

## Εργασία 5 (OpenMP).

Υλοποιείτε σε OpenMP τις ασκήσεις

### 1. Εργασία 1, άσκηση 2

Η άσκηση λύθηκε στο μάθημα. Εκτελέστε τα ίδια πειράματα και συγκρίνετε χρόνους με την υλοποίηση με Java Threads.

### 2. Εργασία 2

Μια αρχική μορφή της λύσης σε OpenMP βρίσκεται στο αρχείο `omp_intro.zip` που επισυνάπτεται. Ξεκινήστε από το καλύτερο ακολουθιακό αλγόριθμο που είχατε υλοποιήσει και τροποποιείτε ανάλογα την υλοποίηση του OpenMP. Συγκρίνετε χρόνους με την αντίστοιχη υλοποίηση με Java Threads.

Δοκιμάστε να προσθέσετε `collapse (2)` και `collapse (3)`.

Δοκιμάστε κάποιες από τις ιδέες που προτείνονται στο διαγωνισμό της Intel (δείτε το αρχείο της Εργασίας 2).

### 3. Εργασία 3, άσκηση 2

Η άσκηση λύθηκε στο μάθημα. Μια λύση της σε Jomp βρίσκεται στο αρχείο `jomp.zip` που επισυνάπτεται. Υλοποιείτε 2 λύσεις, μια με `pragma omp critical` και μια με `reduction`. Εκτελέστε τα ίδια πειράματα και συγκρίνετε χρόνους με την αρχική υλοποίηση και τις αντίστοιχες υλοποιήσεις με Java Threads.

### 4. Εργασία 3, άσκηση 3

Η λύση είναι παρόμοια με αυτή της προηγούμενης άσκησης αλλά (μάλλον) δεν μπορεί να χρησιμοποιηθεί το `reduction` σε δομή δεδομένων. Επομένως προσπαθείστε για λύση με `pragma omp critical` και συγκρίνετε χρόνους με τις αντίστοιχες υλοποιήσεις με Java Threads.

## Σημείωση

Στα zip αρχεία θα βρείτε ορισμένους ενδεικτικούς κώδικες και μέτρηση χρόνου σε OpenMP.

Αν θέλετε μπορείτε να χρησιμοποιήσετε Java με Jomp ή `omp4j`. Προτιμώ τη λύση σε C γιατί υλοποιεί πληρέστερα το OpenMP.

Για το Jomp δείτε οδηγίες στο επισυναπτόμενο `jomp.zip` καθώς και στο σύνδεσμο <http://www.it.uom.gr/teaching/ParallelDistributedJava/software/jomp/jompAPI.pdf>

Στο σύνδεσμο

<http://www.it.uom.gr/teaching/ParallelDistributedJava/threadsoftware.html#jomp>

θα βρείτε και ορισμένα ακόμη παραδείγματα.

Για το `omp4j` βρείτε downloads και δείτε οδηγίες (παρόμοιες με το Jomp) στο σύνδεσμο

<http://www.omp4j.org/download>

Στο σύνδεσμο

<http://www.omp4j.org/tutorial>

θα βρείτε αρκετά παραδείγματα.

