



**Universidad Nacional
Autónoma de México**

Facultad de Ingeniería

Sistemas Operativos



Integrantes:

Chagoya González Leonardo
Cervantes García Eduardo

Grupo 06

Profesor: Gunnar Eyal Wolf Iszaevich

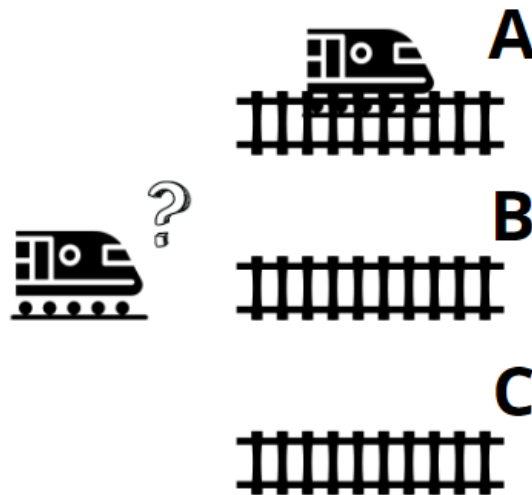
Semestre 2024-1

Planteamiento del problema

¿Cuántas veces no hemos ocupado el metro?, ¿Cuántas veces no hemos estado en una terminal y observamos como dos o inclusive tres metros se encuentran listos para salir? Las cosas en la ciudad de México pueden parecer un perfecto caos y el metro no es la excepción, con este código intentaremos simular el comportamiento de los vagones del metro y de las personas que lo abordan.

Una terminal de metro tiene 3 posibles carriles A, B y C en los cuales un vagón puede colocarse para recibir gente, en caso de que exista gente esperando por el vagón. El metro se colocará en uno de los carriles que estén libres, abrirá sus puertas por unos segundos, hasta este momento las personas podrán abordar el metro correspondiente, el metro cerrará sus puertas y procederá a marcharse.

Es importante resaltar que dos metros no pueden llegar a un mismo carril, no queremos accidentes, así como dos metros no pueden salir al mismo tiempo. De igual forma una persona elegirá un metro por el cual abordar y esperará a que este llegue para abordarlo, nuevamente no queremos accidentes.



Entorno y dependencias

El código fue realizado en el lenguaje de programación de Python en su versión 3.9.13, en el editor de código Visual Studio Code . Se utilizaron las bibliotecas threading, random y time. Es por ello que para ejecutar el código es recomendable tener Python 3.9 o una versión superior.

En cuanto al Sistema Operativo, el código fue creado y ejecutado en Windows.

Identificación del estado compartido (variables o estructuras globales)

```
id_carriles = {'A':0, 'B':1, 'C':2}
```

```
id_carriles_inv = {0:'A', 1:'B', 2:'C'}
```

Los diccionarios son globales debido a que es importante que todo el programa pueda conocer los carriles disponibles y sus identificaciones

```
carriles = ['Empty','Empty','Empty']
```

La lista de carriles es global debido a que solo existe una estación, todo el programa usa la misma estación, por tanto el mismo arreglo. Cualquier cambio y su estado debe ser el mismo para todos los hilos de ejecución, ya que todos los hilos o trenes pelean por esos 3 espacios.

```
listaEspera=[]
```

La lista de espera es global por razones similares a los carriles, es necesario acceder a ella en todo momento para cualquier tren y su función. A la vez sólo puede haber una ya que todas las personas esperan el metro en la misma terminal.

```
listaEsperaA=[]
```

```
listaEsperaB=[]
```

```
listaEsperaC=[]
```

Estas listas son globales porque todos los trenes, para el hecho de solo evaluar si continúan o no trabajando, necesitan saber si hay personas. Además, de que necesitan saber que personas subirán a su tren según el andén en el que esperaron y esto debe ser coherente con la decisión tomada por la persona y la lista de espera general de la estación. Por ello, se necesitan listas que contengan esta información general y actualizada para cualquier conductor o trabajador que la consulte en cualquier momento.

Descripción algorítmica del avance de cada hilo/proceso

Este ejemplo se realizará tomando en cuenta solo una persona o un hilo de persona y un tren o hilo de trenes, de esta manera será más sencillo comprender lo que pasa en cada hilo y se abarcan dos hilos que hacen distintas tareas en el programa.

Lo primero que se comienza a instanciar son los hilos de personas, así que supongamos que una persona es lo primero que entra en ejecución en el

programa. La persona llama a su función de llegada, en ella se usa el mutex “m_persona” el objetivo de esta función es dar información al usuario, agregar a la persona a la espera de la estación y llamar a la siguiente función para el hilo. El mutex es usado para que ninguna otra persona pueda entrar a formarse a la estación antes de esta que llegó primero o que lo puedan eliminar por error de la espera ya que el planificador podría cortar la ejecución del hilo en cualquier momento.

