

Programación Concurrente y de Tiempo Real

Asignación de Prácticas 5

José Joaquín Arias Gómez-Calcerrada

November 18, 2016

Algoritmo de Dekker para 3 procesos

El algoritmo de Dekker modificado para 3 hilos en vez de para 2, como fue originalmente diseñado, sigue cumpliendo las condiciones de dijkstra de la exclusión mutua.

Aquí le muestro una tabla con el número de iteraciones de cada hilo, siendo el valor de la variable común 0 y teniendo en cuenta que dos de los hilos incrementan esta variable en 1 cada iteración y el tercer hilo se encarga de decrementarla.

Número de iteraciones	Valor variable común real	Valor variable común esperado
10	10	10
100	100	100
1000	1000	1000
10000	10000	10000
100000	100070	100000
1000000	999999	1000000

Como se puede observar, a partir de un número elevado de iteraciones se produce entrelazado, es decir, el programa no devuelve el valor que debería debido a que la variable compartida es modificada por más de un hilo al mismo tiempo.

Algoritmo Eisenberg implementado en java

El algoritmo de Eisenberg fue la primera solución tomada para el problema de la sección crítica y fué diseñado para N procesos, aunque en mi implementación sólo hay 2. Posee una variable turno, que oscilará entre 0 y n-1, siendo n el número de procesos. Este algoritmo posee un tipo enumerado que permite 3 posiciones: idle, waiting y active para un control más completo del deseo de acceso de cada hilo a la sección crítica.

Aquí le muestro una tabla con el número de iteraciones de cada hilo, siendo el valor de la variable común 0 y teniendo en cuenta que un hilo incrementa esta variable en una unidad cada iteración y el segundo hilo se encarga de decrementarla.

Número de iteraciones	Valor variable común real	Valor variable común esperado
10	0	0
100	0	0
1000	0	0
10000	0	0
100000	0	0
1000000	0	0

Como se puede observar, este algoritmo da solución por completo al dilema de la sección crítica independientemente del número de iteraciones.