

Proyecto 1 Sistemas Operativos

Objetivos

Aplicar los conceptos de programación paralela, concurrencia y sincronización en el diseño y ejecución de un programa sencillo que implemente múltiples hilos de procesamiento.

- Comprender el uso de hilos y estructuras básicas de sincronización para programas paralelos.
- Diseñar soluciones paralelas que consideren dependencias funcionales, regiones críticas, y necesidades de sincronización

Descripción del problema

Diseñe una aplicación que permita crear una simulación de tráfico vehicular. Su simulación deberá tener una interfaz que le permita al usuario crear una red de carreteras usando grafos. Cada nodo del grafo es una intersección, y cada arista una vía, asuma que las vías son siempre de doble sentido. Cuando el usuario cree un nuevo nodo, deberá poder crear aristas que conecten con los otros nodos, el usuario deberá ser capaz de asignar a cada arista un valor d que represente la distancia de la arista en km, de forma que si el usuario ingresa el valor $d = 40$ esa arista mide 40 km. Asuma que el grafo no es fuertemente conexo. También es importante que el usuario pueda configurar en cada nodo un valor α que significa la tasa de creación del nodo dada en autos por segundo, este valor puede ser un número flotante, de forma que si el usuario ingresa el valor $\alpha = 0.5$ ese nodo creará un auto nuevo cada 2 segundos (0.5 autos por segundo).

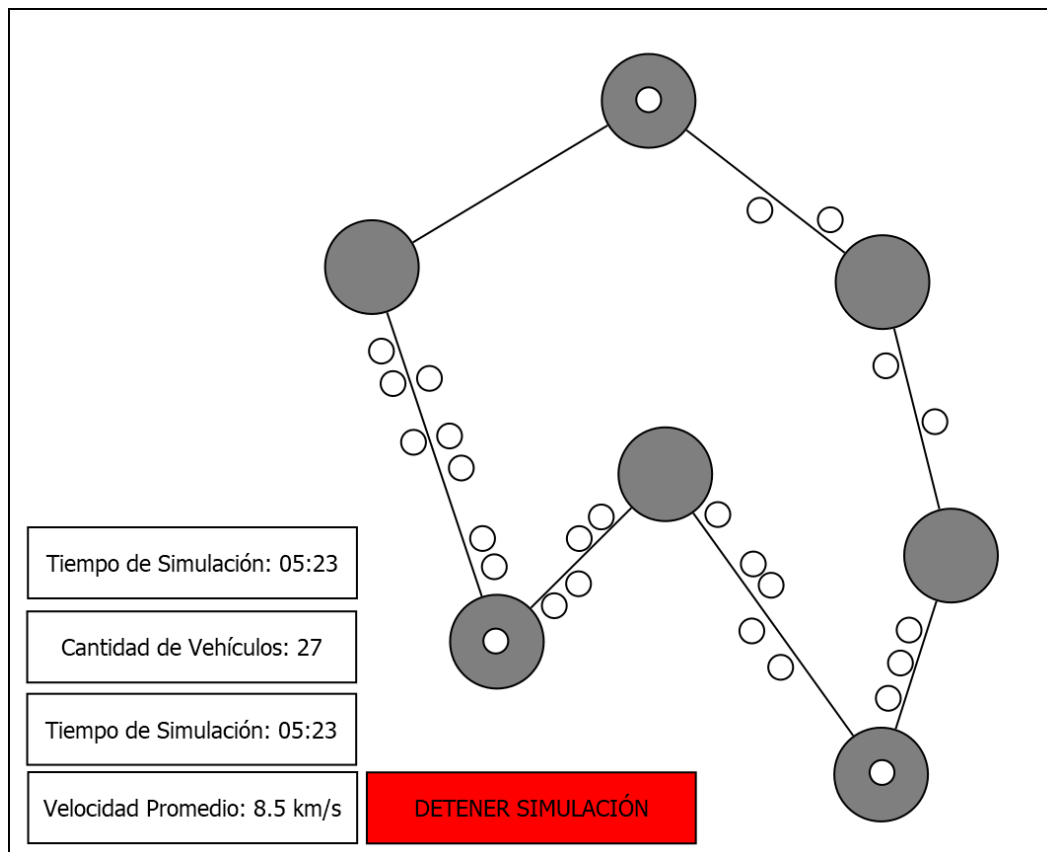
Una vez esté completo el grafo, el usuario deberá poder "INICIAR LA SIMULACIÓN". Al hacer esto cada nodo deberá producir un vehículo según su valor α . Cada vehículo, al momento de creación, deberá seleccionar un nodo del grafo, debe ser un nodo distinto al que lo creó, y viajar hacia ese destino usando la ruta más corta (investigue el algoritmo de Dijkstra). Asuma que cada vehículo tiene una velocidad de 10km/s. Cuando un vehículo llega a un nodo, tarda 2 segundos en cruzarlo. Si un vehículo llega a su nodo destino, termina su ejecución. Sólo puede haber un vehículo por nodo, si un vehículo llega a un nodo ocupado deberá esperar hasta que esté desocupado.

Su programa deberá representar cada vehículo en su posición actual en la red y deberá indicar los siguientes datos:

- Tiempo de simulación
- Cantidad de vehículos activos
- Velocidad promedio (considere que un vehículo en movimiento tiene velocidad de 10km/s, pero un vehículo detenido tiene velocidad de 0km/s)

Además deberá contar con un botón de **Detener Simulación** que destruya todos los vehículos y permita al usuario volver a editar el grafo.

Ejemplo de la visualización



Forma de Entrega

Deberá entregar un informe en formato PDF que incluya un enlace al código fuente de su proyecto. Su proyecto deberá tener un archivo `install.sh` que permita compilar o instalar todas las dependencias de su proyecto en la última versión de Fedora Linux y un `run.sh` que corra el proyecto. Si su proyecto no compila o no corre, **no se calificará**.

El informe deberá tener la siguiente información:

- Portada
- Introducción (10%)
 - Explicación del problema
 - Introducción a los conceptos clave (hilos, sincronización, mutex)
- Desarrollo (30%)
 - Explicación de la solución
 - Detalle de las bibliotecas y funciones usadas para obtener el comportamiento paralelo
 - Descripción de los elementos del programa que corren de forma paralela
 - Descripción de todas las regiones críticas y estrategias usadas para evitar conflictos
- Enlace de descarga del proyecto funcionando (40%)
- Conclusiones (20%)
 - Al menos 3 conclusiones de aprendizajes obtenidos a partir del proyecto
- Referencias bibliográficas
 - Todo dato obtenido de una fuente externa deberá ser citado apropiadamente, de lo contrario se anulará su entrega.

Utilice esta plantilla para dar formato a su proyecto:

