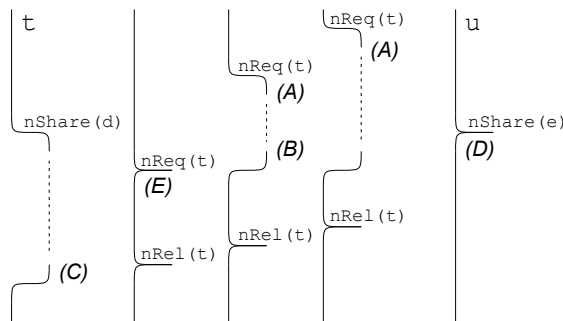


Pregunta 1

Se le pide a Ud. agregar una nueva primitiva de sincronización para compartir datos en nSystem. La API de este mecanismo es la siguiente:

<i>Procedimiento</i>	<i>Significado</i>
<code>void nShare(char* data);</code>	Ofrece datos para compartir
<code>char *nRequest(nTask t);</code>	Solicita datos provenientes de la tarea t
<code>void nRelease(nTask t);</code>	Notifica que los datos ya no se usarán

La figura muestra que nRequest debe esperar que t invoque nShare (ver A). Cuando t invoca nShare, se desbloquean todas las tareas que esperaban un nShare(data) de t (ver B), retornando todas ellas el puntero data. nShare se bloquea hasta que todas las tareas con un nRequest pendiente de t notifiquen mediante nRelease que data de t no se usará más (ver C). Si t invoca nShare y no encuentra tareas solicitando datos de t, nShare retorna de inmediato (ver D). Si se invoca nRequest(t) mientras nShare está activo en t, nRequest retorna de inmediato los datos que actualmente comparte t (ver E).



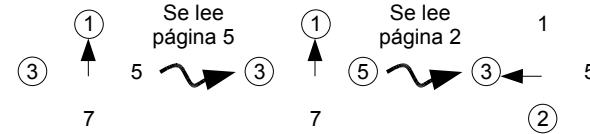
Implemente esta API usando los procedimientos de bajo nivel de nSystem (START_CRITICAL, Resume, PutTask, etc.). Ud. *no puede usar* otros mecanismos de sincronización ya disponibles en nSystem, como semáforos, monitores, mensajes, etc.

Pregunta 2

Parte a.- Implemente un buffer de tamaño uno para el problema de múltiples productores y múltiples consumidores usando 2 spin-locks. Concretamente Ud. debe implementar las siguientes rutinas en no más de 15 líneas, en forma correcta, pero no necesariamente en forma eficiente:

<i>Procedimiento</i>	<i>Significado</i>
<code>void put(Item item);</code>	Deposita un ítem en el buffer
<code>Item get();</code>	Extrae un ítem del buffer

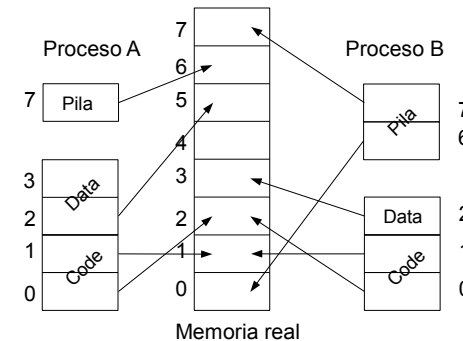
Parte b.- La siguiente figura muestra los estados sucesivos de la memoria al realizar 2 accesos en un sistema de memoria virtual que usa la estrategia del reloj para el reemplazo de páginas.



Las páginas que tienen el bit de referencia en 1 aparecen encerradas en una circunferencia. La flecha indica la posición del puntero de la estrategia del reloj. Siguiendo el mismo esquema de la figura, muestre todos los estados por los que pasa la memoria al realizar los siguientes accesos:

1 4 2 1 7 5 2 4 3

Parte c.- Un sistema Unix que implementa demand paging con páginas de 4 KB, ejecuta los procesos A y B. El estado actual de la memoria es el siguiente:



Haga la tabla de páginas para el proceso A indicando en la tabla: número de página virtual, número de página real y atributos de validez y escritura (V y W).

Parte d.- Suponga que el proceso A de la parte c.- invoca fork y el sistema operativo implementa la estrategia copy-on-write. En seguida el scheduler de procesos retoma el proceso hijo el cual modifica sus páginas virtuales 7 y 3. Para este instante modifique el diagrama de la parte c.- de modo que muestre una asignación válida de la memoria. Explique sus supuestos. (No necesita confeccionar tablas de páginas como la que se pidió en la parte c.)