



Departamento de Programación
Facultad de Informática
Universidad Nacional del Comahue



Programación Concurrente



Sincronización: Problemas clásicos



Problemas Clásicos

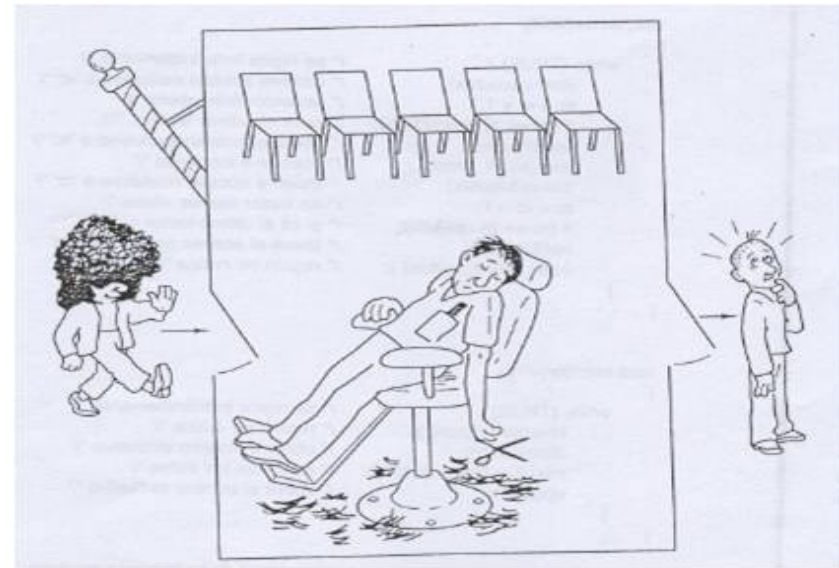
- 1.- Taxista
- 2.- Barbero Dormilón
- 3.- Productor / Consumidor
- 4.- Lectores / Escritores
- 5.- Filósofos cenando

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

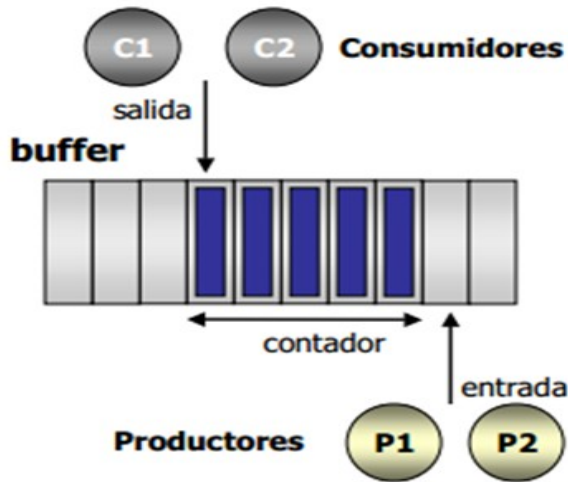


(si todas las sillas están ocupadas por clientes esperando, se va).

Problema de sincronización:

Productor/Consumidor

Una parte produce algún producto (datos en nuestro caso) que se coloca en algún lugar (una cola, por ejemplo) para que sea consumido por otra parte.



- El productor genera sus datos en cualquier momento
- El consumidor puede tomar un dato sólo cuando hay
- Para el intercambio de datos se usa una cola a la cual ambos tienen acceso, así se garantiza el orden correcto (Esto puede variar)
- Todo lo que se produce debe ser consumido

Tenemos que garantizar que el consumidor no consuma más rápido de lo que produce el productor

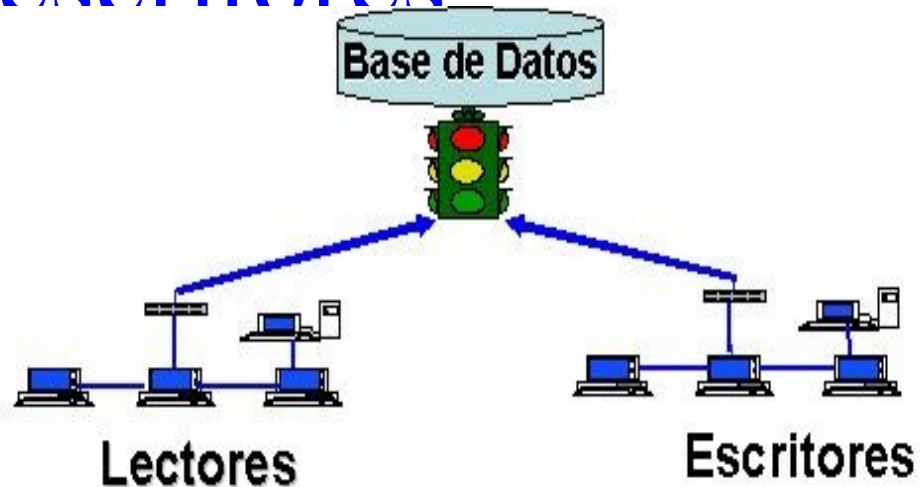
Acceso de los productores y consumidores de forma de asegurar consistencia de la información almacenada en la cola.

La cola puede tener capacidad limitada: a) un productor no puede producir un elemento en una cola llena, b) un consumidor no puede extraer un elemento de una cola vacía

Problema de sincronización:

Lectores/escritores

Dos grupos de procesos (Lectores/escritores) que utilizan los mismos recursos (documento)

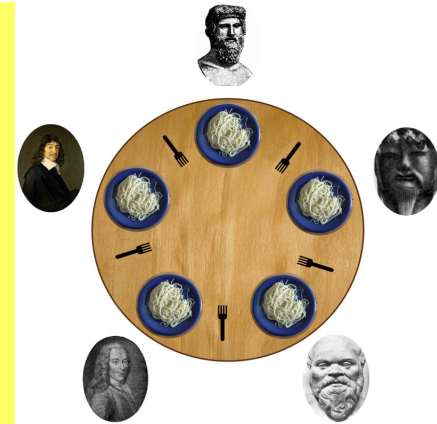


- Un grupo de lectores, sólo leen los datos.
- Los escritores, leen y escriben.
- Varios lectores pueden acceder simultáneamente a un proceso compartido
- Se debe evitar que accedan simultáneamente un proceso escritor y cualquier otro proceso.

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.



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 (deadlock), entonces los filósofos se morirán de hambre



Como abordar este tipo de problemas

- Identificar los hilos
 - identificar las entidades que forman parte del problema
 - identificar las clases correspondientes
- Identificar los recursos compartidos.
 - es necesario asociar herramientas de sincronización para garantizar la seguridad.
- Identificar los eventos de sincronización
 - Para delimitar el orden en que deben ocurrir los eventos