

TAREA 2 Ejercicios de sincronización

Vazquez Sanchez Erick Alejandro

Problema a resolver

El elevador.

Lenguaje y Entorno de desarrollo

Utilice Python 3 para desarrollar este problema.

Para ejecutar mi programa se necesita hacer uso del comando `python3` seguido del nombre del archivo: **`python3 elevador.py`**

El programa entra en un ciclo infinito pues el elevador queda activo por si en algún momento se quiere acceder de nuevo a él, por lo que para concluir su ejecución se deberá intervenir en el proceso con un típico `CONTROL + C`

Estrategia de Sincronización

Utilice Semáforos y Mutex para implementar este ejercicio.

Refinamiento

Creo que si estoy implementando el refinamiento propuesto en el ejercicio, pues mi solución da prioridad al usuario más antiguo dentro del elevador yendo primero al piso que este usuario solicitó.

Dudas/Mejoras

En cuanto a dudas, en la línea 57 y 63 conforme a mi razonamiento creo que funciona tal y como lo coloque en los comentarios

```
# El elevador se duerme en lo que espera una llamada
mueve_elevador.acquire()

# Posteriormente se libera para mantener un balance
# pues cuando un usuario baja del elevador se libera
# de la llamada inicial que hizo el usuario y en cada
# iteracion se estaria bloqueando continuamente, creando un desbalance
mueve_elevador.release()
```

pues en la línea 154 despierto a mi semáforo por primera vez y en la 76 lo duermo cuando un usuario se baja del elevador. Y como comento en el código se debe de mantener un balance, pero quisiera saber si esto es correcto.

Y hablando de mejoras, ejecutando varias veces con distintos parámetros descubrí que algunas veces el usuario no se baja del elevador. Esto es debido a la operación pop que viéndolo cuidadosamente puede afectar en los dos for que tengo dentro del método funcionamiento de mi clase Elevador, pues utilizó un índice para hacer este pop, lo que hace que la lista tenga menos elementos y mi for cambie su funcionamiento drásticamente, pues por ejemplo, al tener una lista = [1,2,3,4] y un índice que lo acompañe y haciendo un pop del índice 2 quedaria [1,2,4], haciendo que la lista tenga 3 elementos y mi índice quede inservible para el índice 3 pues ya no existe ese elemento y lo ignoraria. Se me ocurre que se puede hacer que cambie de valor a None en lugar de eliminarlo y después en un ciclo aparte quitar esos None de la lista, pero el tiempo y sobretodo el estrés ya no me permite implementarlo, sin embargo probandolo varias veces y analizando su funcionamiento creo que funciona la mayoría de las veces. Este error solo me ocurrió 3 veces y 2 de esas veces cambiando la capacidad del elevador a 2 usuarios. Espero no marearlo con mi explicación. Anexo captura de este error.

```
erickescorpions@erick:~/workspaces/sistop-2022-1/tareas/2/...
Usuarios usando elevador: [[5, 3, '10']]
Elevador en piso 1
E: Moviendome de piso
Usuarios usando elevador: [[5, 3, '10']]
Elevador en piso 2
E: Moviendome de piso
Usuarios usando elevador: [[5, 3, '10']]
Elevador en piso 1
E: Moviendome de piso
Usuarios usando elevador: [[5, 3, '10']]
Elevador en piso 2
E: Moviendome de piso
Usuarios usando elevador: [[5, 3, '10']]
Elevador en piso 1
E: Moviendome de piso
Usuarios usando elevador: [[5, 3, '10']]
Elevador en piso 2
E: Moviendome de piso
Usuarios usando elevador: [[5, 3, '10']]
Elevador en piso 1
E: Moviendome de piso
Usuarios usando elevador: [[5, 3, '10']]
Elevador en piso 2
E: Moviendome de piso
Usuarios usando elevador: [[5, 3, '10']]
Elevador en piso 1
E: Moviendome de piso
Usuarios usando elevador: [[5, 3, '10']]
Elevador en piso 2
E: Moviendome de piso
Usuarios usando elevador: [[5, 3, '10']]
Elevador en piso 1
E: Moviendome de piso
^CException ignored in: <module 'threading' from '/usr/lib/python3.8/threading.p
y'>
Traceback (most recent call last):
  File "/usr/lib/python3.8/threading.py", line 1388, in _shutdown
    lock.acquire()
KeyboardInterrupt:
→ VazquezErick git:(main) X
```