

Οδηγίες Υλοποίησης :

Read_Priority: Εδώ διαβάζουμε όλα τα αρχεία που βρίσκονται μέσα στο txt σε αύξουσα σειρά βάση του ArrivalTime του κάθε αρχείου. Η **loadFile** έχει δηλωθεί ως static προκειμένου να μπορέσουμε να τη χρησιμοποιήσουμε τον reader στις **AlgorithmB**, **AlgorithmC** ξεχωριστά χωρίς να δημιουργούμε αντικείμενα της.

Writer: Οι **AlgorithmB**, **AlgorithmC** υλοποιούν ξεχωριστά μέσα τους τον δικό τους writer (Η **AlgorithmB** χρησιμοποιεί ειδικά την toString της κλάσης PrintJob).

PrintJob: Η κλάση αυτή δημιουργεί αντικείμενα τύπου PrintJob καθώς επίσης αρχικοποιεί και την προτεραιότητα κάθε αρχείου. Ακόμα υλοποιεί και την compareTo (όπου γίνεται σύγκριση 2 αντικειμένων τύπου (PrintJob) με βάση την προτεραιότητα τους.

AlgorithmB: Η υλοποίησή της γίνεται σε δύο εμφωλευμένα while-loops. Το εξωτερικό διαρκεί όσο η λίστα δεν είναι άδεια. Όσο διαρκεί αυτό, πρώτα αφαιρούμε από τη λίστα ένα αντικείμενο (πάντα από την αρχή της λίστας) και έπειτα το εκχωρούμε στην ουρά προτεραιότητας. Στη συνέχεια γίνεται έλεγχος για ταυτόχρονη άφιξη 2 ή παραπάνω αρχείων και εκχωρούνται και αυτά στην ουρά. Έπειτα στο δεύτερο loop (όσο δηλαδή η ουρά προτεραιότητας δεν είναι άδεια) αφαιρούμε το στοιχείο με τη μέγιστη προτεραιότητα (μικρότερο maxsize) και τέλος αφού ενημερωθούν κατάλληλα ο χρόνος αναμονής και ο χρόνος άφιξης εκτυπώνεται το αρχείο. **Σημείωση:** ο χρόνος αναμονής υπολογίζεται στο τέλος γιατί δεν επηρεάζει την προτεραιότητα των αρχείων. Τέλος, γίνεται εισαγωγή των αρχείων που ήδη έχουν καταφτάσει μέχρι εκείνη τη χρονική στιγμή στην ουρά προτεραιότητας.

AlgorithmC: Η υλοποίησή της είναι παρόμοια με της **AlgorithmB** με την διαφορά ότι η προτεραιότητα του κάθε αρχείου εξαρτάται και από το χρόνο αναμονής του στην ουρά. Συνεπώς, τη στιγμή που προστίθεται στην ουρά ένα αντικείμενο, αρχικοποιείται κατάλληλα ο χρόνος παραμονής του. Η βοηθητική μεταβλητή sumTime αποθηκεύει τον συνολικό χρόνο που έχει περάσει από το τελευταίο update. Όταν sumTime >= 15 (δευτερόλεπτα) γίνονται update όλα τα priority και τα waitingTime κατάλληλα. Σε διαφορετική περίπτωση γίνεται update μόνο το waitingTime.

MaxPQ: Η γενική υλοποίησή της είναι παρόμοια με αυτή του εργαστηρίου. Επιπλέον η updateWaitingTime αλλάζει το waitingTime του κάθε αρχείου (όσο η ουρά προτεραιότητας περιέχει αντικείμενα τύπου PrintJob). Η updatePriorities κάθε 15 δευτερόλεπτα αλλάζει το priority του κάθε αρχείου (όσο η ουρά προτεραιότητας περιέχει αντικείμενα τύπου PrintJob) βάση του τύπου $priority = \min(127, priority + waitingTime)$.

Συγκρίσεις :

Παρατηρήσεις : Τις περισσότερες φορές οι average waitingTime και maximum waitingTime του **AlgorithmB** είναι μικρότερα του **AlgorithmC**. Αυτό συμβαίνει διότι στον **AlgorithmB** όλα τα αρχεία με μεγάλο μέγεθος εκτυπώνονται στο τέλος, ενώ στην αρχή εκτυπώνονται όλα τα αρχεία με μικρό μέγεθος. Έτσι στην αρχή προστίθενται πολύ μικροί μέσοι όροι στο average waitingTime. Σε αντίθεση, στον **AlgorithmC** τα αρχεία με μεγάλο μέγεθος δεν εκτυπώνονται όλα μαζί στο τέλος, αλλά σε μια ενδιάμεση χρονική στιγμή το οποίο αυξάνει τα waitingTime των αρχείων με μικρότερο μέγεθος (τα οποία δηλαδή εκτυπώνονται μετά από τα μεγάλα αρχεία). Συνεπώς, το **AlgorithmB** αρμόζει σε περιπτώσεις όπου χρειαζόμαστε τα αρχεία να εκτυπώνεται σε μικρό σχετικά χρόνο, ενώ το **AlgorithmC**

στις περιπτώσεις τις οποίες ενδιαφερόμαστε κυρίως για την δίκαιη προτεραιότητα των αρχείων προς εκτύπωση και όχι τόσο για το συνολικό χρόνο.

