Universidad Nacional Autónoma de México

Facultad de Ingeniería

SISTEMAS OPERATIVOS

Proyecto No.2 Una situación cotidiana paralelizable

Alumnos:
Christian Abraham Lara Aguilar
Lissett Zuñiga Reyes

Grupo 06
Semestre 2024-1

Fecha de entrega: 06 de Noviembre del 2023

1. Situación:

Abordar y descender de los vagones del metro de forma ordenada y segura.

2. Funcionamiento

2.1. Al estar el metro cerrado este deberá abrir sus puertas

2.2. Al estar abiertas las puertas tras haber estado cerradas, deberá esperar a que todos los pasajeros bajen del metro sin que ninguno pueda subir a él.

2.3. Los pasajeros que estén esperando a subir al metro, deberán esperar de modo que primero se permita la salida para poder ingresar.

Figura 1: Pasajeros esperando de forma oportuna su turno.

2.4. Ya que todos los pasajeros que se encontraban en espera de abordar el vagón ingresan a este, las puertas deben cerrarse.

El número de pasajeros puede cambiar, es decir, puede haber un número no definido de cuántos pasajeros pueden estar en el metro.

3. Descripción de los mecanismos de sincronización empleados

El código utiliza dos mecanismos de sincronización para garantizar que las operaciones concurrentes se realicen de manera segura.

- Bloqueo.

  Se emplea un objeto de bloqueo (mutex). Este mecanismo de sincronización se utiliza para garantizar la exclusión mutua, lo que significa que solo un hilo puede adquirir el bloqueo a la vez. Cuando un hilo adquiere el bloqueo, se asegura de que ninguna otra operación crítica pueda ocurrir hasta que el hilo que tiene el bloqueo lo libere.

- Condición.

  El objeto de condición se utiliza para coordinar hilos y notificarles cuando ciertas condiciones se cumplen o cambian. En este caso, se utiliza para coordinar el momento en que los pasajeros pueden subir o bajar del metro.

4. Lógica de operación

- Abrir puertas: Este método se utiliza para abrir las puertas del metro. Cuando se llama a este método, se establece el atributo self.puertas-abiertas en True, indicando que las puertas están abiertas.

- Cerrar puertas: Este método se utiliza para cerrar las puertas del metro. Cuando se llama a este método, se establece el atributo self.puertas-abiertas en False, indicando que las puertas están cerradas.

- Bajar pasajero: Este método simula a un pasajero que desea bajar del metro. Para que un pasajero pueda bajar, se deben cumplir las siguientes condiciones: Las puertas del metro deben estar abiertas (self.puertas-abiertas == True).

  No debe haber pasajeros subiendo al metro(self.pasajeros-subiendo==0).

  Si alguna de estas condiciones no se cumple, el hilo se bloquea y espera utilizando self.cond.wait(). Una vez que las condiciones son satisfechas, se incrementa el contador de pasajeros bajando (self.pasajeros-bajando').

- Subir pasajero: Este método simula a un pasajero que desea subir al metro. Para que un pasajero pueda subir, se deben cumplir las siguientes condiciones:

  - Las puertas del metro deben estar abiertas (self.puertas-abiertas == True).

  - No debe haber pasajeros bajando del metro (self.pasajeros-bajando == 0).

  Si alguna de estas condiciones no se cumple, el hilo se bloquea y espera utilizando self.cond.wait(). Una vez que las condiciones son satisfechas, se incrementa el contador de pasajeros subiendo (self.pasajeros-subiendo).

- Terminar bajar: Este método se utiliza para indicar que un pasajero ha terminado de bajar del metro. Se reduce el contador de pasajeros dentro del metro (self.pasajeros-dentro) y el contador de pasajeros bajando (self.pasajeros-bajando).

- Terminar subir: Este método se utiliza para indicar que un pasajero ha terminado de subir al metro. Se incrementa el contador de pasajeros dentro del metro (self.pasajeros-dentro) y se reduce el contador de pasajeros subiendo (self.pasajeros-subiendo).

4.1. Descripción algorítmica del avance de cada hilo/proceso

- **Proceso de Subida de Pasajeros:**

  Un hilo que representa a un pasajero que desea subir al metro. El hilo verifica si las puertas del metro están abiertas y si no hay pasajeros bajando del metro. Si alguna de estas condiciones no se cumple, el hilo se bloquea y espera hasta que ambas condiciones sean verdaderas. Cuando se cumplen las condiciones, el hilo incrementa el contador de pasajeros subiendo. Cuando las puertas se abren y el hilo es notificado, se considera que el pasajero ha subido al metro.

- **Proceso de Bajada de Pasajeros:**

  Un hilo que representa a un pasajero que desea bajar del metro. El hilo verifica si las puertas del metro están abiertas y si no hay pasajeros subiendo al metro. Si alguna de estas condiciones no se cumple, el hilo se bloquea y espera hasta que ambas condiciones sean verdaderas. Cuando se cumplen las condiciones, el hilo incrementa el contador de

pasajeros bajando. Cuando las puertas se abren y el hilo es notificado, se considera que el pasajero ha bajado del metro.

