N 个缓冲区的读写问题

```
int in=0, out = 0;//定义目前存入和取出的指针位置
item B[n];//定义容量为 n 的缓冲池
semaphore S=1;//互斥信号量
semaphore empty=n;//缓冲区的空闲
semaphore full=0;//满缓冲区
void poducer()
{
    while(true)
    {
        poducer an item namev:
        P(empty);//申请一个空缓冲区
        P(S);//申请一个缓冲区使用权
        B[in] = namev;//放入产品
        in = (in+1)% n;//记录下一个缓冲区
        V(S);//释放一个缓冲区使用权
        V(full);//释放一个满缓冲区
    }
}
void consumer()
{
    while(true)
    {
        P(full);//申请一个满缓冲区
        P(S);//申请一个缓冲区使用权
        namep=B[out];//记录缓冲区内产品
        out=(out+1)%n;//记录下一个缓冲区
        V(S);//释放一个缓冲区使用权
        V(empty);//释放一个空缓冲区
        consumer the item in namep;
    }
}
void main(){
    cobegin
        producer();consumer();//并发执行俩个进程
    coend
}
```

家庭吃水果问题

```
semaphore S=1;//互斥信号量
semaphore empty=n;//盘子的空闲
semaphore apples=0;//苹果信号量
semaphore oranges=0;//橘子信号量
void father()
{
    while(true)
    {
        P(empty);
        P(S);
        father puts an fruits;
        V(S);
        if(fruits = orange)
            V(oranges);
            father puts an orange;
        else
            v(apples)
            father puts an apples;
    }
}
void son()
{
    while(true)
    {
        P(apples);
        P(S);
        Take one apple;
        V(S);
        V(empty);
        son eat a apple;
    }
}
void daughter()
{
    while(true)
    {
        P(oranges);
        P(S);
        Take one orange;
        V(S);
        V(empty);
        daughter eat a orange;
        }
}
void main(){
        cobegin
                father();son();daughter();//并发执行进程
        coend
}
```

用管程解决读者写者问题

```
type read-write=monitor
var   readcount : integer;
        write : boolean;
        readqueue , writequeue : condition; //申请读和写的队列
procedure startread()
        begin
                if (write)    wait(readqueue) ;
                readcount++;
                signal(readqueue);
        end
procedure endread()
        begin
                readcount--;
                if (readcount==0) signal(writequeue);
        end
procedure startwrite()
        begin
                if (readcount>0 or write) wait(writequeue);
                write=true;
        end
procedure endwrite()
        begin
                write=false;
                if (readqueue!=NULL) signal(readqueue);
                else signal(writequeue);
        end
begin
        readcount=0;
        write=false;
end
reader:
        begin
                repeat
```

```
            read-write.startread;
            读者读书中
            read-write.endread;
        until false;
    end
writer:
    bigin
        repeat
            read-write.startwrite;
            写者写书中
            read-write.endwrite;
        until false
    end
```