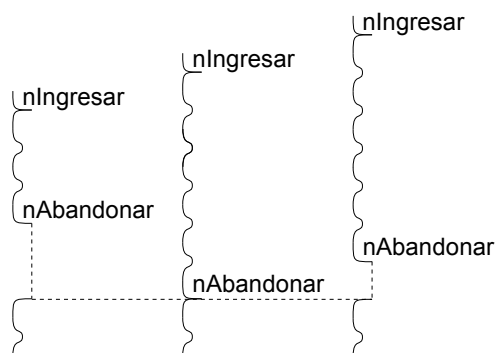


Pregunta 1

Se requiere implementar en nSystem un sistema de reuniones. Las funciones son las siguientes:

- *void nIngresar()*: notifica el ingreso a la reunión.
- *void nAbandonar()*: solicita finalizar la reunión. Se bloquea en espera hasta que todas las tareas que hayan ingresado hayan solicitado finalizar la reunión. Esto incluye aquellas tareas que ingresen después de esta invocación de *nAbandonar*.

La siguiente figura muestra con un diagrama de threads un ejemplo de uso.



Observe que cuando se invoca el último *nAbandonar* todas las tareas pasan a estado READY.

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. Ud. podría necesitar agregar nuevos campos al descriptor de tarea de nSystem, nuevos estados, nuevas variables globales, etc.

Pregunta 2

Resuelva el mismo problema de la pregunta 1 considerando ahora una máquina *octa-core* en la que no existe un núcleo de sistema operativo y por lo tanto la única herramienta de sincronización preexistente es el *spin-lock*. Las funciones que Ud. debe implementar son las siguientes:

```
void ingresar();
void abandonar();
```

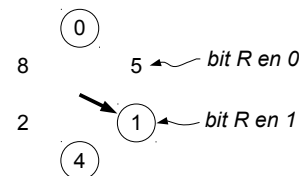
Para programar su solución Ud. dispone de *spin-locks* y la función

coreId() que entrega el número del core que la invoca. Necesitará declarar variables globales.

Restricción: Dado que no hay un núcleo de sistema operativo, la única forma válida de esperar el fin de la reunión es utilizando un *spin-lock*. Otras formas de *busy-waiting* no están permitidas. No hay *malloc* ni *fifoqueues*.

Pregunta 3

I. Considere un sistema Unix que implementa la estrategia del reloj. El sistema posee 6 páginas reales disponibles y corre un solo proceso. La figura de la derecha indica el estado inicial de la memoria, mostrando las páginas residentes en memoria, la posición del cursor y el valor del bit R:



Dibuje los estados por los que pasa la memoria para la siguiente traza de accesos a páginas virtuales:

8 6 0 1 7 8

II. Se acaban de leer los bloques 200 y 300 en ese orden en un disco duro. Ahora 6 procesos se encuentran en espera con peticiones para leer los bloques 100, 800, 200, 700, 900 y 400. En qué orden se harán las lecturas cuando la estrategia es: (a) *shortest seek first*, (b) método del ascensor o *look*.

III. Suponga que Ud. necesita incrementar una variable global al programar código de un núcleo de Unix para una máquina multicore. Explique si será necesario asegurar la exclusión mutua y cómo lo haría de la manera más eficiente. Responda una vez considerando un núcleo clásico y una segunda vez considerando un núcleo moderno.

IV. Un computador posee un disco duro con un tiempo de acceso de 10 milisegundos y una velocidad de transferencia de 100 MB/seg. Estime cuánto tiempo tomaría mostrar todas las líneas que contengan el string “j Perez” en (a) un archivo de texto desordenado de 10 MB, y (b) 1000 archivos de texto desordenado de 10 KB cada uno, en el mismo directorio pero dispersos en el disco.