

Documentație Laborator 3

Muntea Andrei - Marius (235)

I Cerință

Implementați o clasă *SortedList* - listă înlănțuită sortată cu elemente de tip real astfel încât să fie thread-safe. Operațiile furnizate de clasă sunt:

- Insert(a:Real)
- Delete(a:Real)
- GetIterator(): Iterator

Clasa *Iterator* va fi definită corespunzător cu specificația clasică a unui iterator (implementează interfața iterator cu operațiile *GetNext()*; *IsValid()*; *GetElement()*).

Se cere implementarea a două variante:

- Sincronizare la nivel de nod sau porțiune din listă (*Fine Grained synchronization*)
- Sincronizare la nivelul întregii liste (*Coarse Grained synchronization*)

La testarea iteratorului se va folosi întotdeauna sincronizarea întregii liste.

II Idee de rezolvare

Vom implementa o clasă abstractă *DoublyLinkedList<T>* care va avea 2 metode pur virtuale (adăugarea, respectiv ștergerea unui element). În această clasă vom ține un obiect de tipul *shared_mutex*. Iteratorul va fi definit tot în această clasă și va încerca să obțină o blocare exclusivă a mutexului respectiv. Vom avea alte două clase, *FineGrainedList<T>* și *CoarseGrainedList<T>* care vor moșteni din *DoublyLinkedList<T>*. În clasa *CoarseGrainedList* se va încerca o blocare exclusivă a mutexului din listă la operațiile de ștergere respectiv de adăugare, iar la *FineGrainedList* se va încerca o blocare de tip shared la cele două operații, sincronizarea făcându-se la nivel de nod prin metoda *Hand-Over-Hand Locking*.

III Metode de testare

Vom avea două teste (Testul1 și Testul2, ambele având 4 threaduri) fiecare dintre cele două teste fiind rulat de 10 ori pentru fiecare dintre cele 2 implementări ale listei. Timpii măsurați nu vor fi foarte preciși deoarece fiecare thread așteaptă timp de 15 ms după fiecare operație de adăugare, respectiv ștergere pentru a putea calcula mai ușor timpul de așteptare necesar threadului care se ocupă cu iterarea elementelor.

- Testul1:
 - Threadul 1 adaugă 10 valori
 - Threadul 2 adaugă 5 valori
 - Threadul 3 șterge 5 valori
 - Threadul 4 iterează lista în mod repetat la un interval t și se termină după ce threadurile 1-3 se termină
- Testul 2:
 - Threadul 1 adaugă 100 valori
 - Threadul 2 adaugă 50 valori
 - Threadul 3 șterge 50 valori

- Threadul 4 iterează lista în mod repetat la un interval t și se termină după ce threadurile 1-3 se termină

Fiecare test o să înregistreze într-un fișier (*Log1.txt*, respectiv *Log2.txt*) operațiile efectuate și indicele threadului care le efectuează.

T1	Coarse Grained Synchronization	Fine Grained Synchronization
1	165809.388000	169145.077000
2	169252.056000	170424.875000
3	170790.026000	171568.879000
4	169981.960000	169252.056000
5	171588.222000	165818.467000
6	171545.194000	171103.856000
7	170075.912000	166970.760000
8	171777.310000	169134.024000
9	168546.627000	166687.720000
10	169668.918000	172701.434000

T2	Coarse Grained Synchronization	Fine Grained Synchronization
1	1600780.432000	1572767.753000
2	1603801.895000	1595442.936000
3	1565200.674000	1601753.111000
4	1578448.688000	1572331.943000
5	1598000.956000	1572208.779000
6	1557893.740000	1585061.641000
7	1561232.586000	1568895.986000
8	1568270.298000	1438260.235000
9	1585960.501000	1626708.011000
10	1561625.368000	1579727.697000

