

Παράλληλος Προγραμματισμός

Διακουμάκος Βασίλειος

Άσκηση 2

Σκοπός:

Ο σκοπός της εργασίας είναι η δημιουργία ενός προγράμματος σε C ,το οποίο θα δημιουργεί τον αλγόριθμο ταξινόμησης Quicksort χρησιμοποιώντας ένα thread pool.

Ουρά Μηνυμάτων

Αρχικά τα μηνύματα είναι struct μορφής με 3 πληροφορίες για το μήνυμα type = τύπος μηνύματος start = αρχική θέση end=τελική θέση.Επίσης ο τύπος του μηνύματος μπορεί να πάρει τις τιμές WORK (σήμα εκκίνησης) ,DONE (ότι έγινε η ταξινόμηση 'ενημερώνεται η main'),END (για τον τερματισμό μετά την ταξινόμηση). Ουρά μηνυμάτων msg_Queue[N] όπου N επίσης defined,οι συναρτήσεις msg_send και msg_rcv για προσθήκη και αφαίρεση μηνυμάτων στην ουρά.Επίσης με την χρήση mutex και condition variables αποφευχονται τα διάφορα προβλήματα συγχρονισμού.

Συναρτήσεις

Στην isort συνάρτηση γίνεται ταξινόμηση (insertion sort) ανάλογα με το (end-start)που θα πρέπει να είναι μικρότερο από το threshold που έχουμε κάνει define αλλιώς παίρνει σειρά η partition.Η partition είναι βοηθητική συνάρτηση της quicksort η οποία κάνει σπλιτ τον πίνακα ανάλογα με το pivot που ορίζουμε στο

κέντρο του πίνακα. Η quicksort χρησιμοποιεί τα αποτελέσματα της partition και εκτελείται αναδρομικά. Η thread_func καλείτε για το εκάστοτε thread, Ελέγχει τον τύπο των μηνυμάτων αν είναι DONE, WORK, END. Αν ο τύπος δεν είναι η done η end γίνεται η ταξινόμηση που προαναφέρθηκε ανάλογα με το threshold

MAIN()

Τέλος στην main γίνεται η αρχικοποίηση του πίνακα array με τυχαίες τιμές από την διαίρεση της συνάρτησης RAND με την RAND_MAX, γίνεται αρχικοποίηση της ουράς. Ακόμα δημιουργείται το thread pool (pool_t) και στην συνέχεια γίνεται έλεγχος για το αν έχει δημιουργηθεί σωστά το thread αν ναι συνεχίζει αλλιώς αποδεδμεύει τον πίνακα και σταματάει την εκτέλεση. Στην συνέχεια αφού σταλεί το πρώτο μήνυμα κρατάμε ένα counter για τα sorted στοιχεία, όταν το counter είναι ίσο με το μέγεθος στέλνεται το σήμα τερματισμού END και γίνεται αποδέσμευση των πόρων.