【例1】 桌上有1空盘，允许存放1个水果。爸爸向盘中放苹果，也可以向盘中放桔子。儿子专等吃盘中的桔子，女儿专等吃盘中的苹果。规定当盘空时一次只能放1个水果供吃者取用。请用Wait()、Signal()原语实现爸爸、儿子、女儿三个并发进程的同步。

【南京大学2000】

【分析】这是复杂情况的“生产者—消费者”问题，既有同步又有互斥。爸爸进程与儿子进程、女儿进程需要同步，儿子进程与女儿进程需要互斥。设置4个信号量S（盘子是否为空，初值为1）、So（盘中是否有桔子，初值为0）、Sa（盘中是否有苹果，初值为0）和mutex（用于对盘子的互斥访问，初值为1）。由于只有一个盘子（相当于只有一个buffer），对盘子的互斥访问发生在对盘子的存取操作上，S、So和Sa就可以保证对盘子的互斥操作了，故mutex也可以省略。

解：设三个信号量：

S --- 盘子是否为空，初值为1；

So --- 盘中是否有桔子，初值为0；

Sa --- 盘中是否有苹果，初值为0；

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

Main() {

Cobegin

Father();

Son();

Daughter();

Coend

}

Father() {

While(1) {

Wait(S); 将水果放入盘中；

If (放入的是桔子) Signal(So);

Else Signal(Sa);

}

}

Son() {

While(1) {

Wait(So); 从盘中取出桔子；Signal(S); 吃桔子；

}

}

Daughter() {

While(1) {

Wait(Sa); 从盘中取出苹果；Signal(S); 吃苹果；

}

}

【例2】桌上有1空盘，允许存放N个水果。爸爸每次向盘中放1个苹果或1个桔子。多个孩子，每个孩子每次只能取1个苹果和1个橘子。 请用wait( )、signal( )原语实现爸爸、孩子并发进程的同步。

答：

semaphore empty=N; //盘中空闲容量

semaphore Apple=0; //盘中苹果数量

semaphore Orange=0; //盘中橘子数量

semaphore sA= N-1; //盘中**尚能放**苹果的最大数量。

semaphore sO= N-1; //盘中**尚能放**橘子的最大数量。

semaphore mutex=1; //互斥使用盘子

**爸爸的活动:**

do{

While(1) {

if 拿到一个苹果 {

P(sA);

P(empty);

P(mutex);

将苹果放入盘中；

Signal(mutex);

Signal(Apple);

}

else if 拿到1个橘子 {

P(sO);

P(empty);

P(mutex);

将橘子放入盘中；

Signal(mutex);

Signal(Orange);

}

}while(1)

**孩子的活动:**

do{

Wait(Apple);

Wait(Orange);

Wait(mutex);

取1个苹果; 取1个橘子；

Signal（mutex）;

Signal（empty）;

Signal（empty）;

Signal（sA）;

Signal（sO）;

}while(1)

【例3】桌上有1空盘，允许存放N个水果。爸爸每次向盘中放1个苹果或1个桔子。多个孩子，每个孩子每次只能取2个苹果和1个橘子。 请用wait( )、signal( )原语实现爸爸、孩子并发进程的同步。

答：

semaphore empty=N; //盘中空闲容量

semaphore Apple=0; //盘中苹果数量

semaphore Orange=0; //盘中橘子数量

semaphore sA= N-1; //盘中**尚能放**苹果的最大数量。

semaphore sO= **N-2**; //盘中**尚能放**橘子的最大数量。

semaphore mutex=1; //互斥使用盘子

**爸爸的活动:**

do{

While(1) {

if 拿到一个苹果 {

P(sA);

P(empty);

P(mutex);

将苹果放入盘中；

Signal(mutex);

Signal(Apple);

}

else if 拿到1个橘子 {

P(sO);

P(empty);

P(mutex);

将橘子放入盘中；

Signal(mutex);

Signal(Orange);

}

}while(1)

**孩子的活动:**

do{

**Wait(Apple);**

**Wait(Apple);**

Wait(Orange);

Wait(mutex);

取2个苹果; 取1个橘子；

Signal（mutex）;

Signal（empty）;

Signal（empty）;

Signal（empty）;

**Signal（sA）;**

**Signal（sA）;**

Signal（sO）;

}while(1)

【例4】桌上有1空盘，允许存放N个水果。爸爸每次向盘中放1个苹果或1个桔子。多个孩子。1个孩子**连续取5次**1个苹果和1个橘子后其他孩子才能取。 请用wait( )、signal( )原语实现爸爸、孩子并发进程的同步。

答：

semaphore empty=N; //盘中空闲容量

semaphore Apple=0; //盘中苹果数量

semaphore Orange=0; //盘中橘子数量

semaphore sA=N-1; //盘中**尚能放**苹果的最大数量

semaphore sO=N-1; //盘中**尚能放**橘子的最大数量

semaphore mutex=1; //互斥使用盘子

semaphore mutex1=1; //孩子互斥取水果

**爸爸的活动:**

do{

While(1) {

if 拿到一个苹果 {

P(sA);

P(empty);

P(mutex);

将苹果放入盘中；

Signal(mutex);

Signal(Apple);

}

else if 拿到1个橘子 {

P(sO);

P(empty);

P(mutex);

将橘子放入盘中；

Signal(mutex);

Signal(Orange);

}

}while(1)

**孩子的活动:**

do{

Wait(mutex1);

for (i=1; i<=5; i++) {

Wait(Apple);

Wait(Orange);

Wait(mutex);

取1个苹果; 取1个橘子；

Signal（mutex）;

Signal（empty）;

Signal（empty）;

Signal（sA）;

Signal（sO）; }

Signal（mutex1）;

}while(1)