

ST0257 – Sistemas operativos
Proyecto 2: Gestión de Tráfico en una Ciudad de Robots

MBA, I.S. José Luis Montoya Pareja
Departamento de Informática y Sistemas
Universidad EAFIT
Medellín, Colombia, Suramérica

Resumen

Este proyecto implica el desarrollo de un sistema de gestión de tráfico dentro de una ciudad simulada operada por robots, utilizando la biblioteca KarelJRobot en Java. El objetivo es asegurar que múltiples robots puedan navegar en una cuadrícula de la ciudad siguiendo las reglas de tráfico y evitando colisiones, especialmente en intersecciones y en un camino bidireccional de un solo carril hacia una ubicación remota. Los estudiantes aplicarán conceptos de concurrencia, como locks, variables de condición, semáforos y deadlocks, para gestionar las interacciones entre robots. Este proyecto desafía a los estudiantes a diseñar, implementar y probar un sistema de control de tráfico robusto, demostrando su comprensión de conceptos clave de sistemas operativos.

Palabras clave

Concurrencia, Gestión de Tráfico, KarelJRobot, Locks, Variables de Condición, Semáforos, Deadlocks, Sincronización, Sistemas Operativos.

Introducción

En las ciudades modernas, la gestión del tráfico es esencial para prevenir accidentes y asegurar el movimiento eficiente de vehículos y peatones. Este proyecto simula una versión simplificada de la gestión de tráfico en una ciudad operada por robots, donde múltiples robots autónomos (actuando como vehículos) navegan por rutas predefinidas dentro de una cuadrícula. El desafío principal es implementar un sistema que coordine estos robots, permitiéndoles compartir recursos comunes como intersecciones y caminos estrechos sin colisionar ni causar deadlocks.

KarelJRobot, una herramienta educativa basada en Java, se utilizará para simular este entorno. Los estudiantes aprovecharán las técnicas de concurrencia para sincronizar los movimientos de los robots, gestionar recursos compartidos y prevenir deadlocks en secciones críticas de la ciudad. Este proyecto proporciona una aplicación práctica de los conceptos de sistemas operativos, particularmente en las áreas de sincronización y gestión de recursos.

Objetivo Principal

Diseñar, implementar y probar un sistema de gestión de tráfico en una ciudad operada por robots utilizando KarelJRobot, asegurando una navegación segura y eficiente mediante la aplicación de conceptos de concurrencia.

Objetivos Específicos

1. **Diseñar e Implementar Roles y Rutas de los Robots:** Desarrollar roles de robots distintos, cada uno con tareas específicas y rutas dentro de la ciudad, incluyendo la navegación por intersecciones y un camino bidireccional de un solo carril hacia una ubicación remota.
2. **Implementar Mecanismos de Concurrencia para el Control de Tráfico:** Utilizar locks, semáforos y variables de condición para gestionar el acceso a recursos compartidos (por ejemplo, intersecciones), asegurando que los robots no colisionen ni causen deadlocks.
3. **Prevención y Recuperación de Deadlocks:** Desarrollar estrategias para prevenir deadlocks en secciones críticas e implementar mecanismos de recuperación para resolver deadlocks si ocurren.
4. **Simular y Probar el Sistema de Gestión de Tráfico:** Ejecutar simulaciones para probar el sistema de gestión de tráfico bajo diversas condiciones, asegurando que opere de manera fluida y eficiente sin causar deadlocks ni colisiones.

Roles de los robots, hilos, descripción y viaje remoto

1. **Robot Conductor:**
 - **Tarea:** El rol principal del Robot Conductor es navegar por la cuadrícula de la ciudad, siguiendo una ruta establecida que puede incluir el paso por intersecciones y un camino bidireccional de un solo carril. Debe obedecer las señales de tráfico y esperar a que los caminos estén despejados en las intersecciones.
 - **Responsabilidades:**
 - Recibir la ruta a seguir (a lo Uber, vaya del punto A al punto B).
 - Detenerse en las intersecciones si el camino está ocupado.
 - Esperar la señal para proceder en los semáforos.
 - Navegar por el camino bidireccional de un solo carril sin causar deadlocks.
2. **Central de Control de Semáforos:**
 - **Tarea:** Este hilo (**no es un robot**) es responsable de controlar los semáforos en las intersecciones, asegurando que el tráfico fluya sin problemas y que no haya dos robots que entren en una intersección al mismo tiempo.
 - **Responsabilidades:**
 - Gestionar las señales de los semáforos para controlar el flujo de robots en las intersecciones.
 - Coordinarse con los Robots Conductores para evitar colisiones.

