Παράλληλος Προγραμματισμός 2019
Προγραμματιστική Εργασία #2

Ονοματεπώνυμο: Δημήτριος Τριανταφύλλου
ΑΜ: Π2015077

Εισαγωγή

Η παρούσα εργασία αφορά την υλοποίηση του αλγορίθμου Quicksort ώστε να χρησιμοποιεί μία δεξαμενή από threads γνωστή και ως “thread pool” στη γλώσσα προγραμματισμού C. Η δεξαμενή αυτή περιλαμβάνει σταθερό αριθμό από threads τα οποία θα αναλαμβάνουν πακέτα εργασίας από μια καθολική ουρά εργασιών. Οι κώδικες που χρησιμοποιήθηκαν ως βάση της εργασίας είναι αυτοί που αναπτύχθηκαν κατά τη διάρκεια του όγδοου εξαμήνου στο εργαστήριο του μαθήματος “Παράλληλος Προγραμματισμός”. Βέβαια, για τις ανάγκες της τροποποιήθηκαν κατάλληλα, ώστε να τηρούνται οι προϋποθέσεις που ζητούνται.

Συνοπτική περιγραφή κώδικα

Αρχικά, πριν αναλυθεί ο κώδικας καλό θα ήταν να ειπωθεί πρώτα η έννοια της καθολικής ουράς εργασιών καθώς και το τι περιέχει. Πρόκειται για έναν πίνακα μεγέθους “**QUEUE_SIZE**” οποίος περιέχει αντικείμενα τύπου “**struct message**”. Αυτός ο τύπος δεδομένων περιέχει τρεις μεταβλητές οι οποίες είναι οι εξής: αριθμός τύπου μηνύματος (**type**), αριθμός αρχικής θέση πίνακα (**start_pos**) και αριθμός τελικής θέσης πίνακα (**end_pos**). Τα είδη μηνυμάτων που μπορεί να λάβει η μεταβλητή **type** είναι τρία. Συγκεκριμένα, τα μηνύματα “**WORK**” για την επεξεργασία του πακέτου, “**FINISH**” για την ολοκλήρωση του πακέτου από το “main thread” του προγράμματος και συμβαίνει όταν έχει γίνει επιτυχής ταξινόμηση για το τρέχον πακέτου και “**SHUTDOWN**” για την ειδοποίηση στο thread ώστε να πάψει την εκτέλεσή του. Για να γίνει προσθήκη και αφαίρεση των μηνυμάτων από την ουρά χρησιμοποιούνται οι συναρτήσεις “**send**” και “**recv**” αντίστοιχα. Σε αυτές έχουν ενταχθεί δομές όπως mutex και conditional variables για την αποφυγή συγχρονισμού και εργασιών μεταξύ των thread.

Αφού, έγινε η προηγούμενη αποσαφήνιση μπορούν να αναλυθούν οι λειτουργίες του main thread καθώς και αυτές των thread που αναλαμβάνουν την ταξινόμηση.

Η συνάρτηση “**thread_func**” περιέχει τη βασική λειτουργικότητα που εκτελεί κάθε thread που έχει οριστεί στο πρόγραμμα. Το thread ανακτά τον πίνακα προς ταξινόμηση και στη συνέχεια αρχίζει την σάρωση του ουράς εργασιών για διαθέσιμα πακέτα εργασίας. Οι δυνατότητες που παρέχει είναι τρεις, όσες και τα είδη μηνυμάτων στο πρόγραμμα. Στη περίπτωση που ληφθεί ένα μήνυμα “**SHUTDOWN**” το πακέτο προστίθεται στην ουρά εργασίας, γίνεται έξοδος από το thread και παύει η λειτουργία του, Εάν, ληφθεί το μήνυμα “**FINISH**” τότε το πακέτο προστίθεται στην ουρά εργασίας έτσι ώστε αργότερα να το επεξεργαστεί το main thread. Τέλος, με το μήνυμα “**WORK**” διαλέγεται το πακέτο για την ταξινόμηση των στοιχείων του πίνακα από θέσεις “**start_pos**” και “**end_pos**” που του έχουν οριστεί. Σε αυτό εάν το μήκος του πίνακα είναι μικρότερο από ένα κατώφλι (**CUTOFF**) τότε εκτελείται η μέθοδος ταξινόμησης insertionsort, διαφορετικά ο πίνακας χωρίζεται στα δύο και και αυτό με τη συνάρτηση “**partition**” που βγάζει ένα το στοιχείο pivot και στη συνέχεια τοποθετούνται οι δύο πίνακες που προέκυψαν από τα προηγούμενα όρια στην ουρά εργασίας.

Η συνάρτηση “**main**” δέχεται δύο παραμέτρους κατά την εκτέλεση του προγράμματος από το τερματικό. Αυτές είναι ο αριθμός του μεγέθους του πίνακα προ ταξινόμηση “**N**” και ο αριθμός των

thread για χρήση “**THREADS**”. Στη συνέχεια δημιουργεί έναν πίνακα μεγέθους όπως ορίστηκε από την είσοδο και η αρχικοποίηση του γίνεται ψευδοτυχαία με τη χρήση της συνάρτηση “**rand()**” και οι τιμές που ανατίθενται βρίσκονται στο διάστημα [0, 1]. Στη συνέχεια, γίνεται αρχικοποίηση των thread ανάλογα με το πλήθος τους με τη χρήση της συνάρτησης “**pthread_create**”. Το κάθε thread περιέχεται σε έναν πίνακα με όνομα “**threads**”. Για εκκίνηση της διαδικασίας δημιουργείται και εισάγεται ένα πακέτο με τύπο μηνύματος “**WORK**”, “**start_pos**” 0 το οποίο δηλώνει την αρχή και “**end_pos**” N το οποίο δηλώνει το τέλος του πίνακα ταξινόμησης.

Στο main thread κάθε φορά γίνεται ανάγνωση των πακέτοων από την ουρά ταξινόμησης. Στη περίπτωση που ληφθεί πακέτο με μήνυμα “**FINISH**” τότε αυξάνεται ένας μετρητής που δηλώνει το πλήθος των στοιχείων του πίνακα όπου έγινε ταξινόμηση. Διαφορετικά γίνεται προώθηση του μηνύματος. Η ολοκλήρωση της παραπάνω διαδικασίας θα γίνει όταν ο μετρητής γίνει ίσος με το πλήθος των στοιχείων του πίνακα, δηλαδή ίσο με “**N**”. Στη συνέχεια, στέλνεται το μήνυμα με τύπο “**SHUTDOWN**” για να σταματήσουν τα thread τη λειτουργία τους. Έπειτα με τη συνάρτηση pthread_join αναμένονται τα threads για να ολοκληρωθούν και αφού γίνει αυτό γίνεται ο έλεγχος στα στοιχεία του πίνακα για την απόδειξη του γεγονότος της σωστής ταξινόμησης. Τελικά, αποδεσμεύονται ο πίνακα ταξινόμησης καθώς και όλες οι δομές που δημιουργήθηκαν από το pthread.