

# Proyecto 3: Aventuras en *Threadville*

## Lunes 1ro. de Junio

### I. DESCRIPCIÓN

En este proyecto se considerará un manejo no trivial de *threads* y concurrencia. Toda la programación debe realizarse en C sobre Linux, utilizando la biblioteca *Pthreads* para el manejo de hilos de ejecución.

### II. *Threadville*

Esta es una pequeña pero próspera ciudad que ha crecido de forma muy ordenada. Una moderna autopista de cuatro carriles (2 por sentido) cruza la ciudad de Este a Oeste, partiendo a la ciudad en dos barrios llamados *Uptown*, al norte, y *Underground* en el sur. En ambos extremos de esta autopista hay rotondas de un carril las cuales giran en sentido contrario a las manecillas del reloj. Estas rotondas son las únicas entradas o salidas a la autopista. Alrededor de todo *Threadville* hay una carretera de circunvalación de un solo carril donde los autos circulan en el sentido de las manecillas del reloj. Cada mitad de la ciudad es cruzada horizontalmente por una calle de un solo carril. En *Uptown* esta calle circula de Este a Oeste, mientras que en *Underground* va de Oeste a Este.

Hay 5 avenidas de 2 carriles y tránsito en ambos sentidos que cruzan *Threadville* de Norte a Sur. Para pasar sobre la autopista se colocaron 5 puentes cuyos nombres, viniendo de Oeste a Este, son respectivamente **Larry**, **Curly**, **Moe**, **Shemp** y **Joe**. Todos ellos tienen un solo carril de ancho pero permiten el tránsito en ambas direcciones. Cada uno se administra en forma diferente:

- **Larry** y **Joe** siempre tienen un par de policías en sus extremos que dejan pasar  $K$  automóviles en un sentido, y luego  $K$  automóviles en el otro sentido (si hay menos de  $K$  autos en un sentido dejan pasar a los de la otra punta). Los autobuses cuentan por 2 automóviles.
- **Curly** y **Shemp** tienen semáforos (rojo y verde) en cada punta que dejan pasar carros alternadamente en cada sentido durante  $M$  segundos y  $N$  segundos ( $M$  y  $N$  podrían ser diferentes).
- **Moe**, que está en el centro de la autopista, se administra con “la ley de la jungla” donde los autos se meten al puente en cuanto pueden, siendo la única regla que un auto no puede entrar si ya hay otro en sentido contrario dentro del puente.

Como podemos notar hay 12 cuadras en *Uptown* y 12 cuadras en *Underground*. Todas son cuadradas, con bordes que acomodan aproximadamente 6 automóviles uno tras otro. El ancho de las calles de un solo carril es de un sexto de cuadra, y el ancho de las de dos carriles es de un tercio de cuadra. El radio de las rotondas es poco menos de lo que medirían 4 automóviles uno tras otro. Los puentes podrían permitir hasta 6 vehículos uno tras otro.

En el mapa adjunto al final de este documento las cuadras se etiquetan como  $A, B, C, \dots, X$ . A su vez, cada cuadra tiene 6 u 8 paradas marcadas. El extremo Oeste de la rotonda Oeste es conocido como  $Y$ , y el extremo Este de la rotonda Este es conocido como  $Z$ .

### III. TRANSPORTE PÚBLICO

El servicio de buses es muy bueno en *Threadville*. Todos los buses tienen un tamaño equivalente a 2 autos corrientes. Se mueven a una velocidad un poco más lenta que el automóvil promedio y hacen paradas de 5 segundos en puntos establecidos de su recorrido.

Cada uno tiene un color distintivo y una ruta asignada:

