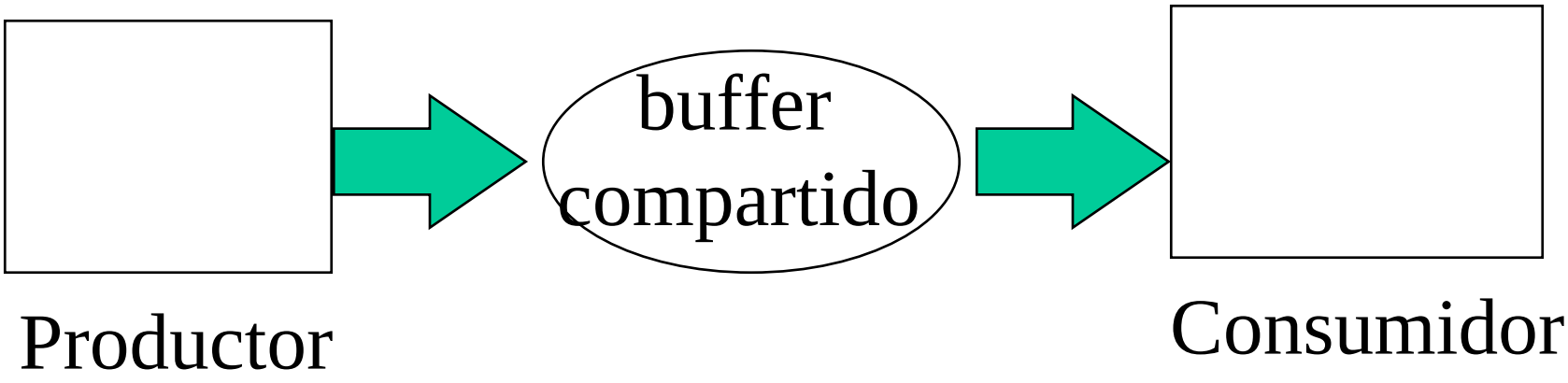
Problemas de sincronización

Los procesos comparten recursos y necesitan , previamente, coordinar sus acciones.



```
#include “prototypes.h”
#define N 100
int count = 0;
```

```
void producer(void)
{
  int item;
  while TRUE {
    producer_item(&item);
    if (count == N) sleep();
    enter_item(item);
    count = count + 1;
    if (count ==1)
      wakeup(consumer);
  }
}
```

```
void consumer(void)
{
  int item;
  while TRUE {
    if (count == 0) sleep();
    remove_item(&item);
    count = count - 1;
    if (count ==N-1)
      wakeup(producer);
   consume_item(item);
  }
}
```

Una solución para el Productor-Consumidor con semáforos

```
semaphore mutex=1;  /*controla el acceso a la SC*/
semaphore empty=N;  /*controla los slots vacios*/
semaphore full=0;  /*controla los slots llenos*/
```

```
Proceso Productor();
while true do{
  produce_item;
  down(empty);
  down(mutex);
  put_item;
  up(mutex);
  up(full);
 }

EndProcess;
```

```
Proceso Consumidor();
while true do{
  down(full);
  down(mutex);
  remove_item;
  up(mutex);
  up(empty);
 }

EndProcess;
```

Una solución incorrecta para el Productor-Consumidor con semáforos

```
semaphore mutex=1;  /*controla el acceso a la SC*/
semaphore empty=N;  /*controla los slots vacios*/
semaphore full=0;  /*controla los slots llenos*/
```

```
Proceso Productor();
while true do{
  produce_item;
  down(empty);
  down(mutex);
  put_item;
  up(mutex);
  up(full);
 }

EndProcess;
```

```
Proceso Consumidor();
while true do{
  down(mutex);
  down(full);
  remove_item;
  up(mutex);
  up(empty);
 }

EndProcess;
```

Si el consumidor corre primero con un buffer vacío:va a dormir sobre down(full) y, depués, el productor va a dormir con down(mutex)

```
semaphore mutex=1;  /*controla el acceso a la SC*/
semaphore empty=N;  /*controla los slots vacios*/
semaphore full=0;  /*controla los slots llenos*/
```

```
Proceso Productor();
while true do{
  produce_item;
  down(empty);
  down(mutex);
  put_item;
  up(mutex);
  up(full);
 }

EndProcess;
```

```
Proceso Consumidor();
while true do{
  down(mutex);
  down(full);
  remove_item;
  up(mutex);
  up(empty);
 }

EndProcess;
```

Una solución incorrecta para el problema del Productor-Consumidor con semáforos

Ha ocurrido un *candado mortal*. Tanto el productor como el consumidor son puestos a dormir y nunca despertarán.