Teoria Współbieżności
Ćwiczenie 4

1 Cel ćwiczenia

Celem ćwiczenia jest zapoznanie studentów z wydajnością blokowania drobnoziarnistego (w Javie).

2 Wprowadzenie teoretyczne

Lock jest przydatny wtedy, gdy operacje zamykania/otwierania nie mogą być umieszczone w jednej metodzie lub bloku synchronized. Przykładem jest zakładanie blokady (lock) na elementy struktury danych, np. listy. Podczas przeglądania listy stosujemy następujący algorytm:

1. zamknij lock na pierwszym elemencie listy
2. zamknij lock na drugim elemencie
3. otwórz lock na pierwszym elemencie
4. zamknij lock na trzecim elemencie
5. otwórz lock na drugim elemencie
6. powtarzaj dla kolejnych elementów

Dzięki temu unikamy konieczności blokowania całej listy i wiele wątków może równocześnie przeglądać i modyfikować różne jej fragmenty.

3 Plan ćwiczenia

1. Proszę zaimplementować listę, w której każdy węzeł składa się z wartości typu ***Object***, referencji do następnego węzła oraz zamka (lock). Lista nie może przechowywać wartości *null*.

   Proszę zastosować metodę drobnoziarnistego blokowania do następujących metod listy:

   - ***boolean*** *contains*(***Object*** *o*); //czy lista zawiera element o
   - ***boolean*** *remove*(***Object*** *o*); //usuwa pierwsze wystąpienie elementu o
   - ***void*** *add*(***Object*** *o*); //dodaje element o na koncu listy

2. Ogranicz listę do n elementów, wsytrzymuj dzialanie remove i add a za pomocą zmiennych warunkowych gdy nie możesz nic usunać albo dodać.

3. Wykonaj symulację/przetestuj.

Problem czytelników i pisarzy możesz rozwiązać przy pomocy ReentrantLock albo ReentrantReadWriteLock.

Wykonaj symulację na przykład dla różnej ilości czytelników (10-100) i pisarzy (od 1 do 10).

Porównaj **wydajność** tego rozwiązania w stosunku do listy z jednym zamkiem blokującym dostęp do całości. Należy założyć, że koszt czasowy operacji na elemencie listy (porównanie, wstawianie obiektu) może być duży - proszę wykonać pomiary dla różnych wartości tego kosztu.

Literatura

[1] Bruce Eckel, "Thinking in Java" - rozdział o wątkach

[2] ReentrantLock - link do dokumentacji

[3] ReentrantReadWriteLock - link do dokumentacji

[4] Zmienne warunkowe - link do dokumentacji

Uwagi: sluzalec@agh.edu.pl