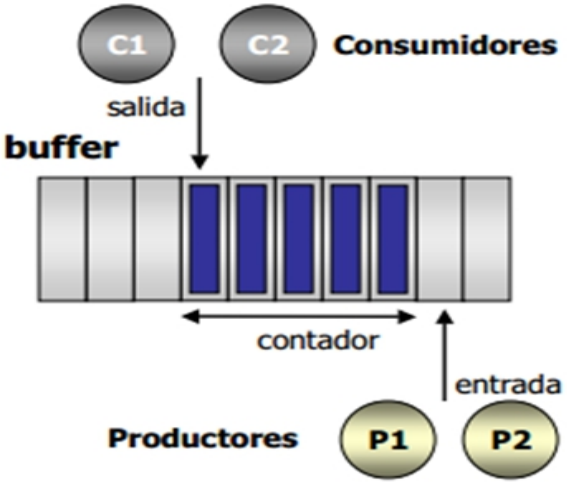
Departamento de Programación
Facultad de Informática
Universidad Nacional del Comahue

Programación Concurrente

Sincronización – Monitores – Semáforos – Locks
Problema del productor/consumidor
Problema de lectores/escritores
Problema de los filósofos
Problema del barbero dormilón

Problema del Productor/Consumidor



Semáforo binario

Semáfor general

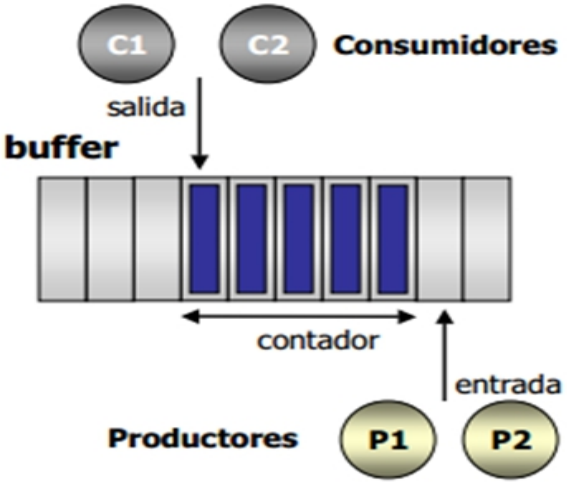
Monitor

Lock + variable de condición

Casos:
- 1 productor / 1 consumidor
- 1 productor / varios consumidores
- varios productores / 1 consumidor
- varios productores / varios consumidores

Buffer ilimitado
Buffer tamaño n

Problema del Productor/Consumidor



Semáforo binario

Semáfor general

Monitor

Lock + variable de condición

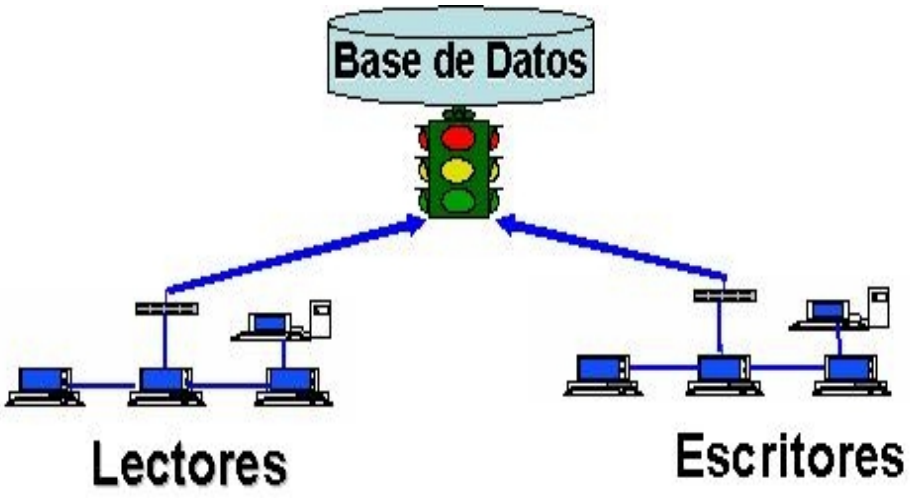
Recuperación selectiva
Recuperación no selectiva

Casos:
- 1 productor / 1 consumidor
- 1 productor / varios consumidores
- varios productores / 1 consumidor
- varios productores / varios consumidores

Buffer ilimitado
Buffer tamaño n

Problema de Lectores/escritores

- Permite el acceso simultáneo de lectores impidiendo el uso a escritores?
- Es posible la inanición de algún tipo de proceso?



