

---

# ΕΡΓΑΣΤΗΡΙΟ ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ II

## Άσκηση #2 - 2021-22 (20%)

Ημ. Παράδοσης: 29/5/2022

---

**A.** Γράψτε ένα πρόγραμμα σε C το οποίο να υπολογίζει «παράλληλα» το εσωτερικό γινόμενο δύο διανυσμάτων A και B ( $a_1b_1+a_2b_2+\dots+a_nb_n$ ) [με τη βοήθεια 'p' threads όπου το κάθε thread θα υπολογίζει το επιμέρους άθροισμα 'n/p' όρων (επιμέρους γινομένων) – τα p, n,  $a_i$  και  $b_i$  θα πρέπει να τα δίνει ο χρήστης ή να τα διαβάζετε από αρχείο – επίσης θεωρείστε ότι το 'n' είναι ακέραιο πολλαπλάσιο του 'p']. Πιο συγκεκριμένα, το κάθε thread θα πρέπει (α) να υπολογίζει τοπικά (σε μια τοπική μεταβλητή – π.χ. local\_sum) το επιμέρους άθροισμα που του αναλογεί και (β) να ενημερώνει στο τέλος με το τοπικό του άθροισμα μια κοινή μεταβλητή (π.χ. total\_sum) η οποία θα αντιπροσωπεύει το τελικό άθροισμα (εσωτερικό γινόμενο). Το τελικό αυτό άθροισμα θα πρέπει στο τέλος να τυπώνεται στην οθόνη.

Υπάρχει κάποια «κρίσιμο τμήμα» στον κώδικά σας που απαιτεί προστασία? Αν ναι, γράψτε επιπλέον τον κώδικα για την προστασία του κρίσιμου αυτού τμήματος.

Δοκιμάστε να τρέξετε το πρόγραμμά σας για μεγάλα και πολύ μεγάλα 'n' (χρησιμοποιώντας για input αντίστοιχα διανύσματα αρχικοποιημένα με τυχαίους αριθμούς) και μετρήστε τους χρόνους εκτέλεσης για ένα και περισσότερα από ένα threads (π.χ. 1, 2, 4 και 8 threads).

**B.** Γράψτε ένα πρόγραμμα σε C, το οποίο θα χρησιμοποιεί την βιβλιοθήκη των POSIX Threads, και θα έχει ως αποτέλεσμα να τυπώνεται επαναληπτικά η ακολουθία:

*What A Wonderful World! What A Wonderful World! What A Wonderful World! ...*

Για να πετύχετε το παραπάνω θα πρέπει να εκκινήσετε στο πρόγραμμά σας τρία (3) διαφορετικά threads (το ένα να τυπώνει "What A ", το άλλο να τυπώνει "Wonderful " και το τρίτο να τυπώνει "World! "), και στη συνέχεια να τα συγχρονίσετε κατάλληλα μεταξύ τους χρησιμοποιώντας **σημαφόρους** (POSIX semaphores).

Στη συνέχεια προσπαθήστε να δώσετε (δεν ζητείται υποχρεωτικά) μια εναλλακτική έκδοση του προγράμματός σας χρησιμοποιώντας για τον απαιτούμενο συγχρονισμό **condition variables** (η προσπάθειά σας για την απαίτηση αυτή θα συνυπολογιστεί προσθετικά στον τελικό βαθμό της άσκησης – θα μετρήσει δηλαδή ως 'bonus').

**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 (είτε απευθείας από το πληκτρολόγιο είτε μέσω ενός αρχείου που αυτός θα υποδείξει).
- Ο server αφού παραλάβει την ακολουθία θα ελέγχει αν οι αριθμοί που περιέχονται σε αυτήν είναι οι N πρώτοι αριθμοί της ακολουθίας *Fibonacci*, και θα στέλνει πίσω στον client ένα κατάλληλο μήνυμα επιτυχίας/αποδοχής της ακολουθίας (π.χ. 'Sequence Ok') ή αποτυχίας (π.χ. 'Check Failed').
- Ο client θα πρέπει απλά μετά από κάθε επικοινωνία να τυπώνει την απάντηση που 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.

### Παραδοτέα:

Κώδικας, σχολιασμός/τεκμηρίωση, ενδεικτικά τρεξίματα.

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.