AlgorithmB

Average waiting time Maximum waiting time |

AlgorithmC

Average waiting time Maximum waiting time

File1	<u>847648</u>	<u>2543335</u>	<u>1274778</u>	<u>2543289</u>
File2	<u>844049</u>	<u>2537092</u>	<u>1264864</u>	<u>2537244</u>
File3	<u>852953</u>	<u>2549403</u>	<u>1269857</u>	<u>2549691</u>
File4	<u>858573</u>	<u>2563091</u>	<u>1278881</u>	<u>2563270</u>
File5	<u>856066</u>	<u>2557219</u>	<u>1279313</u>	<u>2557137</u>
File6	<u>852691</u>	<u>2549210</u>	<u>1272500</u>	<u>2549209</u>
File7	<u>846395</u>	<u>2539266</u>	<u>1269007</u>	<u>2539355</u>
File8	<u>851013</u>	<u>2545495</u>	<u>1268163</u>	<u>2545704</u>
File9	<u>852335</u>	<u>2551245</u>	<u>1272315</u>	<u>2551172</u>
File10	<u>847508</u>	<u>2539587</u>	<u>1267846</u>	<u>2540097</u>
File11	<u>26166</u>	<u>77947</u>	<u>38713</u>	<u>77957</u>
File12	<u>27768</u>	<u>81518</u>	<u>40269</u>	<u>81589</u>
File13	<u>27578</u>	<u>81246</u>	<u>40232</u>	<u>81355</u>
File14	<u>25916</u>	<u>78289</u>	<u>39520</u>	<u>78331</u>
File15	<u>27253</u>	<u>80435</u>	<u>39763</u>	<u>80545</u>
File16	<u>27486</u>	<u>80886</u>	<u>40919</u>	<u>80896</u>
File17	<u>27154</u>	<u>80042</u>	<u>39900</u>	<u>80057</u>
File18	<u>25877</u>	<u>77638</u>	<u>38230</u>	<u>77660</u>
File19	<u>26124</u>	<u>78520</u>	<u>39264</u>	<u>78607</u>
File20	<u>25178</u>	<u>76736</u>	<u>38439</u>	<u>76870</u>
File21	<u>535299</u>	<u>1600709</u>	<u>801080</u>	<u>1600831</u>
File22	<u>531094</u>	<u>1593079</u>	<u>797406</u>	<u>1593269</u>
File23	<u>535130</u>	<u>1597608</u>	<u>798550</u>	<u>1597653</u>
File24	<u>534576</u>	<u>1599488</u>	<u>797067</u>	<u>1599531</u>
File25	<u>534878</u>	<u>1603298</u>	<u>801309</u>	<u>1603651</u>
File26	<u>536367</u>	<u>1600891</u>	<u>799468</u>	<u>1600871</u>
File27	<u>533098</u>	<u>1597197</u>	<u>797959</u>	<u>1597021</u>
File28	<u>535783</u>	<u>1599106</u>	<u>800290</u>	<u>1599137</u>
File29	<u>539530</u>	<u>1608362</u>	<u>801459</u>	<u>1608595</u>
File30	<u>532703</u>	<u>1594830</u>	<u>796104</u>	<u>1594776</u>
File31	<u>155248</u>	<u>466656</u>	<u>232440</u>	<u>466722</u>
File32	<u>161159</u>	<u>478062</u>	<u>238052</u>	<u>478180</u>
File33	<u>159162</u>	<u>474555</u>	<u>235714</u>	<u>474610</u>
File34	<u>156835</u>	<u>469381</u>	<u>232687</u>	<u>469515</u>
File35	<u>158434</u>	<u>471471</u>	<u>234960</u>	<u>471516</u>
File36	<u>158271</u>	<u>473563</u>	<u>237259</u>	<u>473619</u>
File37	<u>157171</u>	<u>469637</u>	<u>233890</u>	<u>469710</u>
File38	<u>156432</u>	<u>468261</u>	<u>232623</u>	<u>468343</u>
File39	<u>157026</u>	<u>470135</u>	<u>234875</u>	<u>470287</u>
File40	<u>155913</u>	<u>469242</u>	<u>233320</u>	<u>469354</u>
File41	<u>342886</u>	<u>1024296</u>	<u>512384</u>	<u>1024292</u>
File42	<u>342618</u>	<u>1023980</u>	<u>508796</u>	<u>1023875</u>

File43	<u>338743</u>	<u>1015348</u>	<u>506551</u>	<u>1015398</u>
File44	<u>338734</u>	<u>1015459</u>	<u>507603</u>	<u>1015629</u>
File45	<u>343096</u>	<u>1022951</u>	<u>511285</u>	<u>1023068</u>
File46	<u>339493</u>	<u>1018289</u>	<u>509515</u>	<u>1018420</u>
File47	<u>339782</u>	<u>1014512</u>	<u>505367</u>	<u>1014489</u>
File48	<u>338762</u>	<u>1014643</u>	<u>506889</u>	<u>1014800</u>
File49	<u>344285</u>	<u>1025495</u>	<u>509419</u>	<u>1025503</u>
File50	<u>339152</u>	<u>1014546</u>	<u>506881</u>	<u>1014710</u>