

# Παράλληλος Προγραμματισμός 2020

## Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: Γρηγορούδης Δημοσθένης  
Α.Μ.: Π2016022

Ο κώδικας δεν είναι ολοκληρωμένος - εκτελέσιμος. Βασίστηκα στους κώδικες του εργαστηρίου για το κομμάτι της υλοποίησης της quicksort και στη συνέχεια πρόσθεσα και έκανα κάποιες απαραίτητες αλλαγές για να ταιριάζει (όσο γίνεται) με τα threads. Στη main γίνεται η αρχικοποίηση της ουράς και επίσης την γεμίζουμε με τυχαίους αριθμούς. Στη συνέχεια δημιουργούμε τα THREADS και στέλνουμε το πρώτο μήνυμα στην ουρά. Έπειτα έρχεται η σειρά μιας while επανάληψης που ελέγχει κάθε μήνυμα που δέχεται από την ουρά. Αν αυτό είναι terminated msg (1) τότε αυξάνει έναν μετρητή στον οποίο βασίζεται στο πότε θα σταματήσει το πρόγραμμα, δηλαδή πότε θα ταξινομηθεί η ουρά (δεν έχει υλοποιηθεί). Τέλος ελέγχουμε αν έγινε σωστή ταξινόμηση και κάνουμε τις απαραίτητες ενέργειες για απελευθέρωση ουράς.

Από συναρτήσεις έχουμε τις quicksort, inssort, partition, thread\_func, send, receive. Η inssort κάνει ταξινόμηση με αλγόριθμο insertion sort. Η quicksort χωρίζει τον πίνακα σε δύο μέρη και καλεί την partition για κάθε μέρος της ουράς. Η partition εκτελεί τον γνωστό αλγόριθμο της quick sort. Η send στέλνει ένα μήνυμα στην ουρά, ενώ η receive επιστρέφει στην ουρά τον αριθμό 1 (terminated) όποτε καλείται.

Τέλος η thread\_func εκτελείται μέχρι να ταξινομηθεί ο πίνακας. Δέχεται μήνυμα από την ουρά και αν η ουρά είναι μεγαλύτερη από το CREATE\_LIMIT δημιουργεί δύο threads για να ταξινομήσουν το κάθε κομμάτι της ουράς, αλλιώς καλείται η quicksort. (η thread\_func είναι ελλιπής)