Παράλληλος Προγραμματισμός 2019

Προγραμματιστική εργασία 2

Ονοματεπώνυμο : Βουτεράκος Γρηγόριος
ΑΜ : Π2013109

Στην εργαστηριακή εργασία που πραγματοποιήθηκε σε γλώσσα προγραμματισμού C υλοποιήθηκε ο αλγόριθμος Quicksort, χρησιμοποιώντας μία δεξαμενή από συγκεκριμένο αριθμό threads, ή thread pool. Τα threads αναλαμβάνουν πακέτα εργασίας αξιοποιώντας μία global ουρά εργασιών. Τόσο ο αριθμός των threads, όσο μέγεθος της ουράς εργασιών αλλά και το μέγεθος του πίνακα προς ταξινόμηση ορίζονται με define στην αρχή του κώδικα.Η ουρά μηνυμάτων αναπαρίσταται από έναν πίνακα μεγέθους Ν από μεταβλητές τύπου message, που είναι ένα struct με τρία διαφορετικά μέλη: τον τύπο του μηνύματος type, τη θέση αρχής start και τη θέση τέλους end. Υπάρχουν τρεις διαφορετικοί τύποι μηνύματος: 1. WORK, που χρησιμοποιείται για να δείξει ότι απαιτείται μία διαδικασία μεταξύ των θέσεων start και end. 2. DONE, που χρησιμοποιείται για την ενημέρωση main ότι έχουν ταξινομηθεί τα στοιχεία του πίνακα μεταξύ των θέσεων start και end. 3. SHUTDOWN, που λειτουργεί ως σήμα για την παύση της εκτέλεσης των threads. Για την προσθήκη μηνυμάτων από την ουρά χρησιμοποιείται η συνάρτηση send ενώ για την αφαίρεση μηνυμάτων η recv. Οι συναρτήσεις αυτές χρησιμοποιούν τις δομές mutex και conditional variables, ώστε να αποφεύγονται προβλήματα συγχρονισμού μεταξύ των διαφόρων threads.Το thread pool αποτελείται από έναν προκαθορισμένο αριθμό threads, που ορίζεται από τη σταθερά THREADS. Κατά τη δημιουργία τους, ως παράμετρος περνιέται ο πίνακας a προς ταξινόμηση, που έχει δημιουργηθεί δυναμικά από τη main. Στην αρχή της λειτουργίας του, κάθε thread ελέγχει την ουρά εργασιών για διαθέσιμα πακέτα, μέσω της συνάρτησης recv. Η συνάρτηση αυτή επιστρέφει με reference τα μέλη του struct που αναπαριστά τα πακέτα εργασίας, πιο συγκεκριμένα τον τύπο του μηνύματος, και δύο δείκτες start, end προς θέσεις του πίνακα, που χρησιμοποιούνται στις διάφορες διεργασίες, απαραίτητες για την ταξινόμηση. Η συνάρτηση recv, για να εξασφαλίσει ότι δεν θα προκύψουν θέματα συγχρονισμού, χρησιμοποιεί τις δομές mutex και conditional variable.Η συνάρτηση main αρχικά δεσμεύει μνήμη για τον πίνακα a προς ταξινόμηση, τύπου double και μεγέθους που καθορίζεται από τη σταθερά SIZE. Αν η δέσμευση δεν ολοκληρωθεί επιτυχώς, η εκτέλεση του προγράμματος τερματίζει. Ο πίνακας αρχικοποιείται με τυχαίες τιμές που προκύπτουν διαιρώντας το αποτέλεσμα της συνάρτησης rand με τη μέγιστη τιμή RAND_MAX που μπορεί αυτή να επιστρέψει. Στη συνέχεια δημιουργείται το thread pool, που αναπαρίσταται ως ένας πίνακας από μεταβλητές τύπου pthread_t, με μέγεθος THREADS. Μετά την ολοκλήρωση αυτού, αρχίζει ο επαναλαμβανόμενος έλεγχος της ουράς για μηνύματα. Η main περιμένει μέχρι να ολοκληρώσουν όλα τα threads, μέσω της συνάρτησης pthread_join και στη συνέχεια ελέγχει εάν υπήρξε κάποιο λάθος κατά την ταξινόμηση. Σε περίπτωση που συνέβη κάτι τέτοιο, εμφανίζεται σχετικό μήνυμα. Εναλλακτικά, εμφανίζεται μήνυμα επιτυχίας, αποδεσμεύονται οι πόροι που χρειάστηκαν και τελειώνει η εκτέλεση του προγράμματος.