**一、对N个共享缓冲区读写问题**

解：

1. 定性：该问题为互斥+同步问题
2. 角色：该问题可确定为两个进程：一个写数据的进程（reader），一个取数据的进程(writer)
3. 确定信号量：

1）s1表示空缓冲区的数目，初值为N

2）s2表示已用缓冲区的数目，初值为0

3）互斥信号量mutex，初值为1

Semaphore s1=N,s2=0,mutex=1;

reader(){

while(1){

生产数据;

P(s1);

P(mutex);

将数据写入缓冲区;

V(mutex);

V(s2);

}

}

Writer(){

While(1){

P(s2);

P(mutex);

从缓冲区读取数据;

V(mutex):

V(s1);

}

}

Void main(){

s1=N;

s2=0;

mutex=1;

cobegin

reader();

writer():

coend

}

**二、 对盘中可放N个水果的吃水果问题**

问题描述：桌上有一空盘，最多允许存放N个水果。爸爸可向盘中放一个苹果或放一个桔子，儿子专等吃盘中的桔子，女儿专等吃苹果。

1. 定性：该问题为互斥+同步问题
2. 角色：该问题可确定为3个进程：父亲往盘子中放水果（Father），儿子只吃桔子（Son）,女儿只吃苹果（Daughter）
3. 确定信号量：
4. S表示盘子上可放水果的剩余空间数目，初值为N
5. So表示盘子中桔子的数目，初值为0
6. Sa表示盘子中苹果的数目，初值为0
7. 互斥信号量mutex，初值为1

Semaphore S=N,So=0,Sa=0,mutex=1;

Father(){

while(1){

P(S);

P(muex);

往盘子放水果;

V(mutex);

if(水果是桔子)

V(So);

else

V(Sa);

}

}

Son(){

while(1){

P(So);

P(mutex);

取桔子;

V(mutex);

V(S);

吃桔子;

}

}

Daughter(){

while(1){

P(Sa);

P(mutex);

取苹果;

V(mutex);

V(S);

吃苹果;

}

}

Void main(){

S=N;

So=0;

Sa=0;

mutex=1;

cobegin

Father();

Son();

Daughter();

coend

}