

Nota: Descargue de U-cursos *ctl.zip* y descomprímalo. Hay programas de prueba en los directorios P1, P2 y P3. Ejecute el comando `make` para recibir instrucciones acerca del archivo en donde debe programar su solución, cómo compilar, ejecutar y depurar. No necesita definir la variable `NSYSTEM`. Vienen incluidos `pSystem` y `nSystem`. No necesita que sus soluciones pasen los tests para obtener una buena nota, pero por lo menos compílela.

Pregunta 1

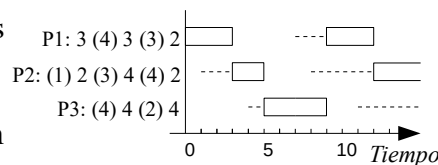
A) Considere la siguiente solución del problema del productor consumidor para el caso de un buffer de tamaño 1, con un solo productor y un solo consumidor.

<pre>int buf; // El buffer de tamaño 1 nSem prod; // = nMakeSem(0); nSem cons; // = nMakeSem(0);</pre>	
<pre>void put(int item) { buf= item; nSignalSem(prod); nWaitSem(cons); }</pre>	<pre>int get() { nWaitSem(prod); nSignalSem(cons); return buf; }</pre>

Haga un diagrama de threads que muestre que si el productor deposita sucesivamente los ítemes 1 y 2, podría ocurrir que el consumidor reciba incorrectamente 2 y 2. ¡A menudo sí funciona correctamente!

B) Corrija la solución de la parte **A)** reescribiendo solamente la función `get`. No puede modificar `put` o la declaración de las variables globales.

C) El diagrama parcial muestra las decisiones de scheduling para 3 procesos. Para cada proceso se indica la duración de la ráfaga y la duración de su estado de espera entre paréntesis. En el diagrama la línea punteada indica que el estado del proceso es `READY`. Si el proceso está en estado de espera el espacio aparece en blanco. ¿De qué estrategia de scheduling se trata? Rehaga el diagrama completo considerando ahora la estrategia *shortest job first* non preemptive. Considere que el predictor de duración para la próxima ráfaga es la duración de la última ráfaga.



Pregunta 2

En una calle hay 10 estacionamientos contiguos al lado derecho (y ninguno al lado izquierdo). Se identifican como 0, 1, 2, 3, ..., 9. Las

tareas de `nSystem` representan a los vehículos que solicitan estacionar llamando a la función `reservar`. Esta función recibe como parámetro el número de estacionamientos contiguos requeridos (k). Los vehículos son de distinto tamaño de modo que un *citycar* requiere solo un estacionamiento pero un camión podría requerir 6 estacionamientos contiguos. Si hay k estacionamientos contiguos disponibles, esta función los reserva y retorna de inmediato la identificación del primer estacionamiento en la serie otorgada (0, 1, 2, etc.). De lo contrario, `reservar` espera hasta que hayan k estacionamientos contiguos disponibles. Cuando un automovilista se va invoca la función `liberar` indicando el primer estacionamiento que ocupaba (e) y cuantos se le habían otorgado (k). Esta función libera todos los estacionamientos de la serie otorgada previamente a ese automovilista con `reservar`, y por lo tanto pueden ser otorgados a otros automovilistas.

Programa las funciones `reservar` y `liberar`. Sus encabezados son:

```
void initReservar(); // inicialización
int reservar(int k);
void liberar(int e, k);
```

Restricciones: Debe atender a los vehículos por orden de llegada y por lo tanto necesitará una *fifoqueue*. Debe evitar cambios de contexto innecesarios. Para la sincronización use un solo monitor de `nSystem` con múltiples condiciones.

Ayuda: El archivo `P2/reservar.c.plantilla` incluye las funciones `buscar` y `marcar` que le economizarán tiempo al resolver este problema.

Pregunta 3

Resuelva el mismo problema de la pregunta 2, pero como herramienta de sincronización nativa de `nSystem`, es decir recurriendo a las funciones de bajo nivel de `nSystem` (`START_CRITICAL`, `ResumeNextReadyTask`, `PushTask`, etc.). Ud. no puede usar otros mecanismos de sincronización ya disponibles en `nSystem` como semáforos, monitores, mensajes, etc. Puede usar las colas de threads de `nSystem` (tipo *Queue*, el mismo de la *ready_queue*). Los encabezados de las funciones que debe implementar son:

```
void InitReservar(); // inicialización
int nReservar(int k);
void nLiberar(int e, k);
```

Ayuda: Resuelva este problema en el archivo `npsystem/nsrc/nReservar.c`. La plantilla es `nReservar.c.plantilla`, en el mismo directorio.