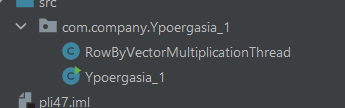
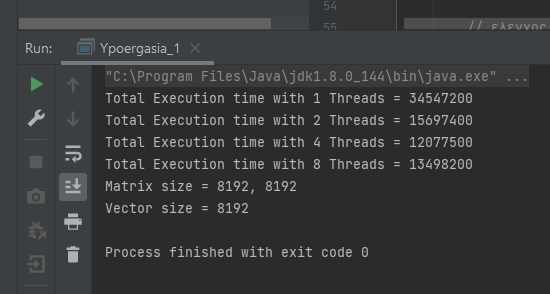
Directory :



PrintScreen εκτελεσης :



Σχολιασμος:

* Το συστημα στο οποιο εγινε η εκτελεση εχει 6 πυρηνες και 12 νηματα.
* Παρατηρουμε αυτο που η λογικη μας λεει οτι τα 4 Thread «τρεχουν» πιο γρηγορα απο τα 2 και αυτα ομοιως απο 1 Thread.  
  Συνεχιζοντας βλεπουμε οτι χρησιμοποιωντας 8 Thread δεν υπαρχει βελτιωση απο την χρηση 4 Thread
* Σημειωση: σε μικροτερο ογκο δεδομενων δεν υπαρχει εμφανης λογος να χρησιμοποιησουμε παραπανω απο 1 νηματα, διοτι η εργασια συνενωσης των αποτελεσματων επιβαρυνει την εκτελεση αρα τα οφελη δεν ειναι εμφανη.  
  Επισης οι ανω αριθμοι στο στιγμιοτυπο οθονης δεν ειναι σταθεροι, εχουν μεγαλες αποκλισεις απο εκτελεση σε εκτελεση. Κατα μεσο ορο ομως οι πιθανοτητες ειναι μια παρομοια εκτελεση αναλογικα με τους χρονους.
* Υπαρχουν βοηθητικες μεθοδοι (helper methods) για εκτυπωση του Πινακα print2D, για εκτυπωση του διανυσματς print1D
* Οι μεθοδοι getVector, getMatrix παραγουν το διανυσμα και τον πινακα, με ορισματα n, m.
* Υπαρχει ελεγχος για ισοτητα των πινακων αποτελεσματων.
* Οι μεθοδοι της Ypoergasias\_1 🡪 oneThreadMult, twoThreadMult, fourThreadMult, eightThreadMult χρησιμοποιουν την κλαση RowByVectorMultiplicationThread για να κανουν τον υπολογισμο αναλογα με τον αριθμο Threads που επιθυμουμε.
* Στην κλαση main συγκεντρωνονται ολα τα παραπανω για να υλοποιηθει το ζητουμενο της ασκησης.