数组A可用

读者进入阅览室，作在readernumber号座位读书;

读者准备离开阅览室

wait(mutexA); //等座位数组A可用

A[readernumber]=NULL; //离开座位，空出座位

signal(mutexA); //置座位数组A可用

signal(seatcount); //座位+1，唤读者

读者离开阅览室；

}

main(){

for(i=0;i<100;i++) A[i].name=null;

cobegin

while(1){

readername=新来的读者名;

reader(readername);

}

coend

}

1. 学校羽毛球馆，馆内提供羽毛球拍和羽毛球若干。有A、B两组学生，A组学生每人都备有羽毛球拍，B组学生每人都备有羽毛球。任意一组学生只要能得到其他一种材料就可以打球。有一个可以存放一个球拍或一个羽毛球的筐子，当筐子中无物品时，管理员就随机的放一个球拍或一个羽毛球供学生取用，每次允许一个学生从中取出自己所需的材料，当学生从筐子中取走材料后允许管理员再放一件材料，请用信号量与P、V操作。

semaphore s, sa , sb, mutexa , mutexb ;   
s =1;mutexa =1;mutexb=1; sa =0; sb= 0 ;   
box (Racket , badminton ) ;

cobegin   
{   
process 保管员  
begin   
repeat   
P ( s ) ;   
take a material intobox ;   
if ( box ) = badminton then V ( sa ) ; //放球通知A组  
else V( sb ) ; //放球拍通知B组  
untile false ;   
end

Process A组学生  
begin   
repeat   
P ( sa ) ;   
P ( mutexa ) ;   
take the badminton from box ;   
V ( mutexa ) ;   
V ( s ) ;   
write a letter;  
untile false ;   
end

Process B组学生  
begin   
repeat   
P ( sb ) ;   
P ( mutexb ) ;   
take the Racket from box ;  
V ( mutexb ) ;   
V ( s ) ;   
write a letter ;   
untile false ;   
end   
}   
Coend

5. 某缴费中心提供1个服务窗口和5个供顾客等待的座位，顾客到达该中心时，若有空座位，则到取号机上领取一个号，等待叫号。取号机每次仅允许一位顾客使用。当收银员空闲时，通过叫号选取一位顾客，并为其服务。顾客和收银员的活动表述如下：

Process 顾客：

{

从取号机获取一个号码；

等待叫号；

获取服务；

}

Process 收银员

{

叫号；

为顾客服务；

}

请添加必要的信号量和P、V(或者wait() 和signal())操作，实现上述过程中的互斥与同步。要求写出完整的过程，说明信号量的含义并赋初值。

**答**：/\*信号量定义\*/

#define N 5

item seat[N]; //5个座位

int in=0,out=0;

semaphore emptyseats=N; //表示空余座位数量的资源信号量，初值为5。

semaphore fullcustomers=0; //表示等待服务顾客数量的资源信号量，初值为0。

semaphore mutexA=1; //管理取号机的互斥信号量，初值为1，表示取号机空闲。

semaphore mutexS=1; //管理共享资源座位seat的互斥信号量，初值为1，表示空闲。

customer(){ //顾客进程

while(1){

wait(emptyseats); //等空位

wait(mutex); //等取号机，互斥资源

cid=从取号机获取一个号码;

signal(mutex); //释放取号机

wait(mutexS); //等seat，互斥资源

seat[in]=cid; in=(in+1)%N; //顾客做进座位

signal(mutexS); //释放seat

signal(fullcustomers); //唤收银员

等待叫号；

获取服务；

}

}

cashier(){ //收银员进程

while(1){

wait(fullcustomers); //等顾客

wait(mutexS); //等seat，互斥资源

cid=seat[out]; out=(out+1)%N; //从座位上读取号码

signal(mutexS); //释放seat

叫号;

为顾客服务;

signal(emptyseats); //服务完成，唤顾客离开空出座位

}

}

main(){

cobegin

cashier(); while(1){ customer();}

coend

}

6. 哲学家进餐问题。规定奇数号哲学家先拿他左边的筷子，然后再去拿右边的筷子；而偶数号哲学家则相反。按此规定，将是1、2号哲学家竞争1号筷子；3、4号哲学家竞争3号筷子。即五位哲学家都先竞争奇数号筷子，获得后，再去竞争偶数号筷子，最后总会有一位哲学家能获得两只筷子而进餐。试用信号量与P、V操作完成上述解决办法。

semaphore chopstick[5]={1,1,1,1,1};

process i

{ while(ture)

{ think();

if(i%2!=0)

{ P (chopstick[i]);

P (chopstick[i+1]%5);}

else

{ P (chopstick[i+1]%5);

P (chopstick[i]);}

eat();

V (chopstick[i]);

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

}

}

7. 进程PC和打印进程PO1、PO2共享一个单缓冲区。计算进程负责计算，并把计算结果放入单缓冲区中。打印进程PO1、PO2负责从单缓冲区中取出计算结果进行打印，而且对于每一个计算结果，PO1和PO2都需分别打印一次。请用记录型信号量机制描述上述三个进程之间的同步关系。

答：/\*定义信号量\*/

semaphore full1=0; //表示缓冲区中是否有可供PO1打印的计算结果，其初值为0；

semaphore full2=0; //表示缓冲区中是否有可供PO2打印的计算结果，其初值为0；

semaphore empty1=1; //表示计算结果是否已被PO1取走，其初值为1；

semaphore empty2=1; //表示计算结果是否已被PO2取走，其初值为1。

PC(){ /\*PC描述\*/

while(1){

计算;

wait(empty1); //等po1取

wait(empty2); //等po2取

计算结果放入缓冲;

signal(full1); //唤po1

signal(full2); //唤po2

}

}

PO1(){ /\*PO1描述\*/

while(1){

wait(full1); //等结果，po1可用

取出计算结果;

signal(empty1); //唤pc，po1用完

打印结果;

}

}

PO2(){ /\*PO2描述\*/

while(1){

wait(full2); //等结果，po2可用

取出计算结果;

signal(empty2); //唤pc，po2用完

打印结果;

}

}

8.有一体育器材保管员，他网球拍和网球若干。有A、B两组学生，A组学生每人都备有网球拍，B组学生每人都备有网球。任意一组学生只要能得到其他一种材料就可以打网球。有一个可以放一个球拍或一个网球的筐子，当筐子中无物品时，保管员就可任意放一个球拍或一个网球供学生取用，每次允许一个学生从中取出自己所需的材料，当学生从筐子中取走材料后允许保管员再放一件材料，请用信号量与P、V操作实现。

semaphore s, sa , sb, mutexa , mutexb ;   
s =1;mutexa =1;mutexb=1; sa =0; sb= 0 ;   
box (Racket , Tennis ) ;

cobegin   
{   
process 保管员  
begin   
repeat   
P ( s ) ;   
take a material intobox ;   
if ( box ) = Tennis then V ( sa ) ; //放球通知A组  
else V( sb ) ; //放球拍通知B组  
untile false ;   
end

Process A组学生  
begin   
repeat   
P ( sa ) ;   
P ( mutexa ) ;   
take the Tennis from box ;   
V ( mutexa ) ;   
V ( s ) ;   
write a letter;  
untile false ;   
end

Process B组学生  
begin   
repeat   
P ( sb ) ;   
P ( mutexb ) ;   
take the Racket from box ;  
V ( mutexb ) ;   
V ( s ) ;   
write a letter ;   
untile false ;   
end   
}   
Coend