

**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**Άσκηση 4 “Προγραμματισμός Συστήματος” 2007-08 (Προθεσμία 26/01 – 18:00)**

**Στόχος:** Εξοικείωση με τον πολυνηματικό προγραμματισμό και τη χρήση δυαδικών σηματοφορέων (mutexes) και μεταβλητών συνθήκης (condition variables).

**ΠΡΟΣΟΧΗ: Η ΑΣΚΗΣΗ ΑΥΤΗ ΕΙΝΑΙ ΑΤΟΜΙΚΗ.**

Ζητούμενο είναι να δημιουργήσετε ένα πρόγραμμα παράλληλου υπολογισμού συναρτήσεων χρησιμοποιώντας νήματα και μεθόδους συγχρονισμού τους. Στο πρόγραμμά σας θα υπάρχουν 2 είδη νημάτων: νήματα παραγωγού και νήματα καταναλωτές. Οι καταναλωτές θα υπολογίζουν το άθροισμα όλων των ακεραίων αριθμών σε μία σειρά από διαστήματα  $[α,β]$  που θα παρέχουν οι παραγωγοί. Οι παραγωγοί θα διαβάζουν αυτά τα διαστήματα από αρχεία (ένα ανά παραγωγό) **και θα τα μεταφέρουν σε μία συνδεδεμένη λίστα απεριόριστου μεγέθους που θα προσπελαύνεται από όλα τα νήματα.** Οι καταναλωτές θα διαβάζουν τα διαστήματα από αυτή τη λίστα, και θα αποδεσμεύουν το χώρο των διαστημάτων που διάβασαν.

**Νήμα Παραγωγού**

Το k-νήμα παραγωγού ανοίγει το αρχείο `argv[1].k` όπου `argv[1]` δηλώνει το πρώτο όρισμα του προγράμματος. Κάθε τέτοιο αρχείο έχει την ακόλουθη μορφή: Η πρώτη γραμμή του περιγράφει πόσες γραμμές  $X$  με διαστήματα έχει το αρχείο (επεξηγείται παρακάτω). Στη συνέχεια ακολουθούν  $X$  γραμμές, καθεμία από τις οποίες περιέχει 2 θετικούς ακεραίους. Το κάθε ζεύγος ακεραίων  $α,β$  υποδηλώνει την αρχή και το τέλος του διαστήματος τιμών  $[α,β]$  (δηλαδή τα  $α,β$  **συμπεριλαμβάνονται** στο διάστημα και ΠΑΝΤΑ  $α \leq β$ ). Ένα παράδειγμα τέτοιου αρχείου είναι το:

```
3
1 2
24 36
152 152
```

Κάθε παραγωγός διαβάζει μία-μία τις γραμμές, δημιουργεί τα διαστήματα και τα εισάγει **στο τέλος της κοινώς προσπελάσιμης συνδεδεμένης λίστας** (θα υπάρξει σημαντική ποινή σε όσους δεν εισάγουν τα δεδομένα στο τέλος της λίστας). Όταν τελειώσει να διαβάζει το αρχείο του, κάθε παραγωγός κάνει ότι λειτουργίες απαιτούνται (πχ αποδέσμευση πόρων), και τερματίζει ΧΩΡΙΣ να περιμένει κανένα άλλο νήμα.

**Νήματα καταναλωτή**

**Ο κάθε καταναλωτής διαβάζει διαστήματα από την αρχή της συνδεδεμένης λίστας** (θα υπάρξει σημαντική ποινή σε όσους δεν διαβάζουν από την αρχή της λίστας) και υπολογίζει το άθροισμα όλων των ακεραίων αριθμών μέσα στο κλειστό αυτό διάστημα. **Για κάθε διάστημα που διαβάζει, το νήμα αφαιρεί το αντίστοιχο στοιχείο από τη συνδεδεμένη λίστα (και αποδεσμεύει το χώρο του),** για να μην υπάρξει πρόβλημα μνήμης από το μεγάλο αριθμό διαστημάτων που θα υπάρχουν.

Το κάθε νήμα καταναλωτή πρέπει να μπορέσει να καταλάβει πότε τερμάτισαν τη λειτουργία τους τα νήματα παραγωγού, και δεν υπάρχουν άλλα διαστήματα για επεξεργασία. Σε αυτή την περίπτωση, προσθέτει το συνολικό άθροισμα που έχει υπολογίσει για όλα τα διαστήματα

που επεξεργάστηκε σε μία κοινή μεταβλητή, η οποία τυπώνεται **μόνο** αφού όλα τα νήματα καταναλωτή έχουν κάνει αυτή τη λειτουργία (πχ, μπορεί να την τυπώνει το αρχικό νήμα του προγράμματος).

