



Universidad Nacional Autónoma de México

Facultad de Ingeniería



Sistemas Operativos

Tarea 01: Ejercicios de Sincronización.

Acosta Jacinto Alan, "320101179"

Rubio Carmona Jose Angel, "320118937"

Sexto semestre.

Fecha de entrega : 27 de Marzo de 2025

Profesora: Gunnar Eyal Wolf Iszaevich

Semestre 2025-2

Ciudad de México, 27 de marzo de 2025.

- **Problemática**

Para esta tarea elegimos el problema del **Cruce del río**, donde como planteamiento nos dice que existe un lugar donde asisten hackers (desarrolladores de linux) y serfs (desarrolladores de microsoft), el problema es que para asistir al evento deben cruzar un río y donde la única opción es cruzar sobre una balsa.

En la balsa pueden abordar únicamente 4 personas, ni más ni menos, pero el problema real es al abordar, ya que como se mencionó solo pueden ser 4 personas, sin embargo se nos menciona que si llegase a sube un serf y tres hackers, estos se pelearon por alguna razón y volcaron la balsa, esto ocurriría igual en un caso inverso, para solucionar esto solo pueden abordar en pares o en cuartetos completos, es decir 2 serfs y 2 hackers, 4 hackers o 4 serfs.

- **Funcionamiento del código**

El código para el problema antes planteado fue realizado en Python, donde implementamos funciones de sincronización por semáforos.

Semáforo **mutex**:

- Se utiliza un semáforo **mutex** con valor inicial **1** para proteger el acceso a los contadores globales de **hackers** y **serfs**.
- Esto evita condiciones de carrera, ya que varias personas pueden llegar al mismo tiempo y deben aumentar el contador de su tipo. El semáforo asegura que solo un hilo pueda acceder y modificar estas variables a la vez.

Semáforo **balsa**:

- El semáforo **balsa** se utiliza para permitir que las personas suban a la balsa. El valor inicial es **0**, lo que significa que no se permite el cruce hasta que haya suficientes personas (se libera el semáforo).
- Una vez que hay suficiente gente (2 hackers y 2 serfs o 4 del mismo tipo), se libera el semáforo **balsa** un número adecuado de veces (4 para permitir que crucen las 4 personas).
- Luego, los hilos intentan adquirir el semáforo **balsa** para "subir a la balsa", y se asegura que todos los miembros del grupo suban juntos antes de cruzar.

Control de condiciones de cruce:

- El código verifica si se tienen suficientes hackers y serfs para formar un grupo válido:
 - Si hay **2 hackers y 2 serfs**, pueden cruzar.
 - Si hay **4 hackers o 4 serfs**, también pueden cruzar.
- Después de liberar el semáforo **balsa**, las personas "suben" y el número de hackers o serfs se ajusta para reflejar que ya han cruzado.

Creación de hilos:

- Cada persona (hacker o serf) es representada por un hilo. Los hilos se crean de manera concurrente, lo que simula la llegada aleatoria de las personas al río.
- Cada hilo ejecuta la función `persona_llega`, donde se realiza la verificación de los contadores y la lógica para determinar si las personas pueden cruzar.

Control de tiempos:

- Se utiliza `time.sleep(random.uniform(1, 1))` para simular que las personas llegan al río en tiempos aleatorios, lo que ayuda a mostrar la ejecución concurrente de los hilos.

- *¿Qué ejecuta el código?*

El código nos presenta la llegada de los serfs y hackers, donde estos llegan de forma aleatoria, estos tendrán un número que los identifica, esto para reconocer cuando llegue el último, si se reúnen los suficientes para subir a la balsa estos abordarán ,pero ¿qué pasa si llega alguna persona que no cubre el requisito con las otras 3 personas?, lo que pasa es que este tendrá que esperar hasta que pueda abordar cumpliendo otro requisito,es decir, si llegase algún hacker y posteriormente 3 serfs, estamos conscientes de que no pueden viajar juntos, así que deberán esperar a un hacker mas o un serf, en caso de que quien llegue sea otro serfs, el hacker tendrá que esperar hasta que lleguen 3 hackers o otro hacker y 2 serf. En seguida de cada viaje la balsa volverá de inmediato, en el código justo en `for i in range(1,21)`, tendremos seleccionadas a la cantidad de personas que llegan, sin por alguna casualidad llegaran a sobrar, el programa marcará fin del código con una leyenda "\nLos viajes finalizaron. Lamentamos aquellos que ya no pudieron viajar.", esto puede ocurrir si son un número no múltiplo de 4, o si de casualidad los últimos 4 , son 3 y 1, no podrán viajar de la misma forma.