

Práctica 4: Candados clásicos

Cómputo Concurrente 2024-2

Kassandra Mirael ♡

2024

Tiempo estimado de realización: Unas 16 hrs (depende de que tan bueno sea con el código sin modificar mucho)

Objetivo: Implementar algoritmo del Filtro Modificado y Peterson, solucionar un problema concurrente, así como también ver que si el candado cumple con Deadlock-free, el sistema en el que se usa puede llegar a tenerlo.

Introducción

Big Boss celebrará su cumpleaños por lo que ha decidido invitar a todos sus colegas y amigos. Dadas las últimas adquisiciones hechas para reconstruir M.B, no tuvo suficiente dinero para los utensilios de comedor, y no compró suficientes tenedores para todos sus colegas. El problema es que cada persona necesita dos tenedores para comer, si solo tienen uno entonces no pueden comer.

Adicionalmente, como la celebración se realiza en una plataforma especial, los invitados requieren ingresar con vehículos, por fortuna la plataforma tiene un pequeño estacionamiento subterráneo que cuenta con varios subniveles, por lo que hay espacio suficiente para la mayoría de invitados. Sin embargo, no hay un sistema de *valet-parking*, así que cada invitado debe estacionar su vehículo solo y ese es un gran problema.

Especificaciones

Se incluyen dos proyectos en Maven, uno hace referencia al primer problema y el otro al segundo problema.

Para el primero, se incluyen las instrucciones en cada método y clase, por lo que deberás completar el código para que funcione todo correctamente. Finalmente deberás pasar todos los test.

Para el segundo proyecto, se te incluyen clases vacías con sus respectivas instrucciones y funcionamiento, completa las clases para que funcionen correctamente y posteriormente genera el sistema completo.

Descripciones de los problemas

El problema 1 esta basado en el problema clásico de los filósofos, en el cual se resolverá con cadados programados por ustedes, siendo el de Peterson y el del Filtro Modificado.

El segundo ejercicio consiste en el de un estacionamiento, en este llegan un número **n** de carros, dentro del estacionamiento se permite una entrada a un número **m** tal que $n \geq m$, es decir tiene entrada para **m** carros, estos se separaran en **h** niveles o pisos (veanlo con un estacionamiento de plaza comercial, donde hay varios pisos). Los automóviles estarán intentando ingresar de manera constante.

Consideraciones

No está permitido utilizar cosas (herramientas) que no vengan declaradas para cada problema. Para el primero **NO se puede usar ni reentrant lock, ni Semaphore, ni atomic**

Para el segundo problema solo puedes utilizar Semáforos débiles, todo lo demás queda prohibido.

Deberás agregar un par de test extra para comprobar su funcionamiento, se incluyen algunos en el proyecto, pero debes completar los que faltan.

Cuestionario

Contesta las siguientes preguntas (Para cada proyecto):

- ¿Tu solución cumple con Exclusión mutua? Argumenta porqué.
- ¿Tu solución cumple con Deadlock-free? Argumenta porqué.
- ¿Tu solución cumple con Libre de Hambruna (Starvation-free)? Argumenta porqué.
- ¿Tu solución cumple con Justicia? Argumenta porqué. En caso de que no lo cumpla, como podrías garantizarla.

Finalmente, escribe lo aprendido en esta práctica, así como las dificultades obtenidas.

Por último...

