ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ – ΠΑΡΑΛΛΗΛΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΕΡΓΑΣΙΑ 2

Ονοματεπώνυμο: Τασιόπουλος Μανώλης

ΑΜ : Π2015046

Διδάσκοντας : Μιχάλης Στεφανιδάκης

Αριθμός σελίδων : 6

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΓΡΑΦΗ

QUICKSORT.C

-Το "quicksort.c" είναι ένα πρόγραμμα γραμμένο σε C κώδικα το οποίο υλοποιεί τον αλγόριθμο QuicSort() με διάφορες παραλλαγές, αξιοποιώντας τη βιβλιοθήκη <pthread.h>.

Αποτελείται από τα εξής χαρακτηριστικά:

- **Έναν πίνακα που περιέχει τα Threads**

  Τα threads αποθηκεύονται σε
  έναν πίνακα που ονομάζεται
  "thread pool" και αναλαμβάνουν
  την επεξεργασία διαφόρων
  πακέτων με τη χρήση μιας ουράς
  εργασιών

- → **Ουρά εργασιών**

  Η ουρά αυτή είναι υπεύθυνη
  για την προώθηση και την
  αποθήκευση των πακέτων
  εργασίας τα οποία επεξεργάζονται
  στα threads του thread poll

-Στη συνέχεια θα δούμε πιο αναλυτικά τον τρόπο με τον οποίο μπορούν να χρησιμοποιηθούν αυτά τα δύο χαρακτηριστικά για την ταξινόμηση ενός πίνακα Ν θέσεων

THREAD POOL

- Το thread_pool είναι μια μεταβλητή της συνάρτησης main() και θα χρησιμοποιηθεί για την αποθήκευση των thread που θα χρειαστούν για την ταξινόμηση του πίνακα.

Η λειτουργείες της μεταβλητής αυτής είναι οι εξής:

- Έχει μέγεθος που ορίζεται από τη σταθερά "**NUM_THREADS**"
- Δημιουργείται με τη συνάρτηση **pthread_create()**
- Ως ορίσματα περνιούνται το thread_pool, η συνάρτηση **threrad_func()** και ο πίνακας για την ταξινόμηση

-Η thread_func() είναι μία συνάρτηση η οποία επικοινωνεί με την main() και ελέγχει ποια πακέτα θα επεξεργαστούν και με ποιο τρόπο.

-Ξεκινώντας τα threads ελέγχουν την ουρά για πακέτα με τη χρήση της συνάρτησης receive(). Η συνάρτηση αυτή επιστρέφει:

- By reference έναν τύπο **struct** που έχουμε ορίσει για τα πακέτα μας
- Έναν δείκτη start για την αρχή του πίνακα
- Έναν δείκτη end για το τέλος του πίνακα

-Κατά τη λειτουργία της η receive():

- **Κλειδώνει** το mutex που είναι απαραίτητο για το συγχρονισμό
- Ελέγχει τον αριθμό των μηνυμάτων με τη μεταβλητή **msg_count**
- Αν δεν υπάρχει άλλο μήνυμα στην ουρά **ξεκλειδώνει** το mutex
- Περιμένει σήμα για το **conditional variable msg_in**
- Αν εντοπιστούν οι τιμές που του μηνύματος **αποθηκεύονται** στα ορίσματα
- Στέλνει σήμα msg_out και **ξεκλειδώνει** το mutex

-Υπάρχει περίπτωση να έχουν κολλήσει τα threads αν δεν υπάρχει αρκετός χώρος στην ούρα, οπότε και για αυτό ξεκλειδώνουμε τον mutex.

THREAD_FUNC()

-Η συνάρτηση "thread_func()" περιμένει να λάβει ένα σήμα μνήματος. Μόλις εντοπίσει ένα μήνυμα ελέγχει την μεταβλητή του struct που το αποτελεί και περιέχει τον τύπο του. Το πρόγραμμα αναγνωρίζει 3 τύπους μηνύματος [WORK = **0**, FINISH = **1**, STOP = **2**]

Αναλυτικότερα για τον τύπο:

- **WORK**: Για το μήκος του κάθε κομματιού πίνακα που λαμβάνει υπολογίζεται το μήκος του με τη διαφορά (end – start).