## Παράμετροι προγράμματος

Όπως εξηγήσαμε, το κύριο πρόγραμμα parallelSum θα παίρνει ως παραμέτρους το πρόθεμα των αρχείων εισόδου για τα νήματα παραγωγούς, τον αριθμό των νημάτων παραγωγών και τον αριθμό των νημάτων καταναλωτή. Στο τέλος θα τυπώνει μόνο το συνολικό άθροισμα που υπολόγισε.

## Χρήσιμα εργαλεία

Τα κεφάλαια 16.1-16.6 (εκτός του κεφαλαίου 16.3) του ebook περιγράφουν ένα πολύ συγγενές πρόβλημα με κυκλικό buffer αντί για συνδεδεμένη λίστα και ένα παραγωγό αντί για πολλούς. Προφανώς, θα βρείτε πολλές ομοιότητες με αυτή την άσκηση.

## Πράγματα προς σκέψη

- Προσέξτε τη συγχρονισμένη εισαγωγή/διαγραφή στοιχείων από τη συνδεδεμένη λίστα. Λάθη συχνά θα οδηγήσουν στη λήψη segmentation fault. Σκεφτείτε ότι η λίστα μπορεί να περιέχει 0 (head=tail=NULL), 1 (head=tail) ή περισσότερα (head <> tail) αντικείμενα, **σε κάθε λειτουργία (ανάγνωση/εισαγωγή)**. Σκοπός είναι η εισαγωγή και διαγραφή στοιχείων να γίνεται παράλληλα. Αν επιτρέπεται μόνο διαγραφή ή μόνο εισαγωγή κάθε φορά, θα υπάρξει επίπτωση στη βαθμολογία. Επίσης είναι αποδεκτό κάθε παραγωγός (καταναλωτής) να εισάγει (διαβάζει) **ένα μόνο διάστημα τιμών** κάθε φορά που αποκτάει έλεγχο στη λίστα. Πχ, μην υλοποιήσει κάποιος μία τεχνική όπου ο παραγωγός αποκτάει πλήρη έλεγχο, εισάγει όλα τα διαστήματα του αρχείου του, και σε αυτό το διάστημα κανένας άλλος δεν μπορεί να εισάγει/διαβάσει δεδομένα από τη λίστα.
- Προσέξτε πώς θα καταλάβουν τα νήματα καταναλωτές ότι έχουν τερματίσει τα νήματα παραγωγού. Το κεφάλαιο 16.6 σίγουρα θα σας βοηθήσει.
- Προσέξτε να συνυπολογίσετε τα αποτελέσματα όλων των νημάτων.

Σαν συμβουλή, κάνετε ελέγχους με πολλά νήματα και μεγάλα αρχεία. Στον έλεγχο θα έχω δημιουργήσει αρχεία με εκατοντάδες χιλιάδες γραμμές (προφανώς με τρόπο που να ξέρω το σωστό άθροισμα), και θα ζητήσω τη δημιουργία πολλών νημάτων ανά τύπο. Κάντε ελέγχους με το valgrind (πέραν από προβλήματα στην προσπέλαση μνήμης, μπορεί να σας ενημερώσει και για μνήμη που δημιουργήσατε και δεν αποδεσμεύσατε), και δημιουργείστε κάποιο απλό script (ή ένα απλό C πρόγραμμα) για τη δημιουργία των αρχείων εισόδου (1-2 βρόχοι for είναι αρκετοί για το script).

Όπως πάντα, θα πρέπει να παραδώσετε ένα tar αρχείο που να περιέχει README σε greeklish, Makefile, και όλα τα αρχεία του κώδικα (\*.h και \*.c). ΜΗΝ βάζετε στο tar αρχείο εκτελέσιμα. Και φυσικά να έχετε αφαιρέσει όλα τα δικαιώματα στο tar αρχείο από όλους τους υπόλοιπους χρήστες. Η πολιτική καθυστερημένης παράδοσης θα είναι όπως είχε αναφερθεί στο πρώτο μάθημα.

Αντώνιος Δεληγιαννάκης