- Cambiar los semáforos según un mecanismo de temporización predefinido o en respuesta a las condiciones del tráfico.

3. Viajes:

- **Tarea:** Asignar las rutas que cada robot debe seguir. Uno (o varios) de los robots puede recibir la misma ruta y la ruta puede ir a una ciudad remota. La ciudad está ubicada en una montaña de roca y como el presupuesto de la ciudad es limitado, solo se ha podido construir un solo carril para subir y bajar de la ciudad. Cada cierta distancia, hay una bahía donde se puede parquear un carro mientras pasa el otro. Como no se pueden chocar, alguno deberá devolverse al punto de partida o a la bahía más cercana.
- **Responsabilidades:**
 - Asignar la ruta a cada robot cuando el robot se inicializa.

Comando a ejecutar

```
java traffic_management [opciones]
```

DESCRIPCIÓN

Simula un sistema de gestión de tráfico en una ciudad de robots basada en una cuadrícula 20x20 utilizando la biblioteca KarelJRobot. El sistema gestiona múltiples robots que navegan a través de intersecciones y un camino bidireccional de un solo carril, aplicando conceptos de concurrencia para asegurar un funcionamiento fluido.

OPCIONES

`--run-simulation`

Ejecuta la simulación de gestión de tráfico con la configuración predeterminada.

`--config-file <archivo>`

Especifica un archivo de configuración para personalizar los roles de los robots, las rutas y las reglas de tráfico.

`--verbose`

Activa la salida detallada de las acciones de los robots, incluyendo los cambios de semáforos y los movimientos de los robots.

`--test-deadlock`

Ejecuta un caso de prueba específico diseñado para inducir un escenario de deadlock, utilizado para evaluar los mecanismos de prevención y recuperación de deadlocks del sistema.

--help
Muestra este mensaje de ayuda.

EJEMPLOS

`traffic_management --run-simulation`

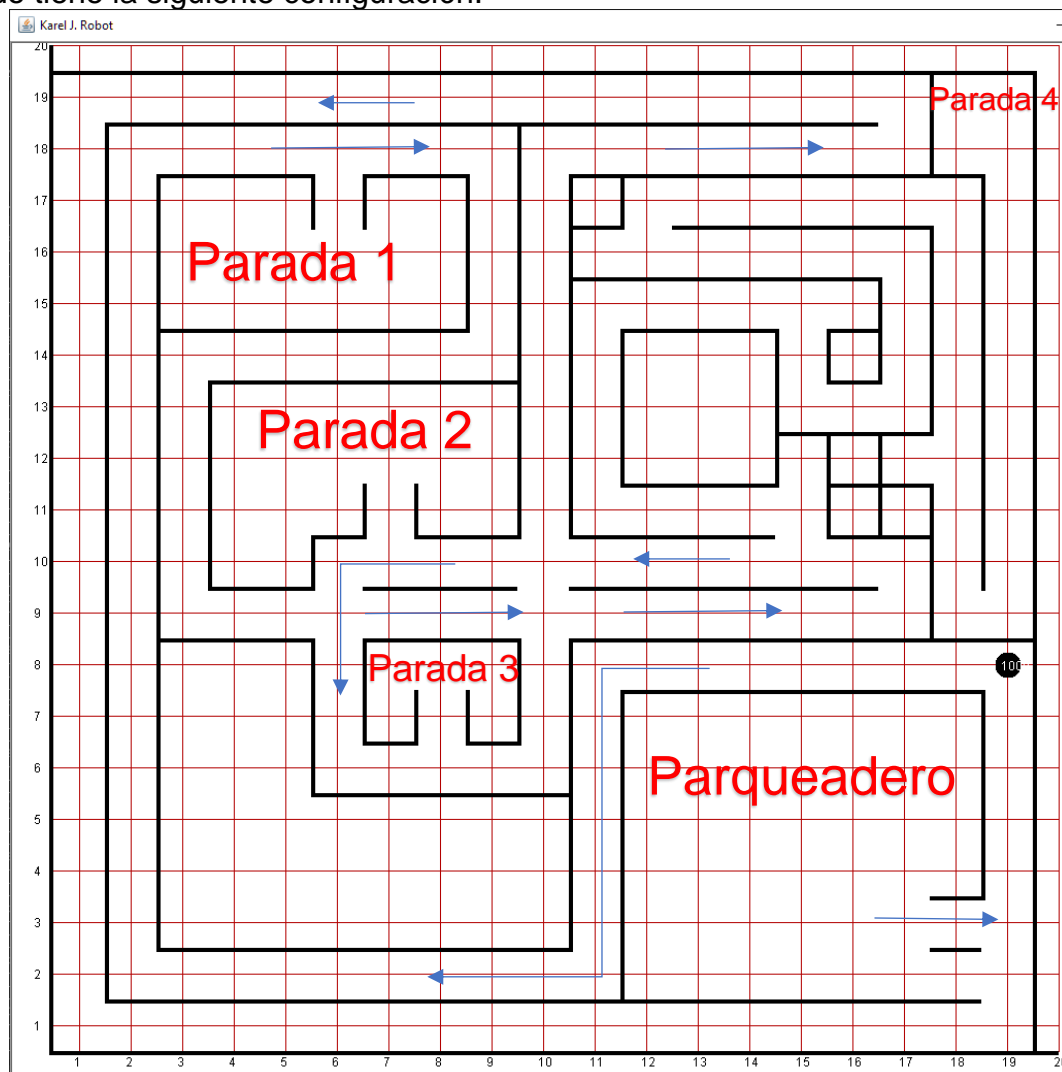
Ejecuta la simulación predeterminada con la configuración estándar de gestión de tráfico.

