

Παράλληλος προγραμματισμός 2018

Προγραμματιστική εργασία #2

Ονοματεπώνυμο: Ηλέκτρα Βλάχου
ΑΜ: Π2013073

Περιγραφή του κώδικα (Main Thread):

1. Δημιουργείται ένα σύνολο αριθμών με αριθμό στοιχείων `ARRAY_SIZE`.
2. Σε καθένα από τα στοιχεία αυτά δίνεται μια τυχαία τιμή.
3. Αρχικοποιείται η κυκλική ουρά εργασιών.
4. Δημιουργούνται `N` Worker Thread.
5. Στην ουρά στέλνεται μήνυμα τύπου `MESSAGE_PARTITION` για όλα τα στοιχεία του συνόλου αριθμών.
6. Η ουρά παρακολουθείται για μηνύματα τύπου `MESSAGE_DONE` και κρατείται ο αριθμός των ταξινομημένων στοιχείων.
7. Αν τα ταξινομημένα στοιχεία είναι ίσα με τον αριθμό όλων των στοιχείων, η ταξινόμηση έχει τελειώσει και στέλνεται μήνυμα τύπου `MESSAGE_SHUTDOWN`.
8. Όταν όλα τα Worker Thread τερματίσουν γίνεται έλεγχος για την ορθή ταξινόμηση των στοιχείων και αποδεσμεύονται όλες οι δυναμικές δομές.

Περιγραφή του κώδικα (Worker Thread):

1. Τα ελεύθερα Worker Thread διαβάζουν τα μηνύματα της ουράς.
2. Αναλογα του τύπου μηνύματος, το Worker Thread χωρίζει ένα τμήμα σε δύο μικρότερα, ταξινομεί ή τερματίζει τη λειτουργία του (βλ. παρακάτω).

Περιγραφή της ουράς μηνυμάτων:

Η ουρά μηνυμάτων είναι κυκλική (το προηγούμενο του πρώτου στοιχείου είναι το τελευταίο και το αντίθετο). Η ουρά αυτή αρχικοποιείται με τη συνάρτηση `QueueMake()` και έχει ως παράμετρο το μέγιστο αριθμό μηνυμάτων που μπορεί να περιέχει. Η διαγραφή της ουράς γίνεται με τη συνάρτηση `QueueDelete()`. Η εισαγωγή μηνύματος γίνεται με τη συνάρτηση `QueueInsert()` όπου το μήνυμα αποθηκεύεται στη θέση που δείχνει η μεταβλητή `front`. Η εξαγωγή γίνεται με τη συνάρτηση `QueuePop()` η οποία επιστρέφει το μήνυμα που βρίσκεται στη θέση που δείχνει η μεταβλητή `back`. Αν η ουρά είναι κενή το μήνυμα που επιστρέφεται είναι τύπου `MESSAGE_INVALID`.

Περιγραφή των τύπων μηνυμάτων:

MESSAGE_INVALID:

Αποστολέας: -

Παραλήπτης: Main Thread/Worker Thread

Το μήνυμα αυτό εμφανίζεται όταν κάποιο νήμα προσπαθεί να διαβάσει το επόμενο μήνυμα από την ουρά, αλλά η ουρά είναι κενή.

MESSAGE_PARTITION:

Αποστολέας: Main Thread/Worker Thread

Παραλήπτης: Worker Thread

Το Worker Thread που επεξεργάζεται αυτό το μήνυμα αναλαμβάνει να χωρίσει ένα κομμάτι του αρχικού πίνακα σε δύο μικρότερα. Ο δείκτης σε πίνακα που περιέχεται στο μήνυμα δείχνει προς το πρώτο στοιχείο του αρχικού κομματιού. Το μέγεθος περιέχεται στο πεδίο size. Για παράδειγμα, το Main Thread θα στείλει το μήνυμα αυτό για ολόκληρο τον πίνακα, δηλαδή με πεδία: {MESSAGE_PARTITION, numbers, arraySize}, ενώ στη συνέχεια τα Worker Thread θα στέλνουν το μήνυμα με πεδία: {MESSAGE_PARTITION, numbers+X, subArraySize (<arraySize)}. Εάν το subArraySize είναι μικρότερο του Cutoff τότε αντί το αρχικό κομμάτι να σπάσει σε δύο, στην ουρά μηνυμάτων εμφανίζεται το μήνυμα MESSAGE_QSORT (βλ. παρακάτω).

MESSAGE_QSORT:

Αποστολέας: Worker Thread

Παραλήπτης: Worker Thread

Το Worker Thread που επεξεργάζεται αυτό το μήνυμα αναλαμβάνει να ταξινομήσει το κομμάτι των αριθμών που περιέχεται στα πεδία του μηνύματος αυτού. Όταν τελειώσει η ταξινόμηση, στην ουρά μηνυμάτων εμφανίζεται το μήνυμα MESSAGE_DONE (βλ. παρακάτω).

MESSAGE_DONE:

Αποστολέας: Worker Thread

Παραλήπτης: Main Thread

Το μήνυμα αυτό δημιουργείται από τα Worker Thread και έχει προορισμό το Main Thread. Στο πεδίο size βρίσκεται ο αριθμός των στοιχείων που έχουν ταξινομηθεί. Το Main Thread μετράει τα στοιχεία που έχουν ταξινομηθεί και όταν αυτά γίνουν ίσα με τον αρχικό αριθμό στοιχείων (arraySize), τερματίζει όλα τα Worker Thread στέλνοντας μήνυμα τύπου MESSAGE_SHUTDOWN (βλ. παρακάτω).

MESSAGE_SHUTDOWN:

Αποστολέας: Main Thread

Παραλήπτης: Worker Thread

Το Worker Thread που επεξεργάζεται αυτό το μήνυμα τερματίζει τη λειτουργία του και ξαναβαάζει το μήνυμα αυτό στην ουρά.

Επειδη και το Main Thread και τα Worker Thread χρησιμοποιουν την ιδια ουρα μηνυματων, οταν καποιο μηνυμα διαβαστει απο ενα νημα για το οποιο δεν προοριζεται, πρεπει το νημα αυτο να ξαναεισαγει στο τελος της ουρας το μηνυμα αυτο.