Semáforo binario

Semáfor general

Monitor

Lock + variable de condición

Problema de sincronización:

Barbero dormilón

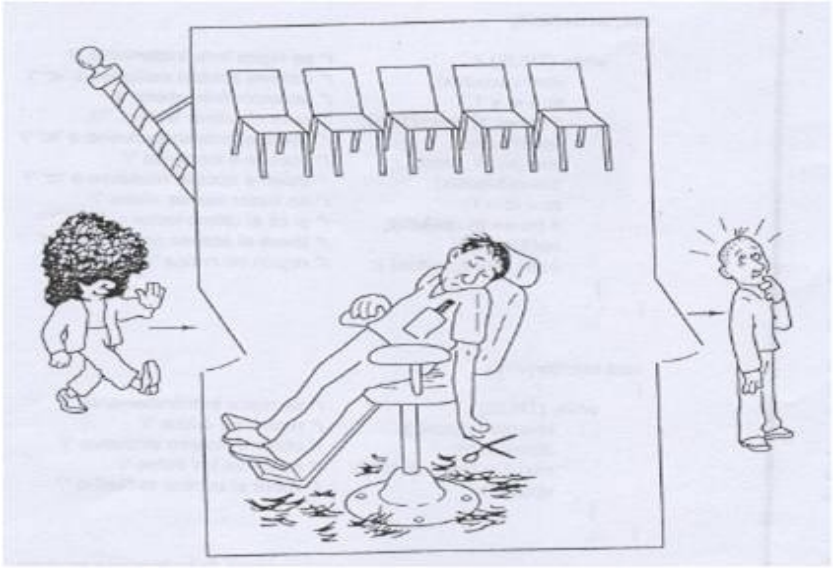
En una barbería trabaja un barbero que tiene un único sillón de barbero y varias sillas para esperar.

- Cuando no hay clientes, el barbero se sienta en una silla y se duerme.

-Cuando llega un nuevo cliente,

-si el barbero duerme: despierta al barbero o

-si el barbero está afeitando a otro cliente: se sienta en una silla



(si todas las sillas están ocupadas por clientes esperando, se va).

Problema del Barbero dormilón

Se pueden considerar 2 casos:

- se respeta el orden de llegada de los clientes

Se requiere el uso de una estructura para considerar las sillas de la sala de espera

- no se respeta el orden de llegada

Se puede trabajar con la cantidad de sillas o cantidad de clientes

- ¿se podra utilizar un semáforo general?

Objetos activos: barbero y clientes

Objetos pasivos: recursos compartidos (sillas de espera y sillón del barbero)

Problema del Barbero dormilón

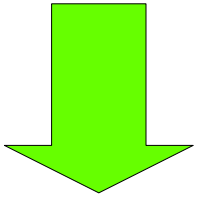
Sincronización

- existe una interacción directa entre el cliente y el barbero. El cliente que pasa al sillón debe despertar al barbero y el barbero debe notificar al cliente cuando ha terminado de cortarle el pelo.

Problema del Barbero dormilón

Sincronización

- existe una interacción directa entre el cliente y el barbero. El cliente que pasa al sillón debe despertar al barbero y el barbero debe notificar al cliente cuando ha terminado de cortarle el pelo.



Rendezvous

Problema del Barbero dormilón

Accionar del barbero

- permanece dormido a la espera de 1 cliente nuevo (es decir bloqueado sobre un semáforo o un monitor)
- corta el pelo a un cliente
- notifica al cliente para que se retire

Accionar de un cliente

- llega a la barberia
  - sillas ocupadas, se va
  - sillas libres, espera su turno
- se sienta en el sillón del barbero, libera su silla y despierta al barbero,
- espera mientras se le realiza el corte
- abandona la barberia

Monitores
Semáforos
Locks

Problema de sincronización:

Cena de filósofos

Cinco filósofos se sientan alrededor de una mesa y pasan su vida cenando y pensando. Cada filósofo tiene un plato de fideos y un tenedor a la izquierda de su plato. Para comer los fideos son necesarios dos tenedores y cada filósofo sólo puede tomar los que están a su izquierda y derecha.

Este problema es uno de los **problemas clásicos de sincronización** entre procesos, donde es necesario gestionar el acceso concurrente a los recursos compartidos (los tenedores) . Un tenedor puede considerarse como un recurso indivisible, es decir, en todo momento se encontrará en un estado de uso o en un estado de no uso.

Problema de sincronización:

Cena de filósofos

Si cualquier filósofo toma un tenedor y el otro está ocupado, se quedará esperando, con el tenedor en la mano, hasta que pueda tomar el otro tenedor, para luego empezar a comer.

Si dos filósofos adyacentes intentan tomar el mismo tenedor a una vez, ambos compiten por tomar el mismo tenedor, y uno de ellos se queda sin comer.

Si todos los filósofos toman el tenedor que está a su derecha al mismo tiempo, entonces todos se quedarán esperando eternamente. Entonces los filósofos se morirán de hambre

DEADLOCK

Problema de sincronización:

Cena de filósofos

Opciones para evitar el interbloqueo:

- limitar el **número de filósofos a cuatro**.

- permitir que un filósofo tome sus tenedores si y sólo si **ambos (izq y der) están disponibles**

- establecer un orden en la obtencion de los tenedores, por ejemplo que los filosofos en posiciones pares tomen primero el tenedor de la izquierda y los impares el de la derecha.