Trabajo Obligatorio - Segunda Entrega

Matías Gonnet, Nicolás Pavon, Adrián Tesore
Mayo 2020.

Universidad Católica del Uruguay
Montevideo, Uruguay
Sistemas Operativos I

Tabla de contenidos

Capítulo 1

Identificación de Recursos

1. Se cuenta con diez casillas de peaje, que serán las encargadas de leer la matrícula de los vehículos que desean transitar el peaje. En todo momento habrá una cantidad asignada al sentido este, y el resto estará asignada en sentido oeste. Se puede decidir ajustar esta cantidad, en función del volumen de tráfico.
2. Se cuenta con una pantalla en cada senda en la boca del peaje, que indicará a cuál casilla de peaje ingresar.
3. Se cuenta con un sensor por cada senda a 100 metros de la boca del peaje, que identificará a los vehículos que se están acercando, con el fin de identificar tempranamente que casilla de peaje indicarles.

Identificación de Procesos

1. Proceso que decide si se debe permitir el pasaje o no de un vehículo.
2. Evaluar nivel de congestión de las vías de circulación. Las mismas pueden estar con tráfico fluido, congestionadas, o muy congestionadas. Este proceso se realiza periódicamente.
3. Proceso para cambiar el sentido de una casilla. El mismo agrega una casilla al sentido que haya sido identificado como muy congestionado, siempre y cuando el otro sentido se encuentre con un tráfico fluido.
4. Identificar los vehículos que están por entrar a la zona de peaje (Sensor ubicado a 100 metros)

5. Decidir a qué casilla de peaje enviar el próximo vehículo que entre a la zona de peaje, teniendo en cuenta el input de todos los sensores, y el tiempo de espera de las casillas.

Criterios de optimización

1. Se identifica la necesidad de diferenciar los vehículos que desean atravesar el peaje según su categoría. Los mismos pueden ser:
    a. Vehículos de emergencia: Ambulancia, Policía, Bomberos
    b. Ómnibus y transporte público
    c. Auto particular
    d. Camión liviano
    e. Camión pesado

2. Tiempo de espera. Además de considerar el tipo de vehículo, en caso de conflicto, se ha de considerar la diferencia de tiempo que cada uno tendrá que esperar en la casilla. Si esta diferencia supera un cierto umbral, el vehículo de menor prioridad puede ser considerado como de una categoría superior.

    Estadísticamente se ha observado que los distintos tipos de vehículo demoran distinta cantidad de tiempo en cruzar las casillas de peaje (debido a sus velocidades de marcha y aceleración).
    a. Ambulancias: 2 segundos
    b. Ómnibus 6 segundos
    c. Autos 4 segundos
    d. Camiones livianos 5 segundos
    e. Camiones pesados 7 segundos

Ordenación de los criterios de optimización

f. Vehículos de emergencia

g. Proceso de evaluación de congestión

h. Proceso de cambio de sentido de casilla

i. Ómnibus y transporte público

j. Auto particular

k. Camión liviano

l. Camión pesado

Capítulo 2

Identificación de conceptos similares

La situación planteada es similar al concepto de productor - consumidor estudiado en sistemas operativos. A alto nivel, los detectores de vehículos a la entrada de la ruta "producen" vehículos, que deberán ser "consumidos" por las casillas de peaje. Concretamente, serían cuatro productores (cuatro sensores, un por carril, siendo dos carriles de ruta en cada sentido), y diez consumidores (cada una de las casillas de peaje).

Las casillas de peaje no consumen inmediatamente cada vehículo, como se explicó anteriormente, cada tipo de vehículo tarda un tiempo (una cantidad discreta de segundos, simplificados como ciclos en este análisis) distinto en ser procesado. De ahí que sea importante aplicar un criterio para decidir a qué casilla de peaje enviar cada nuevo vehículo, ya que es posible que todas las casillas estén ocupadas si llegan más vehículos que los que se pueden procesar en un ciclo dado. Estos criterios son los mencionados en el *capítulo 1*.