`traffic_management --config-file custom_config.kwld --verbose`

Ejecuta la simulación con configuraciones personalizadas definidas en el archivo de configuración especificado, con salida detallada.

Consideraciones Generales

1. El mundo tiene la siguiente configuración:



Los robots (la cantidad que ustedes quieran, mayor o igual a 8) arrancan en cualquier posición del parqueadero. Cuando se da la indicación de arrancar la simulación, salen del

parqueadero a recoger un pasajero (un beeper) de la esquina de la avenida 19 con calle 8 donde hay 1000 beepers a transportar. El viaje es aleatorio, es decir, antes de recoger el pasajero, ellos aleatoriamente reciben el viaje desde el punto de recogida hasta la parada elegida aleatoriamente entre la 1 y la 4. El robot viaja con el pasajero hasta su parada, lo deja y regresa a recoger pasajero. Antes de recoger el pasajero, repite el proceso de selección aleatoria de la parada.

La parada 4 es la ciudad remota. Como el camino es de una sola vía, se tienen zonas de descanso donde si se encuentran dos robots de frente pueden devolverse hasta que pase alguno.

Cuando no hayan más pasajeros, los robots regresan al parqueadero y se apagan.

2. El desarrollo de la práctica puede ser individual o en equipos de máximo tres personas.
3. La entrega de la práctica se realizará adjuntando el código fuente y el informe por el buzón recepción de trabajos de Eafit Interactiva (cualquier otro medio no será admitido).
4. Se debe informar al profesor a más tardar el **domingo 8 de septiembre** las 6:00 p.m. los integrantes del equipo.
5. El informe final es una presentación que deberá contener una breve descripción de cómo funciona el programa, que dificultades debieron superar para el desarrollo de la práctica y posibles mejoras que consideran, se puede hacer a la misma.
6. La práctica se debe realizar en Java.
7. Cada semana los jueves, se sacará un espacio de 10 a 15 minutos al inicio de la clase para hablar de la práctica y resolver dudas.
8. Criterios de evaluación (ver Anexo 1)

Fecha de entrega

Jueves 12 de septiembre en clase a través de Eafit Interactiva.

Sustentación

Jueves 12 de septiembre en clase. El mecanismo de sustentación es el siguiente:

1. Cada equipo muestra su desarrollo ejecutando en vivo.
2. Debe mostrar cómo solucionó cada uno de los retos y cómo lo relacionaron con conceptos vistos en clase.
3. Se realizarán preguntas por parte del docente para validar el entendimiento individual de los conceptos aplicados. Esto significa que, aunque el trabajo es en equipo, la nota del trabajo puede ser diferente para cada uno de acuerdo con la calidad de las respuestas.

