



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ  
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ  
<http://www.cslab.ece.ntua.gr>

## Συστήματα Παράλληλης Επεξεργασίας

9ο εξάμηνο, Ακαδημαϊκή περίοδος 2009-2010

### Άσκηση 2: Συγχρονισμός Νημάτων

#### 1 Κλειδώματα Αναγνωστών-Εγγραφέων

Τα κλειδώματα Αναγνωστών-Εγγραφέων (read-write locks) είναι μηχανισμός συγχρονισμού που διακρίνει δύο είδη προσβάσεων: *ανάγνωση* και *εγγραφή*. Οι κανόνες συγχρονισμού για το συγκεκριμένο κλειδωμα προβλέπουν ότι στο κρίσιμο τμήμα μπορούν να βρίσκονται:

- Είτε **πολλαπλά** νήματα που πραγματοποιούν αναγνώσεις,
- Είτε **ένα και μοναδικό** νήμα εγγραφής.

Ζητείται να υλοποιηθούν δύο κλειδώματα Αναγνωστών-Εγγραφέων χρησιμοποιώντας ενεργό αναμονή (busy-wait)

**Υλοποίηση `rwspin`** Η υλοποίηση χρησιμοποιεί μια κοινή μεταβλητή (τύπου ακέραιου) και ατομικές εντολές. Η ατομική μεταβλητή αρχικοποιείται στην τιμή 0x01000000. Για το κλειδωμα ανάγνωσης η μεταβλητή μειώνεται κατά 1 και αν παραμένει θετική, το νήμα μπορεί να συνεχίσει στο κρίσιμο τμήμα. Για το κλειδωμα εγγραφής αφαιρείται από τη μεταβλητή η τιμή αρχικοποίησης και αν η νέα τιμή είναι 0, το νήμα συνεχίζει στο κρίσιμο τμήμα.

**Υλοποίηση `brlock`** Η υλοποίηση βασίζεται σε πολλαπλά κλειδώματα ενεργού αναμονής (spinlocks) – ένα για κάθε νήμα. Για το κλειδωμα ανάγνωσης το νήμα “παίρνει” το κλειδωμα που του αντιστοιχεί, ενώ για το κλειδωμα εγγραφής το νήμα θα πρέπει να “πάρει” όλα τα διαθέσιμα κλειδώματα.

Ζητείται να υλοποιηθεί ένα πρόγραμμα αξιολόγησης επίδοσης για κλειδώματα Αναγνωστών-Εγγραφέων.

Το πρόγραμμα θα δέχεται τις παρακάτω παραμέτρους:

- Αριθμός νημάτων.
- Αριθμός πράξεων που θα πραγματοποιηθούν.

- Πιθανότητα η πράξη να είναι ανάγνωση.
- Αριθμός των επαναλήψεων εντός του κρίσιμου τμήματος.
- Μέγιστος αριθμός επαναλήψεων εκτός του κρίσιμου τμήματος.

Το κάθε νήμα θα πραγματοποιεί μια σειρά από πράξεις – είτε εγγραφές, είτε αναγνώσεις – ανάλογα με την αντίστοιχη πιθανότητα. Για να προσομοιωθεί η συμπεριφορά εφαρμογών το κάθε νήμα θα πραγματοποιεί άσκοπες επαναλήψεις (idle loop) εντός και εκτός του κρίσιμου τμήματος. Οι επαναλήψεις εντός κρίσιμου τμήματος είναι σταθερές, ενώ οι επαναλήψεις εκτός κρίσιμου τμήματος είναι από 0, έως μια μέγιστη τιμή. Η έξοδος του προγράμματος θα είναι ο *συνολικός χρόνος εκτέλεσης*.

