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

}