



Programación de Sistemas y Concurrencia

Examen 1ª convocatoria ordinaria
Curso 2020-2021

APELLIDOS _____ NOMBRE _____

DNI _____ ORDENADOR _____ GRUPO/TITULACIÓN _____

Bloque 2 - Concurrencia

Descripción del sistema

Una empresa de transportes tiene una flota de furgonetas con tecnología V2V (Vehicle to Vehicle). Han desarrollado una aplicación que permite a las furgonetas coordinarse para formar un convoy capaz de ir de forma autónoma de un origen a un destino.

Vamos a realizar una versión muy simplificada de este sistema. El convoy va a estar formado por todas las furgonetas de la flota. La **furgoneta líder será la primera** que se una al convoy, y esperará a que el resto de las furgonetas se unan. Cuando el convoy está completo, la furgoneta líder calcula la ruta y se pone en marcha, de manera que el resto de furgonetas la siguen. Cuando la furgoneta líder llega al destino, avisa al resto de furgonetas y espera a que abandonen el convoy. Se proporcionan las siguientes clases:

- Furgoneta: Es una clase que hereda de Thread y que modela el comportamiento tanto de la Furgoneta líder como de las seguidoras. Observa que el comportamiento definido en el método run tiene en cuenta las dos posibilidades que pueden ocurrir cuando una Furgoneta se une al Convoy: (1) que sea la primera, en cuyo caso se convierte en la **líder**; (2) o que no sea la primera, en cuyo caso será una **seguidora**. La furgoneta líder se espera en el método calcularRuta hasta que el resto de furgonetas se hayan unido. Una vez que se han unido todas, el tiempo del viaje se modela con un Thread.sleep. Al llegar al destino, la furgoneta líder avisó a las seguidoras. Las furgonetas seguidoras simplemente siguen a la líder hasta llegar al destino. Observa que cuando todas las furgonetas llegan al destino y abandonan el convoy, las hebras finalizan.
- Convoy: Esta clase modela el recurso compartido. Ofrece a las furgonetas los siguientes métodos que deben ser implementados:

public Convoy (**int** tam): El constructor del Convoy recibe como parámetro el tamaño de la flota de furgonetas. Inicialmente no hay furgonetas en el Convoy.

public int unir(**int** id): Una Furgoneta utiliza este método para unirse al Convoy. La primera en unirse se convierte en la **líder**, y el resto se une como seguidora. La última furgoneta que entre en la flota avisa a la líder de que ya están todas para que pueda calcular la ruta. Este método devuelve siempre el id de la furgoneta líder.

public int calcularRuta(**int** id): La Furgoneta líder utiliza este método para calcular la ruta a seguir, pero no puede hacerlo hasta que todas las furgonetas se han unido al Convoy.

public void destino(**int** id): La Furgoneta líder utiliza este método cuando llega al destino y, por lo tanto, las furgonetas seguidoras ya pueden abandonar el Convoy. La Furgoneta líder es la última en abandonar el Convoy, por lo que tendrá que esperar a que todas las demás furgonetas hayan abandonado el Convoy antes de salir.

public void seguirLider(**int** id): Una Furgoneta seguidora espera en este método hasta que el convoy llega al destino. Cuando sabe que ha llegado, abandona el Convoy. La última furgoneta en salir (sin contar la líder) avisa a la líder de que ya están todas fuera del Convoy para que ella también pueda salir.

Además, la clase Convoy incluye el método principal (ya está implementado) que crea la flota y el convoy inicial.

Esta obra está sujeta a la licencia Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

Autor: Profesores de la asignatura



Observa que en este ejercicio hay tres condiciones de sincronización:

CS-FLíder 1: La furgoneta líder no puede calcular la ruta hasta que todas las demás furgonetas se han unido al convoy.

CS-FSeguidora: Una furgoneta seguidora no puede abandonar el convoy hasta que el líder no ha indicado que se ha llegado al destino.

CS-FLíder 2: Una vez terminado el viaje, la furgoneta líder no puede abandonar el convoy, hasta que lo han hecho el resto de furgonetas.

Se pide realizar dos implementaciones de la clase Convoy utilizando, una de tipo 1 y otra de tipo 2.

Tipo 1: Semáforos binarios (5 puntos)

Tipo 2: Monitores (5 puntos) o Locks (5 puntos)

La ejecución del programa puede producir, entre otras, la siguiente ejecución:

**** Furgoneta 6 líder del convoy ****

Furgoneta 0 seguidora

Furgoneta 1 seguidora

Furgoneta 7 seguidora

Furgoneta 3 seguidora

Furgoneta 8 seguidora

Furgoneta 9 seguidora

Furgoneta 4 seguidora

Furgoneta 5 seguidora

Furgoneta 2 seguidora

**** Furgoneta 6 líder: ruta calculada, nos ponemos en marcha ****

**** Furgoneta 6 líder: hemos llegado al destino ****

Furgoneta 0 abandona el convoy

Furgoneta 1 abandona el convoy

Furgoneta 7 abandona el convoy

Furgoneta 3 abandona el convoy

Furgoneta 8 abandona el convoy

Furgoneta 9 abandona el convoy

Furgoneta 4 abandona el convoy

Furgoneta 5 abandona el convoy

Furgoneta 2 abandona el convoy

**** Furgoneta 6 líder: abandona el convoy ****