

ΕΡΓΑΣΤΗΡΙΟ ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ II

Εαρινό Εξάμηνο 2020-21 - Άσκηση #2

Ημ. Παράδοσης: 30/5/2021

A.

Γράψτε ένα πρόγραμμα σε C και με χρήση της βιβλιοθήκης των Pthreads, η εκτέλεση του οποίου θα έχει ως αποτέλεσμα να τυπώνεται επαναληπτικά η ακολουθία:

`<one><two><three><one><two><three><one><two><three>.....`

Θα πρέπει (για να πετύχετε το παραπάνω) να εκκινήσετε στο πρόγραμμά σας τρία (3) διαφορετικά threads (το ένα να τυπώνει `<one>`, το άλλο να τυπώνει `<two>` και το τρίτο να τυπώνει `<three>`), και στη συνέχεια να τα συγχρονίσετε κατάλληλα μεταξύ τους. Για τον απαιτούμενο συγχρονισμό θα πρέπει να χρησιμοποιήσετε **σημαφόρους**.

/ στη συνέχεια προσπαθήστε να δώσετε (δεν ζητείται υποχρεωτικά) μια άλλη έκδοση του παραπάνω προγράμματος, στο οποίο για τον απαιτούμενο συγχρονισμό να χρησιμοποιήσετε **condition variables** (η προσπάθειά σας για την απαίτηση αυτή θα συνυπολογιστεί προσθετικά στον τελικό βαθμό της άσκησης – θα μετρήσει δηλαδή ως 'bonus') */*

B.

Έστω ένας δυσδιάστατος πίνακας ακεραίων A (NxN). Γράψτε ένα πρόγραμμα το οποίο θα βρίσκει το 'μέγιστο' στοιχείο (*m*) του πίνακα A [με τη βοήθεια 'p' threads όπου το κάθε thread θα υπολογίζει το επιμέρους μέγιστο N/p γραμμών του πίνακα – τα 'p', 'N', 'A' θα πρέπει να τα δίνει ο χρήστης ή να τα διαβάζετε από αρχείο – επίσης θεωρείστε ότι το 'N' είναι ακέραιο πολλαπλάσιο του 'p']. Στη συνέχεια το πρόγραμμα θα πρέπει να φτιάχνει επίσης παράλληλα (με τη βοήθεια των 'p' threads που έχουν εξαρχής δημιουργηθεί) ένα νέο πίνακα D (NxN) (τον οποίον θα τυπώνει στο τέλος στην οθόνη), στον οποίον θα αποτυπώνεται η 'απόσταση' του κάθε στοιχείου (A_{ij}) του πίνακα A από το *m*:

$$D_{ij} = m - A_{ij}$$

[χρησιμοποιήστε μεταξύ των άλλων για τον απαιτούμενο συγχρονισμό το μηχανισμό 'barrier synchronization' των Pthreads - συμβουλευτείτε σχετικά το παράδειγμα 'bar-ex.c' από το αρχείο THREADS-EXAMPLES.zip (αναρτημένο στο Eclass), καθώς και το ακόλουθο URL: http://pubs.opengroup.org/onlinepubs/009695399/functions/pthread_barrier_wait.html]

Γ.

Γράψτε δύο προγράμματα, ένα πρόγραμμα **server** και έναν πρόγραμμα **client** (το οποίο θα μπορούν να το εκτελούν εν δυνάμει πολλοί clients - και να μπορούν να εξυπηρετούνται ταυτόχρονα από το server), τα οποία θα μπορούν να επικοινωνούν μεταξύ τους (με χρήση UNIX-domain stream sockets) επαναληπτικά ως εξής:

- Ο client θα στέλνει στο server μία ακολουθία ακεραίων μήκους N, την οποία θα διαβάζει από το χρήστη.
- Ο server αφού παραλάβει την ακολουθία θα υπολογίζει το μέσο όρο της και αν αυτός είναι πάνω από 10 θα στέλνει πίσω στον client τόσο (α) τον μέσο όρο που βρήκε όσο και (β) ένα κατάλληλο μήνυμα αποδοχής (π.χ. 'Sequence Ok'). Ειδάλλως θα στέλνει μόνο ένα μήνυμα αποτυχίας (π.χ. 'Check Failed').
- Ο client θα πρέπει απλά μετά από κάθε επικοινωνία να τυπώνει ότι απάντηση του έστειλε ο server στην οθόνη, και να ζητά την επόμενη ακολουθία από το χρήστη. Η επικοινωνία θα τελειώνει (από πλευράς του client) όταν ο χρήστης δηλώσει ότι δεν επιθυμεί να δώσει άλλη ακολουθία προς έλεγχο.
- Θα πρέπει τέλος επίσης ο server να 'διατηρεί' κάπου (πάντα ενημερωμένο) το συνολικό αριθμό των αιτήσεων/ακολουθιών που έχουν περάσει επιτυχώς τον έλεγχο (πόσες δηλαδή μέχρι αυτή τη στιγμή - από όλους τους clients), καθώς και πόσες έχουν ελεγχθεί συνολικά.

Παραδοτέα:

Κώδικας, σχολιασμός/τεκμηρίωση, ενδεικτικά τρεξίματα και χρόνοι εκτέλεσης (ερώτημα B). Σχετικά με τη δομή και το format του/ων παραδοτέου/ων σας καλείστε να ακολουθήσετε τις οδηγίες που θα σας δώσει πιο συγκεκριμένα ο Καθηγητής του τμήματος σας.