El hilo entra en la función que gestiona a la persona, en ella la persona espera cierto tiempo en la estación hasta que decide a que anden ir, si la persona no esta ya en un andén, va a alguno a esperar abordar el metro, entonces la persona se añade a una lista de los que esperan en cierto andén. Se manda una señalización al conductor del tren, al cual se le dice que ya hay personas en el andén, por lo cual luego de esperar un tiempo, puede salir de la terminal sin problema.

Pensemos que se tuvo la suerte de una ejecución que parece secuencial, en ella lo siguiente que pasa es que se instancia el hilo del tren que la persona que ya está en la estación, está esperando. Entonces, lo primero que hace el conductor es verificar que hay gente para saber si vale la pena dar servicio, luego interfiere el “Múltiplex de la estación” o la comunicación que le indica al conductor que hay un espacio vacío en la estación, por lo tanto puede entrar. Dentro de la estación, al conductor se le asigna porque carril saldrá, el conductor verifica que no haya otro tren en el carril, entra y marca que él está ahí con el mutex “m_carril”. En el carril, vuelve a comprobar que aún haya gente, avisa del estado del tren y abre las puertas cierto tiempo.

Después de esperar un tiempo, se prepara la salida del tren. El hilo del tren, llama a la función o a la salida con toda su información. Espera un poco más con la puerta abierta, y verifica haber recibido la señal de que hay al menos un pasajero dentro y ninguno en la puerta o algo por el estilo. Con un positivo en la señal recibido, el tren toma la salida con el mutex de señalización mandado por los pasajeros o pasajeros en este caso. Debido a que solo puede haber una salida y está disponible, el tren toma el permiso de salida y lo bloquea para los demás con el mutex m_salida. Entonces, el tren actualiza que los pasajeros que esperaban en su respectivo andén salen de la terminal, los quita de la lista de espera de la estación y del andén. En su salida, libera el permiso de salida y un cupo para que pueda haber un tren más en los tres lugares de la estación, con ello también actualiza la información de las pantallas de la estación

Finalmente, el tren logró atender a los pasajeros con éxito y ahora dará una vuelta por la línea que atiende, para cuando termine vuelva a formarse en la terminal hasta que no haya más pasajeros que atender o termine su horario de servicio.

Descripción de la interacción entre ellos (sea mediante los mecanismos de sincronización o de alguna otra manera)

Utilizando Multiplex estacion

El principal y primer problema de concurrencia lo encontramos en los vagones de metro, hay que preparar los vagones para que lleguen a su respectivo carril. Para ello nos apoyamos de un múltiplex inicializado en 3, de esta manera observamos si hay algún carril disponible. En nuestro código multiplex será **estación**.

Ojo esto solo prepara al metro, más adelante explicaremos cómo garantizamos que no exista dos vagones por carril

Utilizando Mutex m_eleccion

Una vez que el metro puede entrar a la estación ,nuestro código elige de forma aleatoria que el carril al cual quiere entrar el metro, esto se logra mediante la funcion **carrilEleccion()**. Para proteger la variable **carril** utilizamos el mutex llamado **m_eleccion**.

Utilizando Mutex m_carrilA , m_carrilB , m_carrilC

Una vez que se elige el carril por el cual llegará al metro, debemos asegurarnos que no exista otro metro en dicho carril, para garantizar este comportamiento utilizamos 3 mutex refiriéndonos a cada carril. Manteniendo el mutex de ese carril podemos ejecutar las operaciones del metro en el carril , las operaciones será mostrar el estado actual de los carriles, que un metro se coloque en su respectivo carril, el metro abrirá sus puertas las cerrará y posteriormente saldrá, esto mediante las funciones **gestiondeCarriles()** y **salida ()**.

Utilizando semáforos semAndenA, semAndenB, semAndenC

Para hablar del funcionamiento de estos semáforos necesitamos hablar de persona, ya que hasta ahora solo hemos hablado de metro.

Una persona X llegará a la estación y se agrega a la lista de espera,llamada **listaEspera** ,esto con el fin de saber que personas se encuentran esperando por abordar un metro, posteriormente entrará a una función llamada **gestionPersona()** , protegemos estas acciones un con el mutex **m_persona**,ya que estamos accediendo a una variable compartida como lo es **listaEspera** así como el comportamiento de una persona al elegir un carril y/o abordar el metro.