Este problema es similar al planteado en el concepto de Planificadores de Largo Plazo en Sistemas Operativos. Los vehículos van a competir por utilizar el tiempo escaso de procesamiento que cada casilla provee, necesitando un planificador encargado de asignar cada vehículo a una casilla. En el análisis de alternativas se estudiará cómo aplicar el concepto de planificadores MLQ, Shortest Job First (SJF) y First Come First Served (FCFS) al problema del peaje.

Descripción de las alternativas a implementar

En orden de diseñar las alternativas se debe tener en cuenta algunos criterios:

Se debe dar prioridad como se menciona en el Capítulo 1 a los automóviles de emergencia, siempre minimizando el tiempo de concurrencia en el peaje, evitando así atascos en los ingresos a las casillas. Se identifica a continuación:

Uso de planificador First Come First Served

Para esta alternativa se plantea la formación de una única vía, a la cual ingresarán los vehículos en orden de llegada y serán asignados a la primer casilla libre en cuanto hubiera una. Una de sus ventajas es que la implementación del algoritmo se realiza fácilmente a través de una cola FIFO y los procesos son ejecutados en el orden que llegan a la cola. Pero su desventaja es que se tratan por igual a todos los vehículos sin tener en consideración los criterios de prioridad definidos en el cap. 1

Shortest Job First - SJF

Para este planificador se tomará en cuenta el tiempo promedio que tiene cada vehículo en pasar por la casilla. Para esto antes de que el vehículo pase por el planificador que le asigna la casilla correspondiente, los vehículos se van a ir formando en varias filas según su tiempo esperado de demora (ómnibus, auto o vehículos de emergencia). Luego a medida que se vayan liberando las casillas se irá mandando al que le lleva menos tiempo procesar.

Si bien esta solución cumple con atender primero a los autos de emergencia y por último a los vehículos pesados, nos encontramos con el problema de que si llegaran a ingresar en un período de tiempo miles de autos juntos y/o vehículos de emergencias los ómnibus/camiones nunca pasarán y se bloquearán en la fila previa a la asignación de casilla.

MLQ

Este planificador formará varias colas. En cada una se irán colocando los vehículos del mismo tipo, según su orden de llegada (FCFS). Las colas estarán ordenadas según el criterio de prioridad definido en el capítulo 1, es decir:

a. Vehículos de emergencia
b. Ómnibus y transporte público
c. Auto particular
d. Camión liviano
e. Camión pesado

Luego, se solicitará a los vehículos de la cola a) que se dirijan a la primera casilla libre, a medida que vayan liberando. Si no hay ningún vehículo en la cola a), se solicita a los de la cola b) y así sucesivamente. Si apareciera un vehículo en una cola superior, cuando se están tomando de una cola inferior, se vuelve a comenzar el proceso, tomando desde la cola superior. De esta forma, se garantiza que los vehículos de emergencias sean atendidos inmediatamente y que los ómnibus tengan prioridad por sobre los vehículos particulares. Es decir, para el planificador de colas, se utiliza un planificador por prioridades.

Justificación de la alternativa a implementar:

Se optó por seleccionar la alternativa que utiliza el algoritmo de planificación “MLQ” sin retroalimentación.

Se descarta el planificador FCFS ya que este no puede satisfacer los criterios de optimización elegidos, que establecen una relación de prioridad entre distintos tipos de vehículos. Esto es debido a que el único elemento considerado por este planificador es el orden de llegada.

Se descarta el planificador SJF ya que si bien permite diferenciar tipos de vehículos según su tiempo estimado de duración, el orden de prioridad que crea, no se corresponde con los criterios establecidos en el capítulo 1. Por ejemplo, el planificador SJF tomaría primero a un ómnibus que a un auto, cuando se estableció lo contrario.

