

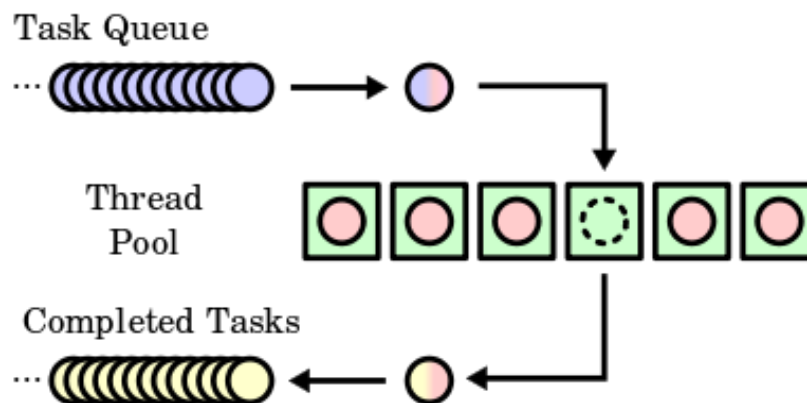
## Παράλληλος Προγραμματισμός 2019

### Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: Ανδρεάνα Στυλίδου  
ΑΜ: Π2015180

### Αναφορά για την Υλοποίηση του Quicksort με Δεξαμενή Threads

Στο πρόγραμμα της παρούσας εργασίας υλοποιείται ο αλγόριθμος Quicksort, χρησιμοποιώντας thread pool τεσσάρων threads, που αναλαμβάνουν πακέτα εργασιών από την κυκλική ουρά εργασιών.



Δεξαμενή threads που επεξεργάζεται πακέτα με χρήση ουράς εργασιών. (Εικόνα: [Wikipedia](https://en.cppreference.com/w/cpp/thread/thread-pool))

Η ουρά των εργασιών υλοποιείται σε μορφή πίνακα **mqueue**, ο οποίος κρατάει πακέτα εργασίας τύπου **message** και είναι **N (1000000)** θέσεων.

Το **message** είναι το C struct που φιλοξενεί τα μηνύματα (πακέτα) που στέλνονται στην ουρά και αποτελείται από τον τύπο του μηνύματος (**int type**) του, την αρχή ενός τμήματος πίνακα (**int start**) και το τέλος του (**int end**).

Ο τύπος ορίζεται με σταθερά στην αρχή του προγράμματος και είναι

- **WORK (0):** απαιτείται επεξεργασία
- **FINISH (1):** ολοκλήρωση επεξεργασίας
- **SHUTDOWN (2):** ολοκλήρωση ταξινόμησης και τερματισμός

Για την υλοποίηση του Quicksort χρησιμοποιήθηκε ο κώδικας του εργαστηρίου ο οποίος αποτελείται από τις συναρτήσεις **partition()** και **inssort()**.

Ο κώδικας που εκτελείται για κάθε thread εμπεριέχεται στη συνάρτηση **thread\_func()**. Για κάθε νέο πακέτο που βρίσκει στην ουρά, εκτελεί τη ταξινόμηση. Εάν το μέγεθος του πακέτου είναι μικρότερο του **CUTOFF (10)** καλείται η **inssort()**, αλλιώς γίνεται **partition** του πίνακα, και τα δύο καινούργια τμήματα προωθούνται σε μορφή πακέτων στην ουρά.

Στη συνάρτηση **main()**:

Δημιουργείται ο πίνακας με τυχαίες τιμές προς ταξινόμηση με τη χρήση της **rand()**.

Δημιουργείται ο πίνακας με τα τέσσερα threads και όρισμα τον πίνακα προς ταξινόμηση.

Στέλνοντας το πρώτο πακέτο στην ουρά με τύπο WORK και μέγεθος ολόκληρο τον πίνακα, ξεκινάει η ταξινόμηση. Έπειτα ελέγχονται συνεχόμενα τα πακέτα της ουράς και αθροίζονται τα ολοκληρωμένα (FINISH) πακέτα στη μεταβλητή **completed**. Μόλις τα συνολικά ταξινομημένα γίνουν ίσα με το μέγεθος του πίνακα, δηλαδή ταξινομηθούν όλα τα στοιχεία, η παρακολούθηση πακέτων σταματάει, στέλνεται μήνυμα τερματισμού (SHUTDOWN) και πραγματοποιείται το **join** των threads. Τέλος, ελέγχεται η ορθότητα της ταξινόμησης και αποδεσμεύονται όλες οι δομές του προγράμματος.