

## 1. **Buffer oscilante.**

En algunos sistemas de buffering las llamadas a las operaciones de insertar y extraer se realizan por rachas. En estas situaciones es deseable la posibilidad de insertar y extraer datos sobre el mismo buffer de forma simultánea.

En general, no se puede asumir que la ejecución simultánea (sin asegurar exclusión mutua) de dos operaciones cualesquiera sobre un recurso del tipo cola deje al recurso en un estado consistente, por lo tanto los accesos sobre dicho recurso deben ser excluyentes.

¿Qué ocurriría si el buffer estuviera formado por dos colas independientes? Pareciera que existe la posibilidad de insertar en una y extraer de la otra de forma simultánea, sólo es necesario establecer un protocolo que asegure que el buffer, en su conjunto, respeta una política FIFO. Dicho protocolo consiste en decidir (oscilar) entre una cola u otra en cada momento. Veamos un escenario de ejecución:

- a. Supongamos las dos colas inicialmente vacías, una de ellas etiquetada para insertar y otra para extraer.

**(insertar) 1 | (extraer) 2**

- b. Si un proceso quiere extraer datos no puede hacerlo porque no hay ningún dato (y más precisamente porque no hay datos en la cola de extraer).
- c. Si un proceso quiere insertar datos tendría que hacerlo en la cola etiquetada para ello (la cola 1).

Supongamos la inserción de un dato d1. Además de insertar d1 en la cola 1 el proceso debería “oscilar” las colas, es decir, cambiar las etiquetas de insertar y extraer para posibilitar la simultaneidad entre una extracción y una inserción.

**(extraer) 1 d1 | (insertar) 2**

- d. Supongamos que llegan dos procesos que quieren insertar datos (d2 y d3), tendrán que insertar en la cola etiquetada para ello y lo tendrán que hacer en exclusión mutua:

**(extraer) 1 d1 | (insertar) 2 d2 d3**

- e. Si ahora un proceso quiere insertar y otro extraer pueden realizar la operación de forma simultánea y, finalmente . . . , hay que volver a “oscilar”:

**(insertar) 1 | (extraer) 2 d2 d3 d4**

Como puede observarse la situación permite que de nuevo puedan ejecutarse de forma simultánea una operación de inserción y una de extracción.

Como puede observarse en los ejemplos, la oscilación debe producirse cada vez que la cola de extracción queda vacía.

Se debe implementar una sistema de buffering como el indicado, considerando *hilos insertores* e *hilos extractores*, y el recurso compartido. Considere ***tamaño ilimitado***

## **2. Buque de autos**

Un transbordador o buque para transporte de vehículos permite pasar autos de un lado de un río al otro. Los autos viajan por el lado este del río, cruzan el río en el transbordador y continúan su viaje por el lado oeste (nunca vuelven). El transbordador tiene espacio para 10 autos y espera a estar lleno para cruzar el río. Cuando ha cruzado y descargado los coches, vuelve vacío. Considere que los autos deben bajar de a uno por vez.

Considere en el transbordador operaciones ir y volver

- ir hace que el transbordador cruce con los autos
- volver lo hace volver vacío.

Se trata de implementar este problema con hilos, resolviendo la concurrencia con el empleo de algún mecanismo de sincronización (monitor, semáforos, locks).

**Nota 1: los autos NO DEBEN caerse al río**

**Nota2: Resolver el problema teniendo en cuenta que, en el caso en que se quiera desbloquear a varios hilos debe hacerse en cadena, esto es, cada proceso debe desbloquear al siguiente.**