### Ζητήματα Υλοποίησης

- Ως αφετηρία για την άσκηση δίνεται κώδικας που περιέχει χρήσιμες συναρτήσεις, καθώς και έναν προτεινόμενο σκελετό υλοποίησης<sup>1</sup> (*/home/parallel/lockperf*).
- Χρησιμοποιήστε τις ατομικές εντολές και τους αντίστοιχους τύπους που δίνονται (*processor.h*).
- Χρησιμοποιήστε τη συνάρτηση *relax\_cpu()* στο εσωτερικό των βρόχων ενεργού αναμονής (*processor.h*).
- Χρησιμοποιήστε τη συνάρτηση *rand\_r()* για την παραγωγή ψευδο-τυχαίων αριθμών από πολλαπλά νήματα (*man rand\_r* για περισσότερες λεπτομέρειες). Αρχικοποιήστε τη γεννήτρια με διαφορετικό αριθμό για κάθε νήμα.
- Χρησιμοποιήστε τη συνάρτηση *do\_loops()* για την υλοποίηση των άσκοπων επαναλήψεων (*processor.h*).
- Χρησιμοποιήστε τον μηχανισμό *barrier* που παρέχει η βιβλιοθήκη *threads*. (*man pthread\_barrier\_wait* και *man pthread\_barrier\_init*).
- Χρησιμοποιήστε τη συνάρτηση *pthread\_setaffinity\_np()* για να περιορίσετε την εκτέλεση των νημάτων σε συγκεκριμένο επεξεργαστή. (*man cpu\_set* και *man pthread\_setaffinity\_np*).
- Χρησιμοποιήστε τα *spinlocks* που παρέχει η βιβλιοθήκη *threads*. (*man pthread\_spin\_lock* και *man pthread\_spin\_init*).
- Θεωρήστε ότι το μέγεθος της *cacheline* του επεξεργαστή είναι 64 bytes.

## 2 Συγχρονισμός σε πίνακα κατακερματισμού

Θεωρούμε πίνακα κατακερματισμού (hash table), με τις παρακάτω λειτουργίες:

- Εισαγωγή ζεύγους τιμής-κλειδιού (*insert*).
- Αναζήτηση τιμής με βάση το κλειδί (*lookup*).
- Διαγραφή με βάση το κλειδί (*delete*).
- Ανταλλαγή τιμών δύο στοιχείων της δομής με βάση τα κλειδιά τους (*swap*).

Η δομή υλοποιείται χρησιμοποιώντας έναν πίνακα σταθερού μεγέθους, την *mod (%)* για συνάρτηση κατακερματισμού και λίστες για την αντιμετώπιση πιθανών συγκρούσεων (hash collisions).

Ζητείται να υλοποιηθεί ο κατάλληλος συγχρονισμός, ώστε οι παραπάνω λειτουργίες να είναι *ατομικές*, όταν η δομή χρησιμοποιείται από πολλαπλά νήματα. Ζητούνται δύο διαφορετικές υλοποιήσεις του συγχρονισμού:

- Κεντρικό κλείδωμα (big lock) – ένα κλείδωμα για όλη την δομή.

<sup>1</sup>Η κατανόηση του κώδικα θα βοηθήσει σημαντικά στην υλοποίηση των ζητούμενων της άσκησης.

- Πολλαπλά κλειδώματα (fine-grain locking) – ένα κλειδωμα ανά στοιχείο του πίνακα κατακερματισμού.

Επιπρόσθετα, **ζητείται** η δημιουργία προγράμματος για την αξιολόγηση της επίδοσης των σχημάτων συγχρονισμού. Το πρόγραμμα θα δημιουργεί νήματα που θα πραγματοποιούν έναν αριθμό από τις λειτουργίες της δομής (σε κάθε επανάληψη επιλέγεται *τυχαία* μια λειτουργία και οι παράμετροί της). Το πρόγραμμα θα δέχεται τις ακόλουθες παραμέτρους ως είσοδο:

- Αριθμός νημάτων.
- Αριθμός λειτουργιών που θα πραγματοποιηθούν.
- Μέγεθος πίνακα κατακερματισμού.
- Μέγιστο κλειδί για τις λειτουργίες του πίνακα.

Η έξοδος του προγράμματος θα είναι ο *συνολικός* χρόνος εκτέλεσης.

### Ζητήματα Υλοποίησης

- Ως αφετηρία για την άσκηση δίνεται κώδικας που περιέχει την υλοποίηση του πίνακα κατακερματισμού, καθώς και την σειριακή έκδοση του προγράμματος αξιολόγησης της επίδοσης. (*/home/parallel/hash*).
- Θα πρέπει να εξασφαλιστεί μέσω της μεθόδου συγχρονισμού ότι δεν υπάρχει πιθανότητα αδιεξόδου (deadlock).
- Χρησιμοποιήστε τα spinlocks που παρέχει η βιβλιοθήκη pthreads, για την υλοποίηση των κλειδωμάτων.
- Δείτε και τα ζητήματα υλοποίησης του προηγούμενου ερωτήματος. (πχ barriers, affinity, κλπ).

## 3 Χρονοδιάγραμμα

- **15/12/2009:** Ενδιάμεση αναφορά, που θα περιέχει:
  - Ψευδοκώδικα για τα κλειδώματα *rwspin* και *brlock*.
  - Σχολιασμό (πιθανών) ζητημάτων που προκύπτουν.
- **12/01/2010:** Επίδειξη Άσκησης.
- **Ημ/νία εξέτασης:** Τελική αναφορά.