Implementación de Algoritmo de Peterson y Filtro modificado

Práctica 3

Integrantes

* Espinal Cruces Martin Felipe
* Fernandez Romero Adrian Felipe
* Sánchez de la Rosa César Gustavo
* Velázquez Caballero Ixchel

Teoría

1. Proponer problemas donde se pueda utilizar el algoritmo de Peterson para su solución

    1. Problema del Consumidor-Productor. Dicho problema describe dos procesos, productor y consumidor, ambos comparten un buffer de tamaño finito. La tarea del productor es generar un producto, almacenarlo y comenzar nuevamente; mientras que el consumidor toma (simultáneamente) productos uno a uno. El problema consiste en que el productor no añada más productos que la capacidad del buffer y que el consumidor no intente tomar un producto si el buffer está vacío.

    2. Problema de la sección crítica.

```
PROCESS Pi
FLAG[i] = true
while((turn != i) AND (CS is !free)){
wait;
}
CRITICAL SECTION FLAG[i] = false;
turn j; //elegir otro proceso para ir a CS
```

    En este caso, se agrega una matriz FLAG[i] de tamaño n, que de forma predeterminada está en FALSE.

    Cada vez que un proceso necesita entrar en la sección crítica, se establece en TRUE. Si el proceso Pi necesita entrar, establecerá FLAG[i] = TRUE.

    Hay otra variable TURN. Cada vez que un proceso sale de la sección crítica, cambiará el TURN a otro número de la lista de procesos listos.

2. Proponer problemas donde se pueda utilizar el Algoritmo del Filtro

    1. Sorting Network. Esta red esta formada por procesos que tienen 2 canales de entrada y 1 de salida. Su función es, a partir de la secuencia ordenada de datos entrantes,
    generar secuencialmente la mezcla ordenada en la salida.
    Utilizando estos procesos podemos formar un ´arbol binario de profundidad n con 2n−1
    hojas. Presentando a la entrada de cada una de estas hojas 2 valores, en la raíz

obtendremos
la secuencia de 2n datos ordenados.

2. Los esquema productor-consumidor muestran procesos que se comunican. Es habitual que estos procesos se organicen en pipes a través de los cuáles fluye la información. Cada proceso en el pipe es un filtro que consume la salida de su proceso predecesor y produce una salida para el proceso siguiente.

Responde las siguientes preguntas, justificando tu respuesta:

* ¿Los algoritmos cumplen con No Deadlock?

  Un No Deadlock se produce cuando el proceso de espera todavía se aferra a otro recurso que el primero necesita antes de que pueda finalizar. La mejor manera de evitar los bloqueos es evitar que los procesos se crucen de esta manera. Reduzca la necesidad de bloquear cualquier cosa tanto como pueda.

  En los algoritmos no es posible que ocurra un Deadlock porque el proceso que primero establece la variable de turno entrará en la sección crítica con seguridad.

* ¿El Algoritmo de Peterson cumple con la propiedad de Justicia?

  Sí, si cumple con la propiedad de Justicia al tener definida la espera finita.

* ¿Cuál de estos algoritmos cumple con la propiedad Libre de Hambruna?

  El algoritmo de Peterson cumple con la propiedad de no inanición que básicamente es no hambruna. Y como el algoritmo de Filtro es peterson generalizado para n, entonces cumple también

* ¿Cumplen con Exclusión Mutua?

  Sí, los algoritmos cumplen con la propiedad de exclusión mutua

Referencias

- https://www.infor.uva.es/~cllamas/concurr/pract98/sisos4/
- https://formella.webs.uvigo.es/doc/cdg11/exclusion.pdf
- https://www.cs.cmu.edu/~kaess/pub/Kaess12fusion.pdf