  → Αν είναι μικρότερο από ένα Threshold που ορίσαμε
  - Τότε ταξινομείται με τη συνάρτηση **Insertion Sort()**
  - Τοποθέτηση του τύπου σε **FINISH**

  → Αλλιώς
  - Ταξινομείται με τη συνάρτηση **Quick Sort()**
  - Διαχωρισμός του πίνακα στο **pivot** με τη συνάρτηση **partition()**
  - **Τοποθέτηση** στην ουρά τα πακέτα για τα δύο υποσύνολα

- **FINISH**: Αν το thread λάβει ένα τέτοιο μήνυμα τότε το προωθεί στην ουρά εργασίας ώστε να ενημερώσει το thread για την κατάσταση της εκτέλεσης

- **STOP**: Αν λάβει το thread αυτό το μήνυμα τότε σταματάει να ελέγχει για καινούργιους τύπους μηνύματος και το τοποθετεί στην ουρά προκειμένου να σταματήσει η εκτέλεση και να αρχίσει ο έλεγχος των μηνυμάτων

-Μόλις ολοκληρωθεί η διαδικασία του ελέγχου των τύπων των μηνυμάτων τότε η εκτέλεση του προγράμματος επιστρέφει στη συνάρτηση main() για να ολοκληρωθεί ο έλεγχος της σωστής ταξινόμησης του πίνακα

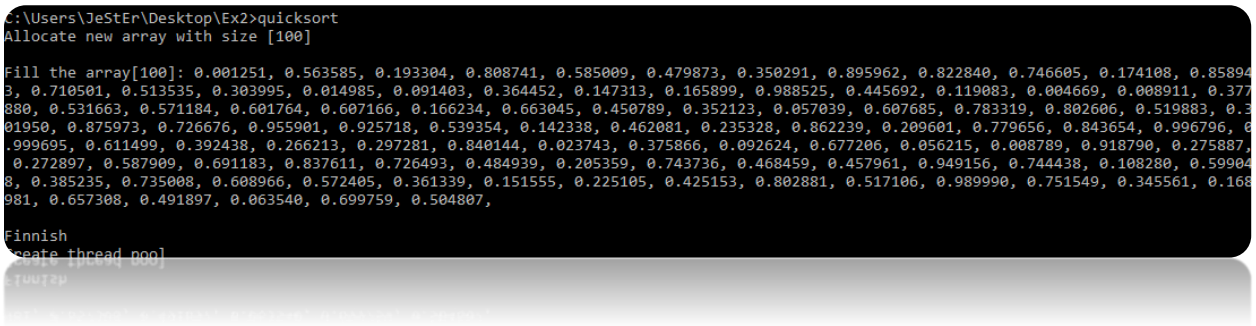
ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΕΞΟΔΟΙ

-Το πρόγραμμα "quicksort.c" μπορεί να ταξινομήσει έναν πίνακα Ν θέσεων με τη χρήση ενός αριθμού threads, μία ουρά εργασίας και τους αλγόριθμους **QuickSort** και **InsertionSort.**

Τα δεδομένα που χρησιμοποιούνται είναι:

1. **ΑΡΙΘΜΟΣ THREADS = 4**
2. **ΜΗΚΟΣ ΟΥΡΑΣ ΕΡΓΑΣΙΑΣ = 1000000**
3. **ΜΗΚΟΣ ΠΙΝΑΚΑ ΤΑΞΙΝΟΜΗΣΗΣ = 100**
4. **THRESHOLD = 10**

Δημιουργία πίνακα και προσθήκη τυχαίων τιμών



Πρόοδος των μηνυμάτων και έλεγχος για τη σωστή ταξινόμηση του πίνακα