- **Bus Rojo:** conocido como la “periférica grande”. Hace paradas en  $A_1, D_1, F_2, L_4, Z$  (dándole vuelta a la rotonda),  $R_2, X_5, U_5, S_6, M_1, Y$  (dándole vuelta a la rotonda),  $G_5$ , y  $A_1$ .
- **Bus Verde:** conocido como la “periférica norte”. Sus paradas son:  $E_2, L_3, Z$  (dándole vuelta a la rotonda), toma la autopista hacia el Oeste,  $Y$  (dándole vuelta a la rotonda),  $G_6, B_1$  y  $E_2$ .
- **Bus Azul:** conocido como la “periférica sur”. Hace paradas en  $R_3, W_5, T_6, M_6, Y$  (dándole vuelta a la rotonda), toma la autopista hacia el Este,  $Z$  (dándole vuelta a la rotonda), y  $R_3$ .
- **Bus Blanco:**  $I_4, O_3, T_5, N_1, H_6, C_1$  e  $I_4$ .
- **Bus Gris:**  $U_5, O_1, I_6, D_1, J_4, P_3$  y  $U_5$ .
- **Bus Negro:**  $J_6, E_1, K_4, Q_3, V_5, P_1$  y  $J_6$ .
- **Bus Rosa:**  $F_5, A_6, A_1, F_2$  y  $F_5$ .
- **Bus Celeste:**  $S_6, S_1, X_2, X_5$ , y  $S_6$ .
- **Bus Naranja:** bus turístico conocido como la “lombriz”. Hace paradas en  $M_3, O_6, C_8, C_3, O_3, Q_6, E_8, E_3, Q_3, X_4, X_7, L_6, J_3, V_4, V_7, J_6, H_3, T_4, T_7, H_6, A_8, A_3$  y  $M_3$ .

La primer parada indicada en cada lista es el punto de arranque de cada bus. En la parte de encima de cada bus siempre se leerá su siguiente parada.

### IV. REPARACIONES

Con su espíritu progresista, *Threadville* le da mantenimiento preventivo a su ciudad. Hay reparaciones que ocurren cada cierta cantidad de segundos (siguiendo una distribución exponencial con una media de 40 segundos). Aletoriamente, cualquiera de los puntos identificados en el mapa podría estar cerrado durante 5, 10 o 15 segundos. Ningún vehículo podrá cruzar por ahí. El despliegue mostrará esa situación.

## V. AUTOMÓVILES

Aparte de los autobuses, hay automóviles livianos en *Threadville*. Físicamente miden “medio autobús” de largo. Los autos siempre aparecen en el punto *Y* - en un instante donde haya espacio en ese lugar - y tendrán que hacer cierto número de viajes antes de “morir”, siendo siempre el último destino final el punto *Z*.

Un viaje se define como ir de un punto a otro de la ciudad. Al llegar a un destino intermedio, el automóvil se detiene por 3 segundos antes de seguir a su próxima visita. Estos viajes son seleccionados en forma aleatoria, sin embargo hay cierto sesgo a visitar las rotondas *Y* y *Z*. Todos los carros<sup>1</sup> mostrarán el número de viajes que aún le quedan y su destino inmediato. Al planificar su ruta un auto debería escoger una ruta óptima (Floyd - Dijkstra) la cual debiera darle preferencia a usar la autopista.

Un automóvil se caracteriza por:

- Número de viajes.
- Destinos (el orden de estas visitas se debe respetar)
- Velocidad y Color.

## VI. AMBULANCIAS

Las ambulancias son vehículos especiales que se mueven a gran velocidad y siempre tienen prioridad sobre los demás automóviles o buses. Las ambulancias “nacen” en *Y* y “mueren” en *Z*. Solamente hacen 2 paradas intermedias (escogidas aleatoriamente) donde se quedarán 8 segundos en cada una.

## VII. VELOCIDADES

La comunidad de *Threadville* es tan organizada que con solo ver el color de un vehículo sabemos cuál es su velocidad máxima. La Tabla 1 muestra las velocidades máximas de los automóviles, autobuses y ambulancias expresadas en la cantidad de segundos que les tomaría recorrer desde  $A_1$  hasta  $F_2$  si no hubiera ningún obstáculo. Estas velocidades su duplican cuando se viaja por la autopista. Siempre que sea posible un vehículo irá a su máxima velocidad. Obviamente, nadie puede pasarle por encima a otro.

**Trabajo extra opcional 1:** en la autopista o en cualquier calle de 2 carriles, los vehículos pueden rebasar a otro de velocidad inferior. Por supuesto, sin accidentes...

## VIII. INTERSECCIONES

Los conductores en *Threadville* son muy civilizados y buenos conductores. Cruzan una intersección... cuando se puede. En cualquier intersección las ambulancias tienen prioridad sobre cualquier otro tipo de vehículo.

