Cómputo Concurrente 2024-2
Mision Secundaria 3
Candado Filtro

Natalia Abigail Pérez Romero
Jonathan Bautista Parra
Valeria Reyes Tapia

7 de marzo de 2024

Plantea un pequeño problema donde se solucione usando este tipo de candado. (Solo describirlo y el como y donde se usaría)

Un problema donde se puede aplicar el uso del candado FiltroLock es un sistema de reserva de asientos para un concierto. Supongamos es el concierto de Latin Mafia el cual cuenta con un número limitado de asientos disponibles y múltiples usuarios que intentan reservar asientos simultáneamente.

El problema consiste en garantizar que dos usuarios no reserven el mismo asiento al mismo tiempo, lo que podría resultar en asientos duplicados o errores de reserva.

Para resolver este problema utilizando el candado FiltroLock, podríamos diseñar un sistema donde cada asiento esté representado por un recurso compartido y el candado se use para garantizar la exclusión mutua al acceder a esos recursos compartidos (los asientos).

El proceso sería algo así:

1. Cada hilo de usuario que intenta reservar un asiento adquiere el candado FiltroLock antes de intentar reservar un asiento.
2. Una vez que el hilo ha adquirido el candado, verifica si el asiento que desea reservar está disponible.
3. Si el asiento está disponible, lo reserva y luego libera el candado.
4. Si el asiento ya está reservado por otro hilo, el hilo actual espera hasta que el asiento esté disponible nuevamente.

Este proceso garantiza que solo un hilo pueda reservar un asiento a la vez, evitando así problemas de concurrencia donde dos hilos podrían reservar el mismo asiento simultáneamente.

Entonces, en la implementación del sistema de reserva de asientos, usaríamos instancias de la clase FiltroLock para controlar el acceso concurrente a la reserva de asientos, asegurando que no se produzcan reservas duplicadas o conflictos de reserva.

También añadimos el Diagrama UML de la Mision