|  |  |
| --- | --- |
| Threads | Timp |
| 2 | 142 |
| 4 | 139 |
| 12 | 148 |

Datele de la laboratorul precedent

|  |  |  |
| --- | --- | --- |
| Secvential | | 223 |
| 1 reader | 4 | 89 |
| 6 | 94 |
| 8 | 90 |
| 16 | 93 |
| 2 reader | 4 | 138 |
| 6 | 143 |
| 8 | 156 |
| 16 | 135 |

Dupa cum se poate observa timpii obtinuti pentru 2 readeari este asemanator la cele doua rulari pentru valori mici ale nr. de writeri, dar la numar mai mare de writeri, varianta cu variabila conditionala este mai eficienta.

Pentru rezolvarea acestei probleme am ales sa construiesc 2 structuri de date, o coada sincronizata care are o capacitate maxima, si operatiile de adaugare si stergere se "blocheaza" in momentul cand datele nu exista sau sunt prea multe. Sincronizarea se face folosind variabile conditionala

Am implementat si o lista simplu inlantuita sincronizata care are un lock pentru fiecare element si sunt blocate elemenetele in functie de ce operatie se realizeaza. Pentru update se blocheaza fiecare nod pe masura ce il citesc, pentru adauagare, aceasta realizandu-se tot timpul la inceputul listei se blocheaza 2 noduri. Iar pentru stergere se blocheaza cate 3 elemente.

Participantii care au primit odata -1 vor fi adaugati intr-o "lista neagra" de unde ii voi adauga la final in lista initiala. Sortarea se face utilizand bubble sort.