

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

Υποχρεωτικό Μάθημα 4^{ου} εξαμήνου

Εαρινό Εξάμηνο 2016-2017

2^η Προγραμματιστική Εργασία

Αντικείμενο: Στα πλαίσια της 2ης εργασίας καλείστε να υλοποιήσετε σε C ένα πρόγραμμα που θα εφαρμόζει μια παραλλαγή του αλγορίθμου mergesort για μονοδιάστατους πίνακες ακεραίων αριθμών. Στη mergesort, ο αρχικός πίνακας προς ταξινόμηση διαιρείται σε υποπίνακες που ταξινομούνται ανεξάρτητα. Οι ταξινομημένοι υποπίνακες συγχωνεύονται (merge) και συνθέτουν τον ταξινομημένο αρχικό πίνακα. Η ανεξάρτητη ταξινόμηση των υποπινάκων καθιστά δυνατή την παράλληλη εκτέλεση των επιμέρους ταξινόμήσεων. Στο πρόγραμμά σας θα πρέπει να αξιοποιήσετε την βιβλιοθήκη pthread (POSIX thread) για να υλοποιήσετε πολλά νήματα που θα ταξινομούν παράλληλα υποδιαιρέσεις του αρχικού πίνακα (υποπίνακες). Ο κώδικάς σας θα πρέπει να τρέχει σωστά στην εικονική μηχανή που έχουμε κάνει διαθέσιμη στα CSLAB και στο διαδίκτυο (βλ. σχετική ανακοίνωση στο eclass).

Απαιτήσεις εργασίας:

- Το πρόγραμμά σας θα δέχεται ως είσοδο α) έναν ακέραιο n που θα παριστάνει το πλήθος των ακεραίων αριθμών που θα πρέπει να ταξινομηθούν σε αύξουσα σειρά, άρα και το μέγεθος του πίνακα που θα δημιουργήσετε, β) έναν ακέραιο s που θα χρησιμοποιείται ως σπόρος (seed) για την παραγωγή n τυχαίων αριθμών που θα τοποθετηθούν στον προς ταξινόμηση πίνακα, γ) έναν ακέραιο $t = 1, 2, 4$ ή 8 με το πλήθος των νημάτων που θα δημιουργούνται για την ταξινόμηση του πίνακα.
- Σε κάθε νήμα θα εκχωρείται ένα μέρος του πίνακα το οποίο θα ταξινομείται μόνο από το συγκεκριμένο νήμα με βάση κάποιον αλγόριθμο ταξινόμησης της επιλογής σας. Για παράδειγμα, για πίνακα 32 θέσεων με 2 νήματα, το 1^ο νήμα θα ταξινομεί τις θέσεις 0-15 και το 2^ο νήμα τις θέσεις 16-31. **Προσοχή:** Οι υποπίνακες είναι απλά υποπεριοχές του αρχικού πίνακα, συνεπώς κάθε νήμα θα εκτελεί την ταξινόμηση του απευθείας μέσα στον αρχικό πίνακα.
- Επιπλέον το κάθε νήμα θα έχει ένα μοναδικό αναγνωριστικό (thread id) το οποίο θα καθορίζεται από τη σειρά με την οποία έχει δημιουργηθεί. Για παράδειγμα το πρώτο νήμα θα έχει thread id = 1. Στο τέλος της εκτέλεσης του το κάθε νήμα θα εκτυπώνει όλους τους αριθμούς που ταξινόμησε. Η σειρά εμφάνισης των μηνυμάτων από τα νήματα θα πρέπει να γίνει σε φθίνουσα σειρά με βάση το thread id. Πιο συγκεκριμένα, πρώτα θα εκτυπώσει τους αριθμούς που ταξινόμησε το νήμα με thread id = t , στη συνέχεια το νήμα με thread id = $t - 1$ και ούτω καθεξής.
- Όταν όλα τα νήματα ολοκληρώσουν τις επί μέρους ταξινόμήσεις, το βασικό/αρχικό νήμα θα συγχωνεύει τις επιμέρους λύσεις, δηλαδή τους ταξινομημένους υποπίνακες, με βάση τον αλγόριθμο συγχώνευσης του mergesort. **Προσοχή:** Αν και ο αλγόριθμος mergesort είναι γενικά αναδρομικός, στη δική σας υλοποίηση υπάρχουν μόνο δύο βήματα, ταξινόμηση υποπινάκων και συγχώνευση όλων των υποπινάκων.
- Ο συγχρονισμός των νημάτων θα γίνεται με αντικείμενα pthread_mutex_t ή/και pthread_cond_t και τη μέθοδο pthread_join(), υλοποιώντας τρεις (3) πολιτικές:
 1. Απόλυτος συγχρονισμός: μόνο ένα νήμα μπορεί να διαβάζει και να γράφει στον πίνακα κάθε στιγμή. Αυτή η πολιτική εγγυάται την αποφυγή ασυνεπειών στη μνήμη, αλλά περιορίζει σημαντικά την παράλληλη επεξεργασία.
 2. Χαλαρός συγχρονισμός: μόνο ένα νήμα μπορεί να γράφει στον πίνακα κάθε στιγμή. Η πολιτική αυτή επιτρέπει σε πολλά νήματα να διαβάζουν ταυτόχρονα τη μνήμη αλλά μόνο σε ένα να την γράφει. Αυτή η πολιτική δεν εγγυάται την

