Lab04 - Producator - Consumator

Varianta secventiala a fost implementata folosind un o lista simplu inlantuita care este updatata constant pentru a ramane in permanenta ordonata descrescator (de fiecare data cand actualizez un element, incerc sa il urc cat de sus posibil in ierarhie, cand primesc -1 pentru element il retrogradez la finalul listei si sterg elementul respectiv, la final scriu rezultatul in fisier)

Pentru varianta paralela am ales sa creez doua clase care vor deservi despre LinkedList si Queue, carora le-am implementat si sincronizare. LinkedList-ul este sincronizat folosind un lock iar Queue-ul folosind mecanismele wait & notifiy disponibile in Java. Threadurile disponibile se impart in doua categorii: readeri si writeri. Readeri primesc un set de fisiere, din care vor citi datele si le vor adauga in coada, coada din care Writerii vor citi cate o intrare pe rand si vor incerca adaugarea ei in lista. De fiecare data cand nu mai exista elemente in coada, se va apela functia wait, astfel thread-ul asteptand pana la "producerea" de noi elemente. De fiecare data cand un reader produce noi elemente acesta ii va notifica pe toti writerii blocati in wait, ca exista date pe care acestia le pot citi. La final se scriu rezultatele in fisier si se verifica corectitudinea datelor folosind rezultatele obtinute la calculul secvential.

Timp:

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