



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

### **Práctica 4**

EQUIPO

**Arroyo Martínez Erick Daniel  
Terrazas Rivera Alex  
Vergara Navarro Mixtli Arturo**

PROFESOR

**Gilde Valeria Rodríguez Jiménez**

AYUDANTES

**Rogelio Alcantar Arenas  
Gibran Aguilar Zuñiga  
Luis Angel Leyva Castillo  
Rogelio Alcantar Arenas**

ASIGNATURA

**Computación Concurrente**

24 de marzo de 2024

# Cuestionario

Contesta las siguientes preguntas (Para cada proyecto):

## Cumpleaños

- ¿Tu solución cumple con Exclusión mutua? Argumenta porqué.  
Nuestra primer impresión es que nuestra implementación cumple esta propiedad, pues usamos el algoritmo del filtro para garantizar que solo un inversionista pueda tomar ambos tenedores simultáneamente. El uso del semáforo en la implementación de los tenedores también asegura la exclusión mutua al acceder a los tenedores. Qué una forma de garantizar, era usar un PetersonLock, pues solo dos inversionistas pueden tratar de acceder a un tenedor, y como sabemos estos locks garantizan la exclusión mutua. Sin embargo, es probable que estados particulares del hardware generen condiciones de carrera.
- ¿Tu solución cumple con Deadlock-free? Argumenta porqué.  
Análogo a lo anterior, si podemos garantizar, de algún modo, exclusión mutua. Entonces se podría implicar esta propiedad. Por lo tanto, en general podemos decir que nuestra implementación cumple con esta propiedad pues como se mencionó controlamos el acceso a la sección critica (tenedores).
- ¿Tu solución cumple con Libre de Hambruna (Starvation-free)? Argumenta porqué.  
Nuestra implementación no garantiza la libertad de hambruna. Pues si un inversionista no puede adquirir los tenedores debido a la competencia con otros inversionistas, podría quedar bloqueado indefinidamente. Por tanto, no hay garantía de que todos los inversionistas tengan la misma oportunidad de acceder a los tenedores, lo que implica que nuestra implementación es susceptible a hambruna.
- ¿Tu solución cumple con Justicia? Argumenta porqué. En caso de que no lo cumpla, cómo podrías garantizarla.  
Nuestra implementación no garantiza la justicia. Pues un inversionista podría quedar atrapado esperando indefinidamente para adquirir los tenedores si otros inversionistas siempre los obtienen primero, lo que implicaría que el orden de inicialización y termino no es consistente. Por tanto, sino existe justicia en todo estado de la ejecución, entonces no se cumple en general. Para garantizar la justicia, podríamos implementar un esquema de asignación de recursos que asegure que todos los inversionistas tengan una oportunidad justa de acceder a los tenedores en orden de inicialización. Esto lo podríamos lograrse mediante una cola de espera donde los inversionistas esperan su turno para adquirir los tenedores.

## Estacionamiento

- ¿Tu solución cumple con Exclusión mutua? Argumenta porqué.  
Si cumple con la exclusión mutua. Puesto que hemos utilizado semáforos para controlar el acceso a los recursos compartidos (los lugares de estacionamiento). Al adquirir el semáforo antes de acceder al lugar de estacionamiento y liberarlo después de salir, garantizas que solo un hilo (carro) pueda ocupar un lugar de estacionamiento a la vez. Además, controlamos el acceso de hilos al estacionamiento (primer sección crítica) también haciendo uso, aumentando más el nivel de sincronización entre los hilos ejecutados.
- ¿Tu solución cumple con Deadlock-free? Argumenta porqué.  
Aparentemente, nuestra implementación cumple con deadlock-free. Dado que que no hay una situación donde los hilos estén esperando indefinidamente a que se libere un recurso que está siendo retenido por otro hilo. Los semáforos que utilizas están correctamente adquiridos y liberados en una secuencia ordenada, lo que evita la posibilidad de deadlocks. Si agregamos que el comportamiento de cada proceso dentro de cada lugar y del estacionamiento es simular un retardo únicamente. Por lo que, los recursos de la sección crítica son independientes entres si.
- ¿Tu solución cumple con Libre de Hambruna (Starvation-free)? Argumenta porqué.  
Podemos inferir que nuestra implementación no garantiza la libertad de hambruna. Esto se debe a que los hilos no tienen prioridad explícita y pueden quedarse esperando indefinidamente para entrar al estacionamiento si el semáforo no se libera. Sin embargo, dado que limitamos la cantidad de carros que pueden entrar simultáneamente al estacionamiento mediante el semáforo, es poco probable que ocurra hambruna en condiciones normales de ejecución. Para garantizar la libertad de hambruna, podríamos implementar un esquema que implique la siguiente propiedad (justicia), que asegure que todos los hilos tengan la oportunidad de entrar al estacionamiento después de un período razonable de espera.

- ¿Tu solución cumple con Justicia? Argumenta porqué. En caso de que no lo cumpla, cómo podrías garantizarla. Nuestra implementación, no garantiza la justicia en el sentido de que algunos hilos podrían tener que esperar mucho tiempo para entrar al estacionamiento si hay una alta demanda, además de que, podría ocurrir que algunos hilos adquieran de forma ordenada el semaforo, generando un nivel de injusticia parcial, que implicaría que toda la implementación no la cumple. Para garantizar la justicia, podríamos implementar una política de cola donde los hilos se asignan a los lugares en el orden en que llegan. Esto podría lograrse utilizando una cola para asignar los lugares disponibles. Así podríamos garantizar que los hilos (carros) que llegan primero tengan prioridad para entrar al estacionamiento, lo que implicaría el cumplimiento de la propiedad de justicia.

## Referencias

- Gafni, E. (2018) Group mutual exclusion in linear time and space. Recuperado de: <https://www.sciencedirect.com/science/article/pii/S0304397517304747>