Referencias

- Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. C. (2018). *Operating Systems: Three Easy Pieces*. Arpaci-Dusseau Books.
- Bergin, J. (2001). *Karel J Robot: A Gentle Introduction to the Art of Object-Oriented Programming in Java*. College of Computing and Information Sciences, New York Institute of Technology.

- Silberschatz, A., Galvin, P. B., & Gagne, G. (2020). *Operating System Concepts* (10th ed.). Wiley.
- Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson.

Nombre de la asignatura: Sistemas Operativos

Competencia a la que aporta la asignatura: Conocer el sistema operativo del computador para un mejor desarrollo, diseño y ejecución de las aplicaciones y aplicar nuevas soluciones

Resultado de asignatura evaluado: Solución de problemas de concurrencia de procesos.

Evento evaluativo: Proyecto 2

Porcentaje del evento evaluativo: 30%

Criterios (que tributen al RA de asignatura)	Cumple con altos estándares (4.5-5)	Cumple a satisfacción (4-4.4)	Cumple parcialmente (3.5-3.9)	Incumple parcialmente (2.5-3.4)	Incumple totalmente (0-2.4)	Peso asignado al criterio sobre la calificación.
Diseño e Implementación de Roles de Robots: Evalúa qué tan bien están diseñados e implementados los diferentes roles de robots, incluyendo su capacidad para seguir rutas y cumplir con las reglas de tráfico.	Entiende completamente la necesidad y plantea varias opciones de solución. Utiliza conceptos adecuadamente para la solución.	Entiende completamente la necesidad y plantea una opción de solución.	Omitió un elemento clave para el entendimiento de la necesidad	Omitió varios elementos para el entendimiento de la necesidad.	Demuestra poco o nulo entendimiento del problema.	30%
Uso Correcto de Mecanismos de Concurrencia: Evalúa la aplicación correcta de	El diseño tiene en cuenta los conceptos vistos en clase y argumenta la elección de su solución. La solución elegida es la óptima para el problema.	El diseño tiene en cuenta los conceptos vistos en clase y argumenta la	Aunque se tuvieron en cuenta los conceptos vistos en clase, no hubo argumentación correcta en la	No se tuvieron en cuenta los conceptos vistos en clase.	Demuestra poco o nulo entendimiento de patrones de paralelismo al momento de	30%

Criterios (que tributen al RA de asignatura)	Cumple con altos estándares (4.5-5)	Cumple a satisfacción (4-4.4)	Cumple parcialmente (3.5-3.9)	Incumple parcialmente (2.5-3.4)	Incumple totalmente (0-2.4)	Peso asignado al criterio sobre la calificación.
mecanismos de concurrencia, como locks, semáforos y variables de condición, para gestionar las interacciones de los robots y los recursos compartidos.		elección de su solución.	elección de la solución.		explicar la solución.	
Prevención y Recuperación de Deadlocks Mide la efectividad de las estrategias para prevenir y recuperar deadlocks en secciones críticas como las intersecciones y el camino de un solo carril.	El programa funciona correctamente.	La solución elegida no es la óptima pero el programa funciona correctamente	El programa con los conceptos vistos en clase no funciona correctamente.	El programa no tiene en cuenta ningún concepto visto en clase y la solución funciona correcta o parcialmente.	Se entrega la solución parcialmente o no se entrega ninguna solución.	20%
Simulación y Pruebas						

Criterios (que tributen al RA de asignatura)	Cumple con altos estándares (4.5-5)	Cumple a satisfacción (4-4.4)	Cumple parcialmente (3.5-3.9)	Incumple parcialmente (2.5-3.4)	Incumple totalmente (0-2.4)	Peso asignado al criterio sobre la calificación.
Incluye la calidad de las simulaciones ejecutadas, los escenarios probados y el rendimiento general del sistema de gestión de tráfico.						
Informe y Documentación Refleja la calidad y la completitud del informe del proyecto y cualquier documentación adjunta, incluyendo la explicación de las decisiones de diseño y los resultados de las pruebas.						5%