Dentro de **gestionPersona()** se elige de forma aleatoria por cuál carril decide esperar al metro y verificamos si no está ya esperando el vagón

Cuando el metro abre sus puertas es momento para que las personas que estaban esperando el metro por ese carril puedan abordarlo, es por ello que cuando se ejecuta la función **salida()** se toma el respectivo semáforo semAnden para que espere a que las personas entren y en **gestionPersona()** se libera indicando que el metro puede cerrar sus puertas. Aquí podemos observar un caso de señalización.

Después de que un metro cierra sus puertas, verificamos el estado actual de los carriles, así como el de los pasajeros que se marchan y podemos hacer que un metro salga, protegiendo esta acción mediante un mutex, para que dos metros no puedan salir al mismo tiempo

Ejecución

```
---El estado de los carriles es:      A:Empty      B:Empty      C:Empty---
Metro 2 puede entrar en la estación.

---El estado de los carriles es:      A:Empty      B:Empty      C:Empty---
Metro 0 llegó al carril B

---El estado de los carriles es:      A:Empty      B:Ocupado por metro:0  C:Empty---
---El estado de los carriles es:      A:Empty      B:Ocupado por metro:0  C:Empty---

Metro 0 abre las puertas
Metro 1 llegó al carril C

---El estado de los carriles es:      A:Empty      B:Ocupado por metro:0  C:Ocupado por metro:1---
Metro 1 abre las puertas
**Persona 42 decide abordar el metro en el anden B.
Persona 43 llega a la estación
**Persona 43 decide abordar el metro en el anden A.
Persona 44 llega a la estación
Metro 1 cierra las puertas
**Persona 44 decide abordar el metro en el anden B.

---El estado de los carriles es:      A:Empty      B:Ocupado por metro:0  C:Ocupado por metro:1---
Persona 45 llega a la estación

      Metro 1 sale de la terminal dejando libre el carril C
Las personas con los ID de tarjeta:[] salen de la terminal por el carril C
```

```

---El estado de los carriles es:      A:Empty      B:Ocupado por metro:0  C:Empty---
Metro 1 puede entrar en la estación.
Metro 0 cierra las puertas

---El estado de los carriles es:      A:Empty      B:Ocupado por metro:0  C:Empty---

      Metro 0 sale de la terminal dejando libre el carril B
Las personas con los ID de tarjeta:[42, 44] salen de la terminal por el carril B

---El estado de los carriles es:      A:Empty      B:Empty      C:Empty---
Metro 0 puede entrar en la estación.

---El estado de los carriles es:      A:Empty      B:Empty      C:Empty---
Metro 2 llegó al carril B

---El estado de los carriles es:      A:Empty      B:Ocupado por metro:2  C:Empty---
---El estado de los carriles es:      A:Empty      B:Ocupado por metro:2  C:Empty---

Metro 2 abre las puertas
Metro 1 llegó al carril C

---El estado de los carriles es:      A:Empty      B:Ocupado por metro:2  C:Ocupado por metro:1---
Metro 1 abre las puertas
**Persona 45 decide abordar el metro en el anden B.
Persona 46 llega a la estación
**Persona 46 decide abordar el metro en el anden A.
Persona 47 llega a la estación

```

```

---El estado de los carriles es:      A:Empty      B:Ocupado por metro:2  C:Ocupado por metro:1---
Persona 48 llega a la estación

      Metro 2 sale de la terminal dejando libre el carril B
Las personas con los ID de tarjeta:[45, 47] salen de la terminal por el carril B

---El estado de los carriles es:      A:Empty      B:Empty      C:Ocupado por metro:1---
Metro 2 puede entrar en la estación.
Metro 1 cierra las puertas

---El estado de los carriles es:      A:Empty      B:Empty      C:Ocupado por metro:1---

      Metro 1 sale de la terminal dejando libre el carril C
Las personas con los ID de tarjeta:[] salen de la terminal por el carril C

---El estado de los carriles es:      A:Empty      B:Empty      C:Empty---
Metro 1 puede entrar en la estación.

---El estado de los carriles es:      A:Empty      B:Empty      C:Empty---
Metro 0 llegó al carril C

---El estado de los carriles es:      A:Empty      B:Empty      C:Ocupado por metro:0---
---El estado de los carriles es:      A:Empty      B:Empty      C:Ocupado por metro:0---

Metro 2 llegó al carril A
Metro 0 abre las puertas

