



# Παράλληλος Προγραμματισμός

Εργασία #2

Ονοματεπώνυμο: Μαρία Παπαναγιώτου  
Α/Μ: Π2016020

Ο κώδικας αποτελείται από μια ουρά δεδομένων τύπου `message`, τις συναρτήσεις `sent`, `receive`, `thread_func` και την `main`.

Ο τύπος `message` είναι ένα `struct` που αποτελείται από 3 μεταβλητές του τύπου(`type`) του μηνύματος που είναι είτε `work` ανάθεση πακέτου για επεξεργασία, είτε `finish` για ολοκλήρωση του πακέτου όταν ταξινομηθεί ή `shutdown` για τον τερματισμό του `thread`, την αρχική θέση (`start_pos`) του πίνακα και την τελική θέση (`end_pos`) του πίνακα.

Για την λειτουργικότητα του προγράμματος χρησιμοποιούνται οι συναρτήσεις `send` και `receive` που προσθέτουν και αφαιρούντα πακέτα από την ουρά αφού κάνουν απαραίτητους ελέγχους με `mutex` και `conditional variables`.

Η κυρία συνάρτηση που τρέχουν τα `threads` είναι η `thread_func` όπου και εμπεριέχει όλη την λειτουργικότητα του `sorting` του πίνακα από το αρχικό σημείο που θα λάβει ως το τελικό όπως τα λαμβάνει από το `struct`.

Έτσι λοιπόν αφού ελέγξει για τον τύπο του πακέτου αν αυτός είναι `work` ελέγχει αν τα στοιχεία που του έχουν ανατεθεί είναι μικρότερα της σταθεράς `LIMIT` όπου είναι το όριο του πίνακα που ταξινομείται επιτόπου από το `thread`. Αν είναι μεγαλύτερο δημιουργεί δύο νέα πακέτα όπου χωρίζει τον πίνακα στην μέση και τα προσθέτει στην ουρά.

Η `main` συνάρτηση δημιουργεί τον πίνακα που θα πρέπει να ταξινομηθεί με τυχαίες τιμές `float` και αρχικοποιεί τα `threads` καθώς και ξεκινάει την διαδικασία με ένα πακέτο από την αρχή του πίνακα ως το τέλος του.

Αν λάβει μήνυμα τύπου `finish` αυξάνει ένα `counter` με το πλήθος των στοιχείων που έχουν ταξινομηθεί, αλλιώς προωθεί το μήνυμα. Όταν όλα τα στοιχεία έχουν ταξινομηθεί τότε στέλνει ένα μήνυμα τύπου `shutdown` για να τερματιστούν τα `threads`.

Τέλος ενώνει τα `threads` και έχουμε ως αποτελεσμάτων ταξινομημένο πίνακα.