PRÁCTICA 2 MONITORES

LA ANCHURA DEL PUENTE NO ADMITE VEHÍCULOS EN AMBOS SENTIDOS

LOS PEATONES Y LOS COCHES NO PUEDEN COMPARTIR EL PUENTE

SÍ PUEDEN PASAR PEATONES EN AMBOS SENTIDOS

coche (cid : int, direction : int , monitor : Monitor)

peatón (pid : int , monitor : Monitor)

SUR = 0 , NORTE = 1

TIEMPO_COCHE_NORTE = 0.5
TIEMPO_COCHE_SUR = 0.5
TIEMPO_PEATON = 5

} tiempo que tarda en querer entrar un nuevo coche / peatón en el puente

TIEMPO_COCHE_PUENTE = (1, 0.5)

TIEMPO_PEATON_PUENTE = (30, 10)

```
class Monitor():
    def __init__(self):
        N=4
```

¿Qué son los monitores?

Sus métodos son ejecutados con exclusión mutua ↗ En cada momento en el tiempo un hilo como máximo puede estar ejecutando cualquiera de sus métodos

Estructuras de datos abstractas

UN MONITOR TIENE CUATRO COMPONENTES

Inicialización. contiene el código a ser ejecutado cuando el monitor es creado

Datos privados: contiene los procedimientos privados, que solo pueden ser usados desde dentro del monitor y no son visibles desde fuera

Métodos del monitor: son los procedimientos que pueden ser llamados desde fuera del monitor

Cola de entrada: contiene a los hilos que han llamado a algún método del monitor pero no han podido adquirir permiso para ejecutarlos.

```
def __init__(self):                    self = Monitor
    self.mutex = Lock()
    self.patata = Value('i', 0)
```

PUENTE DE AMBITE

Monitor

cars_north_waiting. int = 0
cars_south_waiting. int = 0
ped_waiting. int = 0
} colas

ncar_north. int = 0
ncar_south. int = 0
nped. int = 0
} en el puente

INV ≡ { ncar_north ≥ 0, ncar_south ≥ 0, nped ≥ 0, ncar_north > 0 → ncar_south = 0 and nped = 0, ncar_south > 0 → ncar_south = 0 and nped = 0, nped > 0 → ncar_south = 0 and ncar_north = 0 }

want_enter_north_car
{INV}
cars_north_waiting = cars_north_waiting + 1
no_south_car. wait (ncar_south == 0)
no_ped. wait (nped == 0)
ncar_north = ncar_north + 1
{INV}

want_enter_ped
{INV}
ped_waiting = ped_waiting + 1
no_north_car. wait (ncar_north == 0)
no_south_car. wait (ncar_south == 0)
nped = nped + 1
{INV}

want_enter_south_car
{INV}
cars_south_waiting = cars_south_waiting + 1
no_north_car. wait (ncar_north == 0)
no_ped. wait (nped == 0)
ncar_north = ncar_north + 1
{INV}

leaves_north_car
{INV ∧ ncar_north > 0}
ncar_north = ncar_north - 1
no_north_car. signal

leaves_north_car
{INV ∧ ncar_south > 0}
ncar_south = ncar_south - 1
no_south_car. signal

leaves_ped

{ INV ∧ nped > 0 }

nped = nped − 1

no_ped.signal ( )

EL PUENTE ES SEGURO

La condición principal que se debe cumplir para que pueda entrar un elemento de una de las 3 categorías (coches norte, coches sur y peatones) es que no haya en el puente elementos de las otras dos.

Para entrar se espera hasta que esta condición sea cierta

(puente_01.py)

AUSENCIA DE INANICIÓN

Para evitar inanición incorporamos un funcionamiento de turnos

Turno 0: Coches norte → Cuando sale el coche del norte si hay coche del sur esperando, el turno pasa a 1

Turno 1: Coches sur → Cuando sale el coche del sur si hay peatones esperando, el turno pasa a 2

Turno 2: Peatones → Cuando sale el peatón, si hay coches del norte esperando, el turno pasa a 0

Si alguna de las "colas" está vacía se pasa el turno a la otra y si las dos están vacías, el turno lo sigue manteniendo quien lo tenía. Pero siempre siguiendo el orden anterior.

De esta forma se evita que, por ejemplo, los coches se "compinchen" y cuando salga uno del norte entre uno del sur y viceversa. Es decir, no hay problema de inanición. Además, como el turno siempre cambia si hay elem. de alguna de las otras dos categorías se impide que siempre pasen elementos de un solo grupo.

AUSENCIA DE BLOQUEOS

Para evitar los bloqueos se considera que un coche puede pasar si no hay en el puente coches en la otra dirección y tampoco hay peatones y además tiene que ser su turno o que no haya "cola" para los otros dos posibles grupos (peatones y coches en la otra dirección)

Evitamos bloqueos considerando que se va a cambiar el turno si hay algún elemento de esa categoría en la cola.

De este modo se evita por ejemplo estar en el turno 0, con ncar_north == 0 y coches del sur y peatones esperando. Ya que aquí el turno no cambia y como hay 2 colas ninguno de los 2 entraría