```

```

---El estado de los carriles es:      A:Ocupado por metro:2  B:Empty      C:Ocupado por metro:0---

---El estado de los carriles es:      A:Ocupado por metro:2  B:Empty      C:Ocupado por metro:0---

Metro 2 abre las puertas
Metro 1 llegó al carril B

---El estado de los carriles es:      A:Ocupado por metro:2  B:Ocupado por metro:1  C:Ocupado por metro:0---

Metro 1 abre las puertas
**Persona 48 decide abordar el metro en el andén B.
Persona 49 llega a la estación
**Persona 49 decide abordar el metro en el andén B.
Persona 50 llega a la estación
Metro 2 cierra las puertas
Metro 1 cierra las puertas

---El estado de los carriles es:      A:Ocupado por metro:2  B:Ocupado por metro:1  C:Ocupado por metro:0---

---El estado de los carriles es:      A:Ocupado por metro:2  B:Ocupado por metro:1  C:Ocupado por metro:0---

Metro 2 sale de la terminal dejando libre el carril A      Metro 0 cierra las puertas

---El estado de los carriles es:      A:Ocupado por metro:2  B:Ocupado por metro:1  C:Ocupado por metro:0---
Las personas con los ID de tarjeta:[35, 38, 40, 41, 43, 46] salen de la terminal por el carril A

```

```

---El estado de los carriles es:      A:Empty      B:Ocupado por metro:1  C:Ocupado por metro:0---

Metro 1 sale de la terminal dejando libre el carril B

Metro 2 puede entrar en la estación.
Las personas con los ID de tarjeta:[48, 49] salen de la terminal por el carril B

---El estado de los carriles es:      A:Empty      B:Ocupado por metro:1  C:Ocupado por metro:0---

---El estado de los carriles es:      A:Empty      B:Empty      C:Ocupado por metro:0---

Metro 2 llegó al carril A
Metro 1 puede entrar en la estación.

---El estado de los carriles es:      A:Ocupado por metro:2  B:Empty      C:Ocupado por metro:0---

Metro 0 sale de la terminal dejando libre el carril C

Metro 2 abre las puertas
Las personas con los ID de tarjeta:[] salen de la terminal por el carril C

---El estado de los carriles es:      A:Ocupado por metro:2  B:Empty      C:Empty---

Metro 0 puede entrar en la estación.
**Persona 50 decide abordar el metro en el andén C.
Persona 51 llega a la estación
**Persona 51 decide abordar el metro en el andén C.
Persona 52 llega a la estación
**Persona 52 decide abordar el metro en el andén A.
Persona 53 llega a la estación
Metro 2 cierra las puertas

```



```

---El estado de los carriles es:      A:Ocupado por metro:2  B:Empty      C:Empty---

      Metro 2 sale de la terminal dejando libre el carril A
Las personas con los ID de tarjeta:[52] salen de la terminal por el carril A

---El estado de los carriles es:      A:Empty      B:Empty      C:Empty---
Metro 2 puede entrar en la estación.

---El estado de los carriles es:      A:Empty      B:Empty      C:Empty---
Metro 1 llegó al carril A

---El estado de los carriles es:      A:Ocupado por metro:1  B:Empty      C:Empty---
---El estado de los carriles es:      A:Ocupado por metro:1  B:Empty      C:Empty---

Metro 1 abre las puertas
Metro 0 llegó al carril C

---El estado de los carriles es:      A:Ocupado por metro:1  B:Empty      C:Ocupado por metro:0---
Metro 0 abre las puertas
**Persona 53 decide abordar el metro en el andén A.
Persona 54 llega a la estación
**Persona 54 decide abordar el metro en el andén C.
Persona 55 llega a la estación
Metro 0 cierra las puertas

```

El código seguirá ejecutándose hasta que no exista ninguna persona por abordar

```

---El estado de los carriles es:      A:Ocupado por metro:2  B:Empty      C:Ocupado por metro:0---

      Metro 0 sale de la terminal dejando libre el carril C
Las personas con los ID de tarjeta:[92] salen de la terminal por el carril C

---El estado de los carriles es:      A:Ocupado por metro:2  B:Empty      C:Empty---
Metro 0 puede entrar en la estación.

---El estado de los carriles es:      A:Ocupado por metro:2  B:Empty      C:Empty---
Metro 1 llegó al carril C

---El estado de los carriles es:      A:Ocupado por metro:2  B:Empty      C:Ocupado por metro:1---
Metro 1 abre las puertas
**Persona 98 decide abordar el metro en el andén C.
Persona 99 llega a la estación
**Persona 99 decide abordar el metro en el andén B.
Metro 1 cierra las puertas

---El estado de los carriles es:      A:Ocupado por metro:2  B:Empty      C:Ocupado por metro:1---

      Metro 1 sale de la terminal dejando libre el carril C
Las personas con los ID de tarjeta:[98] salen de la terminal por el carril C

---El estado de los carriles es:      A:Ocupado por metro:2  B:Empty      C:Empty---
Metro 1 puede entrar en la estación.

```