

Solución sencilla:

Cada vez que un peatón o un coche quiera pasar, se asegura comprueba si hay algún proceso en la sección crítica. Si quiere entrar un peatón y hay peatones, el peatón entra, en el caso de que haya coches, espera. Lo mismo para los coches en cada dirección. Si hubiera infinitos coches o peatones, entonces podríamos llegar a un problema de inanición. En el caso de que el puente está vacío entra el primero que llegue.

* Invariante del monitor:

número de coches en el túnel ≥ 0

número de peatones en el túnel ≥ 0

num. de coches esperando ≥ 0

numero de peatones esperando ≥ 0

Si num. de peatones en el puente $> 0 \Rightarrow$ num. coches hacia el norte en el puente $= 0$
num. coches hacia el sur en el puente $= 0$

Si num. coches hacia el norte en el puente $> 0 \Rightarrow$ num. de peatones en el puente $= 0$
num. coches hacia el sur en el puente $= 0$

Si num. coches hacia el sur en el puente $> 0 \Rightarrow$ num. de peatones en el puente $= 0$
num. coches hacia el norte en el puente $= 0$

* Demostrar q es seguro

Como no deja entrar a peatones ni a coches a la vez, ni a coches que vayan en direcciones contrarias, será seguro.

* Inanición y Deadlock

El problema que nos daría este código es la posibilidad de llegar a inanición, para eso podemos crear una variable turno, q cada vez que pasen un máximo de cinco coches o personas cambie de turno.

También cambiar el turno cuando no haya nadie esperando.

Tendría que añadir contadores de cuanto gente está esperando.

Al poner el turno en orden evitaba la inanición porque todos los procesos tienen la posibilidad de avanzar.

Al poner q se cambie de turno si la lista está vacía, evitamos deadlocks, al igual que poniendo que al llegar al turno de un proceso puedan pasar como máximo 5 elementos esperando, en el caso de haber menos pasan todos.