- **Operaciones de Apertura y Cierre de Puertas:**

  La apertura y cierre de puertas se manejan mediante los métodos abrir-puertas() y cerrar-puertas(). Cuando las puertas se abren o se cierran, los hilos que están esperando en self.cond.wait() pueden continuar o bloquearse según las condiciones establecidas.

- **Actualización de Contadores de Pasajeros:**

  Los métodos terminar-subir() y terminar-bajar() se utilizan para actualizar los contadores de pasajeros dentro del metro, pasajeros subiendo y bajando, para reflejar cambios en el estado del metro.

5. Descripción del entorno de desarrollo, suficiente para reproducir una ejecución exitosa.

5.1. ¿Qué lenguaje emplea? ¿Qué versión?

El lenguaje de programación utilizado fue Python 3.10.0

5.2. ¿Qué bibliotecas más allá de las estándar del lenguaje?

1. *threading*
2. *time*
3. *random*

5.3. Bajo qué sistema operativo / distribución lo desarrollaron y/o probaron

Sistema operativo Windows.

6. Ejemplos o pantallazos de una ejecución exitosa

Para poder visualizar la ejecución de este programa solo se debe ejecutar el archivo de nombre python $proyecto2_SO.py$

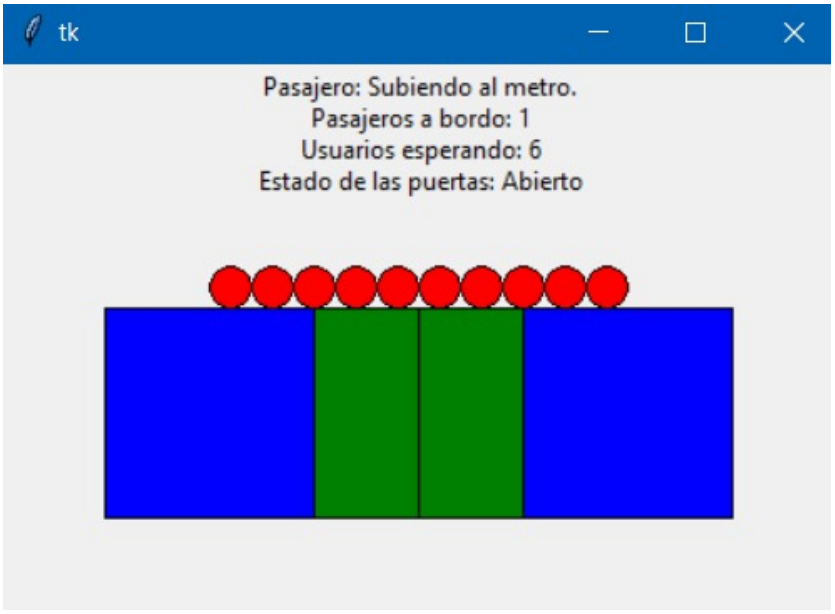

Figura 2: Ejecución del programa.

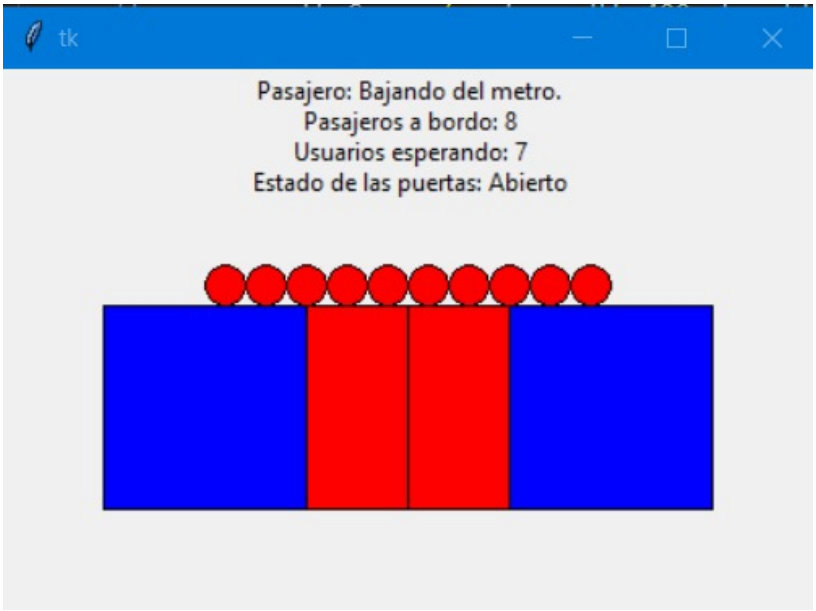

Figura 3: Ejecución del programa.