αποφυγή ασυνεπειών, αλλά δεν δημιουργεί προβλήματα όταν μόνο ένα νήμα διαβάζει και γράφει σε κάθε θέση του πίνακα.

3. Απουσία συγχρονισμού: όλα τα νήματα διαβάζουν και γράφουν στον πίνακα ταυτόχρονα. Σκοπός αυτής της πολιτικής είναι η αύξηση της απόδοσης του προγράμματος μιας και τα νήματα πλέον δε μπλοκάρουν. Αυτή η πολιτική δεν εγγυάται την αποφυγή ασυνεπειών, αλλά δεν δημιουργεί προβλήματα όταν μόνο ένα νήμα διαβάζει και γράφει σε κάθε θέση του πίνακα.

Πρόσθετες απαιτήσεις:

- Το πρόγραμμά σας θα πρέπει να μεταγλωττίζεται με την εντολή `gcc p3xxxxxx-p3yyyyyy-p3zzzzzz-os_ex2 -o os_ex2 -g -pthread` όπου `p3xxxxxx-p3yyyyyy-p3zzzzzz` θα είναι οι αριθμοί μητρώων των φοιτητών της ομάδας.
- Το πρόγραμμά σας θα πρέπει να εκτελείται με την εντολή `./os_ex2 -numbers=X -threads=Y -seed=Z -mode=I` όπου `-numbers=X` δηλώνει το πλήθος των ακεραίων που θα ταξινομηθούν, `-threads=Y` τον αριθμό των νημάτων που θα χρησιμοποιηθούν (1, 2, 4 ή 8), `-seed=Z` τον ακέραιο για την παραγωγή τυχαίων αριθμών και `-mode=I` τον τύπο συγχρονισμού των νημάτων (1, 2 ή 3).
- Το πρόγραμμά σας θα εμφανίζει στο τερματικό διαγνωστικές πληροφορίες, όπως τον αριθμό κάθε νήματος και την περιοχή που έχει αναλάβει (π.χ. `Started thread 3 for sorting cells from 256 to 511`) και το συνολικό χρόνο εκτέλεσης του προγράμματος (π.χ. `Mergesort completed after 912345 nanoseconds`).
Προσοχή: Η έξοδος κάθε νήματος πρέπει να εμφανίζεται σε διαφορετική γραμμή, δηλαδή, δεν θα αναμιγνύονται μηνύματα διαφορετικών νημάτων στην ίδια γραμμή.
- Το πρόγραμμά σας θα γράφει σε ένα αρχείο με όνομα `results.dat` στο φάκελο που βρίσκεται το εκτελέσιμο αρχείο τον πίνακα ακεραίων πριν και μετά την ταξινόμηση (π.χ. `Initial array={12,3,5,...,34}`, `Sorted array={3,5,...,12,34}`).

