

PROGRAMACIÓN PARALELA – PRÁCTICA TUNELES

GRUPO 25

ALEJANDRO CRUZ VELASCO

PABLO MOLLÁ CHÁRLEZ

DIEGO RODRÍGUEZ GARCÍA

PABLO SÁNCHEZ LÓPEZ-PALOMINO

VARIABLES

En primer lugar, mostramos y explicamos nuestras variables:

- NorthCars (int): número total de coches que irán al norte, la tomamos de forma aleatoria entre 4 y 10.
- SouthCars (int): número total de coches que irán al sur, la tomamos de forma aleatoria entre 4 y 10.
- nnorth (int): número de coches que van al norte actualmente.
- nsouth (int): número de coches que van al sur actualmente.
- nnorth_waiting (int): número de coches esperando para ir al norte.
- nsouth_waiting (int): número de coches esperando para ir al sur.
- turn (int): indica el turno (0 significa que hay algún coche hacia el sur, mientras que 1 significa que hay algún coche hacia el norte).
- mutex: Lock
- nonorth: Condition (self.mutex). Sirve para ver que no hay coches hacia el norte.
- nosouth: Condition (self.mutex). Sirve para ver que no hay coches hacia el sur.

INVARIANTE

Definimos el invariante de tal forma:

$I = \{ \text{nnorth} = 0 \text{ or } \text{nsouth} = 0 \}.$

Este indica que no hay simultáneamente coches que van hacia el norte y hacia el sur, lo necesario para poder circular correctamente. A continuación, explicaremos los procesos del código para comprobar que el invariante siempre se cumple. Lo haremos solo en la situación de circulación hacia el sur, ya que el otro caso es completamente análogo.

- **empty_south():** devuelve `south.value == 0 and (turn.value == 0 or nsouth_waiting.value==0).`

Comprueba que no hay coches hacia el sur y que o bien es el turno 0, o bien es el turno 1 pero no hay coches esperando. No cambia ninguna variable, por lo que se mantiene el invariante.

- **wants_go_south():**
mutex.acquire()
nsouth_waiting.value += 1 (no afecta a las variables del invariante)
no_north.wait_for(empty_north) (esperamos a que *nnorth == 0* y no hay coches esperando hacia el norte)
nsouth_waiting.value += 1
nsouth.value += 1 (hemos esperado a que *nnorth == 0*, por lo que el invariante sigue siendo cierto)
mutex.release()

- **go_south():**
mutex.acquire()
nsouth.value -= 1 (reduce *nsouth*, por lo que se sigue cumpliendo el invariante)
turn.value = 0 (cambiamos el turno)
if nosouth.value == 0:
 nosouth.notify_all() (notificamos para los procesos del norte)
mutex.release()