## IX. SIMULACIÓN

Deben escribir una simulación gráfica<sup>2</sup> completa del problema descrito. Todas las entidades del problema (automóviles, policías,

<sup>1</sup>en este documento, las palabras carro, auto o automóvil son todas sinónimas.

<sup>2</sup>Recuerden que toda la programación debe realizarse en C o en ensamblador corriendo sobre Linux. Pueden usar cualquier paquete gráfico compatible con estos requisitos. No se permite el uso de ningún otro lenguaje.

Tipo	Segundos desde $A_1$ hasta $F_2$
Rojo	1
Azul	2
Verde	3
Negro	4
Blanco	5
Amarillo	6
Ambulancia	0.5
Bus Naranja	7
Buses Rosa y Celeste	3
Buses Blanco, Negro y Gris	4
Buses Rojo, Verde y Azul	5

Tabla 1: Velocidades en *Threadville*

ambulancias, buses, semáforos, etc.) serán modeladas por *threads* concurrentes que actualizarán los puntos apropiados de las estructuras de datos que describan la situación actual, la cual será constantemente desplegada en pantalla. Habrá una manera interactiva de alterar todos los parámetros de la simulación.

## X. INTERACCIÓN

Al arrancar la simulación, sólo estará circulando el transporte público. Con una interfaz gráfica apropiada se pueden desactivar o reactivar cualquiera de los buses. Al desactivar un bus, este desaparece de *Threadville*, al reactivarlo este aparece de nuevo en su parada inicial. Nótese que es posible dejar a *Threadville* sin buses. Con una interfaz apropiada (e.g., un botón) se genera un automóvil. Similarmente, habrá otro botón para crear ambulancias. Estos vehículos aparecen en *Y* y empiezan sus recorridos que incluyen paradas generadas aleatoriamente. El “color/velocidad” del automóvil es generado aleatoriamente con una distribución uniforme. Habrá otro par de botones para generar automóviles y ambulancias, pero donde el usuario indicará manualmente los datos de tipo de vehículo, y destinos.

## XI. DISEÑO GENERAL

En *Threadville* hay recursos que deben ser compartidos armoniosamente por todos los vehículos. A fin de cuentas, su trabajo es escribir el Sistema Operativo de esta ciudad, el cual fundamentalmente proporciona **protección**, dejando la **administración** a los procesos. Estos procesos debieran ser escritos de la manera más genérica posible: un automóvil, una ambulancia, un bus... todos son equivalentes (e.g., ¿basta con un único tipo de *thread* que sabe manejar un vehículo arbitrario?).

## XII. REQUISITOS INDISPENSABLES

La ausencia de uno solo de los siguientes requisitos vuelve al proyecto “no revisable” y recibe un 0 de calificación inmediata:

- La colaboración entre grupos se considera fraude académico.

- Todo el código debe estar escrito en C o ensamblador (no se permite ningún otro lenguaje).
- El proyecto debe compilar y ejecutar en Linux. Todo debe estar **integrado**, explicaciones del tipo “*todo está bien pero no pudimos pegarlo*”<sup>3</sup> provocan la cancelación automática de la revisión.
- Si no hay interfaz gráfica el proyecto no será revisado.
- La presentación debe ser de mucha calidad.
- No debe dar “Segmentation Fault” bajo ninguna circunstancia.
- Hacer la demostración en una máquina que levante Linux de manera real (puede ser dual), es decir no usar máquinas virtuales.

### XIII. FECHA DE ENTREGA

Demostraciones en clase el **Lunes 1ro. de Junio del 2020**. También mandar por correo electrónico a `torresrojas.cursos@gmail.com` antes de la clase. Coloque todo lo necesario para compilar, documentar y ejecutar su proyecto en un directorio cuyo nombre esté formado por los apellidos de los miembros de cada grupo separados por guiones. Compacte este directorio en la forma de un `.tgz` llamado igual que el directorio (e.g., `torres-venegas-castro-smith.tgz`) y envíelo por correo. Identifique claramente su correo con el siguiente subject:

[SOA] Proyecto Programado 3 - Apellido 1 -  
Apellido 2 - Apellido 3

Buena Suerte.

---

<sup>3</sup>esto incluye los supuestos casos cuando alguien del grupo de trabajo no hizo su parte – el profesor no está interesado en sus problemas de organización.