El planificador MLQ permite garantizar la priorización de un tipo de vehículo por sobre otro, de acuerdo a los criterios establecidos en el capítulo 1, siendo por esto, la alternativa elegida. Se consideró la opción de utilizar la retroalimentación, de acuerdo a lo planteado en el teórico, sin embargo se ve difícil aplicar en la práctica el cambio de vehículo entre colas

Descripción de escenarios:

A continuación se plantean situaciones en las que se indica el ciclo en el que ingresan una cantidad de vehículos de distinto tipo y el ciclo en el que se espera sean procesados por las casillas. Se asume que al comienzo todas las casillas están vacías, habiendo cinco casillas. Cada tipo de vehículo tarda tantos ciclos en procesarse como los segundos indicados en el capítulo 1.

Vehículos del mismo tipo al mismo tiempo

En ciclo 1 ingresan los autos “A”, “B”, “C”, “D”, “E”, “F”.

En ciclo 4 se termina de procesar los vehículos “A”, “B”, “C”, “D”, “E”

En ciclo 8 se termina de procesar el vehículo “F”.

Nota: Entre vehículos del mismo tipo, en el mismo ciclo, se respetó el orden de llegada

Una ambulancia y cinco autos al mismo tiempo

En ciclo 1 ingresa la ambulancia “E1” y los autos “A1”, “A2”, “A3”, “A4”, “A5”.

En ciclo 2 se termina de procesar “E1”.

En ciclo 4 se termina de procesar “A1”, “A2”, “A3”, “A4”.

En ciclo 6 se termina de procesar “A5”.

Nota: Si en un ciclo entran más vehículos que casillas, el de menor prioridad queda pendiente hasta que se libere una casilla.

Auto y ambulancia alternado

En ciclo 1 ingresa un auto “A1”.

En ciclo 2 ingresa una ambulancia “E1”.

En ciclo 3 ingresa un auto “A2”.

En ciclo 4 se termina de procesar “A1”.

En ciclo 4 se termina de procesar “E1”

En ciclo 7 se termina de procesar “A2”.

Nota: Si entre dos vehículos de la misma prioridad, entra uno de mayor prioridad, se vuelve a tomar el de mayor prioridad.

Ómnibus, auto y emergencia

En ciclo 1 ingresan ómnibus “O1”, “O2”, “O3”, “O4”.

En ciclo 2 ingresan auto “A1” y emergencia “E1”.

En ciclo 4 se termina de procesar “E1”

En ciclo 7 se terminan de procesar “O1”, “O2”, “O3”, “O4”.

En ciclo 8 se termina de procesar “A1”.

Nota: Si cuatro de las casillas están ocupadas, la casilla restante se utiliza respetando la prioridad entre vehículos.

Descripción de la alternativa temporal implementada en código JAVA

Para esta entrega se optó por simplificar la realidad del sistema, en la que existe una única vía o sentido, no existe un criterio de prioridad entre los distintos vehículos y en la que se aplica el algoritmo de planificación FCFS.

En esta solución el "planificador" envía todos los vehículos creados por el productor a una única fila, de la cual el primer vehículo acudirá a la primera casilla vacía que lo solicite, ni bien haya despachado al vehículo anterior.

El propósito de esta versión es demostrar el correcto uso de los semáforos, del reloj y de la funcionalidad más básica del peaje.

Instructivo para correr el programa

Cuando se ejecuta el programa el mismo queda esperando por una entrada, que sería la información de un vehículo en el siguiente formato csv: tiempoReloj;matrícula;tipoAuto

En la carpeta Tests/ se ve que hay 2 archivos de pruebas para el programa, uno simula tráfico liviano y el otro tráfico pesado.

El programa deja de recibir autos, cuando lee en el archivo un *

Ejecución

En la consola git bash o linux pararse en el proyecto y colocar el siguiente comando:

```
cat Tests/pruebas_trafico_liviano.csv | java -jar ProyectoSO/dist/ProyectoSO.jar
```

Diagrama de ejemplo