Universidad Autónoma de México
Facultad de Ingeniería

Tarea 1

“El cruce del río”

Sistemas Operativos

Hernández Saldívar Héctor Saúl
319276017

Ing. Gunnar Eyal Wolf Iszaevich

Grupo: 6

Semestre 2025-1

Fecha de entrega:
22 de octubre de 2024

Problema: El cruce del Río

Planteamiento:

- Para llegar a un encuentro de desarrolladores de sistemas operativos, hace falta cruzar un río en balsa.
- Los desarrolladores podrían pelearse entre sí, hay que cuidar que vayan con un balance adecuado

Reglas:

- En la balsa caben cuatro (y sólo cuatro) personas.
    - La balsa es demasiado ligera, y con menos de cuatro puede volcar.
- Al encuentro están invitados hackers (desarrolladores de Linux) y serfs (desarrolladores de Microsoft).
    - Para evitar peleas, debe mantenerse un buen balance: No debes permitir que aborden tres hackers y un serf, o tres serfs y un hacker. Pueden subir cuatro del mismo bando, o dos y dos.
- Hay sólo una balsa.
- No se preocupen por devolver la balsa (está programada para volver sola)

Lenguaje y entorno de desarrollo:

Para lograr una solución a este problema, se decidió emplear el lenguaje de programación Python en su versión 3.12. Por ende, es necesario que el equipo de cómputo donde se vaya a correr el programa tenga instalada una versión 3.x de este mismo lenguaje, si no se tiene instalado Python no se podrá ejecutar el programa.

Para la programación se decidió utilizar el editor de texto Sublime text, pues este nos permite escribir código en casi cualquier formato, uno de los formatos compatibles con este editor son los archivos .py, siendo estos los creados con Python. Para poder realizar la ejecución es preferible que se utilice la terminal de nuestro equipo, para lograr que se

ejecute nuestro archivo podemos escribir el comando *python archivo.py*, si tenemos tanto Python 2 como Python 3 a la hora de ejecutar nuestro archivo debemos colocar *python3* (es importante que con la terminal estemos dentro de la misma ubicación donde tengamos el archivo .py, porque nos puede marcar que no existe dicho archivo).

Estrategia de sincronización empleada:

Mecanismos:

- Semáforos: Estos tienen la función en el programa de hacer que no se acumulen más de 4 serfs o hackers en espera para poder abordar la balsa. Cada que una persona se sube a la balsa se decrementa su valor, si el viaje aun termina se bloquea la llegada de nuevos serfs o hackers, garantizando que no se permitan más llegadas hasta que un viaje sea completado.
- Mutex: El mutex se encarga para que las variables con valores compartidos serfs, hackers y total_abordo no tengan accesos simultáneos, sin el mutex una variedad de threads puede realizar modificaciones a los contadores de sitios valores causando que se sobrepase el límite de las personas permitidas.
- Eventos: El evento nos indica cuando un viaje ha zarpado con éxito o se ha cancelado. Los threads revisan si el evento está activado para saber si deben seguir esperando o si deben detenerse, esto hace que no se ejecuten más allá del viaje actual para evitar la acumulacion de hilos adicionales y así se garantiza que solo haya 4 personas por viaje.

Patrón utilizado:

Para este problema se puede decir que se utilizó un patrón similar al "Productor-Consumidor", este patrón se basa en el funcionamiento de un almacén donde tenemos productores, un almacén de capacidad limitada donde colocamos los

recursos,condiciones para realizar el consumo de los recursos y reglas de espera, a continuación relacionaremos esto con nuestro programa:

- Productores: En este caso hablamos de los hackers y los serfs, pues estos intentan subirse a la balsa produciendo así una ocupación en esta misma. Esta ocupación se intenta mediante el semáforo de bloqueo.
- Almacén: La balsa tomaría este papel ya que tiene una capacidad limitada, ya que solo se permiten cuatro personas dentro de esta misma.
- Condiciones de consumo: El consumo en esta casa es si el viaje se realiza, para ello se deben tener exactamente 4 personas a bordo, pueden ser 4 de un mismo tipo o 2 serfs y 2 hackers.
- Reglas de espera:
    - Si ya hay 4 personas esperando los serfs y hackers que llegan son bloqueados por el semáforo.
    - Si no se cumplen las reglas de balance de la balsa, el viaje se cancela y todas las personas se bajan.

Refinamientos:

Para esta implementación no utilizamos refinamientos explícitos, pues haciendo el uso de los semáforos se controla la cantidad máxima de desarrolladores permitidos y el mufex nos garantiza un bloqueo a las variables compartidas cada que un serf o hacker llega, esto nos permite exclusión mutua y sincronización.

Dudas:

Una duda que tengo es que a la hora de cancelar el programa de manera manual (en mi caso Ctrl+C)no parece que el programa se deje de ejecutar y parece que sigue corriendo a menos que cierre la terminal. ¿Por qué pasa esto?