Jordi Pereira Gil

Informe Productores consumidores

Varios productores consumidores semántica SC y FIFO

Para empezar al usar una estrategia FIFO hubo que crear las variables primera_ocupada y num_ocupadas para desplazarse de manera circular a través del buffer.

Hubo que crear un array con el tamaño del número de hebras productoras, llamado producidos, para saber cuantos datos ha producido cada hebra.

También tuve que cambiar un poco el método de producir_dato y añadirle un parámetro que usaremos como índice para acceder en el array producidos a la cantidad de datos producidos por la hebra que ha invocado dicho método.

En los métodos leer y escribir cambié las condiciones de espera por bucles while, utilizando la variable num_ocupadas, a parte en los for de los métodos funcion_hebra_productora y funcion_hebra_consumidora ahora van hasta num_items/num_prod y num_items/num_cons respectivamente para que las hebras se repartan equitativamente el trabajo.

Finalmente en el main lo que hice fue inicializar todos los elementos del array producidos a 0 y crear arrays de hebras consumidoras y productoras con sus respectivos join.

Varios productores consumidores semántica SU

Para esta implementación a la clase ProdCons1SC debemos hacer que tenga herencia de la clase HoareMonitor y las variables de condición pasarlas a la clase CondVar e inicializarlas en el constructor.

En cuanto a los métodos leer y escribir eliminamos el guarda ya que con los métodos wait y signal de la clase CondVar se encargan de que no haya exclusión mutua.

En el main creamos una Mref de ProdCons1SC que es lo que enviamos a las funuciones funcion_hebra_productora y funcion_hebra_consumidora.

- **Estrategia FIFO**: Conserva todos los while en leer y escribir igual que la versión con semántica SC hecha anteriormente y mantiene las variables mum_ocupadas y primera_ocupada.

- **Estrategia LIFO**: Se mantienen lo while solo que con las condiciones del LIFO con un productor y un consumidor y solo se trabaja con la variable primera_libre para el acceso al buffer.