Υποδείξεις:

- Για να βρείτε το χρόνο του συστήματος μπορείτε να χρησιμοποιήσετε τη συνάρτηση `gettimeofday()`. Καλώντας τη συνάρτηση αυτή στην αρχή και στο τέλος του προγράμματος μπορείτε να υπολογίσετε τον πραγματικό χρόνο εκτέλεσης του προγράμματος. **Προσοχή:** Ο χρόνος αυτός δεν ταυτίζεται με το χρόνο επεξεργασίας του προγράμματος, γιατί εξαρτάται από το γενικότερο φόρτο του συστήματος.
- Για να κατασκευάσετε τυχαίους αριθμούς, θα χρησιμοποιήσετε αρχικά τη συνάρτηση `srand()` με παράμετρο τον σπόρο που έχει επιλέξει ο χρήστης και στη συνέχεια θα καλέσετε όσες φορές χρειάζεται τη συνάρτηση `rand()` για να σας δώσει μία σειρά τυχαίων αριθμών που θα χρησιμοποιηθούν για την ταξινόμηση.

Παραδοτέα: Ο κώδικάς σας πρέπει να αποτελείται από ένα αρχείο με δηλώσεις και ένα αρχείο με κώδικα C. Τα αρχεία αυτά πρέπει να έχουν ονόματα της μορφής `p3xxxxxx-p3yyyyyy-p3zzzzzz-os_ex2.h` για τις δηλώσεις και `p3xxxxxx-p3yyyyyy-p3zzzzzz-os_ex2.c`, όπου `p3xxxxxx-p3yyyyyy-p3zzzzzz` είναι οι αριθμοί μητρώου σας. Εκτός από τον κώδικα, θα πρέπει να γράψετε μία αναφορά η οποία να περιγράφει το σύστημα που χρησιμοποιήσατε (εικονικό ΛΣ ή όχι, τύπος ΛΣ και αριθμός πυρήνων), προβλήματα που μπορεί να αντιμετωπίσατε και πώς τα διαχειριστήκατε. Η αναφορά σας θα περιέχει επίσης τους μέσους χρόνους εκτέλεσης από 3 επαναλήψεις του προγράμματός σας με 65536 αριθμούς για κάθε έναν από τους 12 δυνατούς συνδυασμούς πλήθους νημάτων και τύπων συγχρονισμού. Η αναφορά πρέπει να είναι ένα αρχείο σε μορφή PDF με όνομα της μορφής `p3xxxxxx-p3yyyyyy-p3zzzzzz-os_ex2.PDF`. Τα τρία αυτά αρχεία θα πρέπει να συμπιεστούν σε ένα αρχείο σε μορφή 7zip με όνομα της μορφής `p3xxxxxx-p3yyyyyy-p3zzzzzz-os_ex2.7z` και να υποβληθούν από ένα μόνο μέλος της ομάδας μέσω της υποβολής εργασιών του `eclass`.

Προθεσμία υποβολής: Τα συμπιεσμένα αρχεία με τις εργασίες σας θα πρέπει να παραδοθούν μέσω του `eclass` μέχρι την Τετάρτη 13/6/2017 και ώρα 23:59.

Βαθμολόγηση και εξέταση: Αρχεία που δεν θα έχουν την ονοματολογία που αναφέρεται παραπάνω δεν θα βαθμολογηθούν. Εργασίες οι οποίες έχουν ομοιότητες που υποδεικνύουν αντιγραφή θα μηδενιστούν όλες. Οι ομάδες σας θα κληθούν για επίδειξη των εργασιών και προφορική εξέταση στα CSLAB. Μέλη ομάδων, ή και ολόκληρες ομάδες, που δεν θα παρευρεθούν στην προφορική εξέταση, θα μηδενιστούν.