

Computación Concurrente

Tarea 5: Estructuras de datos concurrentes

Juan Carlos Bautista Sandoval

Ernesto Muñoz Nieves

Carlos Cruz Rangel

José Demian Jiménez

Victor Hugo Gallegos Mota

Facultad de ciencias, UNAM

4 de noviembre de 2022



Teoría

1. ¿Para que sirve el método Yield? (yield())

El método `yield()` es un método el cual permite detener temporalmente el hilo que actualmente se esta ejecutando permitiendo que se ejecuten otros hilos que están en espera cuya prioridad sea mayor o igual al del hilo detenido . Si no hay hilos en espera o si todos los hilos en espera tienen menor prioridad, el mismo hilo continuará su ejecución. Cabe recalcar que cuando llamamos al método `yield()`, le decimos al scheduler, que el hilo que utilizo `yield` esta listo para pausar su ejecución de ser necesario, sin embargo el scheduler puede ignorar esto, por lo tanto `yield` no garantiza que un hilo de verdad le pase su ejecución a otro, o que este se detenga.

2. ¿Qué es un atributo atómico? (En la biblioteca `atomic` de Java)

Para entender los atributos atómicos en Java, necesitamos entender las operaciones atómicas, las cuales consisten en la creación de algoritmos de no bloqueo para procesos concurrentes. Es decir las operaciones atómicas contrario a las candados, no dejan en espera a otros hilos. En java los atributos atomicos más usados son `AtomicInteger`, `AtomicLong`, `AtomicBoolean` y `AtomicReference`, que representan clases para los tipos primitivos y referencias que pueden ser automáticamente actualizados.

3. Ventajas de usar atributos atómicos

Contrario a los candados, los atributos atómicos no suspenden a otros hilos, en su lugar simplemente le informan a esos hilos que no pudieron actualizar la variable en cuestión, permitiendo que dichos hilos puedan continuar trabajando y evitando así los cambios de contexto los cuales son operaciones muy costosas.

4. Desventajas de usar atributos atómicos

La principal desventaja es que la lógica del programa se vuelve más compleja, ya que debemos de manejar los casos cuando las actualizaciones a las variables no se pudieron realizar, como por ejemplo volver a intentarlo o continuar con otras partes del programa.

5. Da 2 ejemplos en donde se puedan aplicar este tipo de atributos.

- a) Se puede utilizar para incrementar el valor de un contador de manera concurrente.
- b) Par mantener una bandera booleana de manera consistente entre diversos hilos.

6. Algún uso que creas que se da en estructuras de datos concurrentes.

Yo pienso que se puede hacer uso de estas estructuras por ejemplo en los tableros Kanban, podemos suponer que cada contenedor, es una lista por lo tanto cada usuario en el tablero puede agregar tarjetas a un contenedor(lista), además de hacer modificaciones sobre la misma, en estos casos se adaptaría bien un estructura de datos concurrente, para poder mantener la consistencia de la lista entre los múltiples usuarios que la modifiquen.