



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

ΣΥΣΤΗΜΑΤΑ ΠΑΡΑΛΛΗΛΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ 9ο εξάμηνο ΗΜΜΥ, ακαδημαϊκό έτος 2017-18

ΑΣΚΗΣΗ 5

Θέματα Συγχρονισμού σε Σύγχρονα Πολυπύρνα Συστήματα

Προθεσμίες παράδοσης:
Παράδοση Τελικής Αναφοράς 21 Ιανουαρίου ¹

1 Σκοπός της άσκησης

Σκοπός της συγκεκριμένης άσκησης είναι η εξοικείωση με την εκτέλεση προγραμμάτων σε σύγχρονα πολυπύρνα συστήματα και η αξιολόγηση της επίδοσής τους. Συγκεκριμένα, θα εξετάσετε πώς κάποια χαρακτηριστικά της αρχιτεκτονικής του συστήματος επηρεάζουν την επίδοση των εφαρμογών που εκτελούνται σε αυτά και θα αξιολογήσετε διάφορους τρόπους υλοποίησης κλειδωμάτων για αμοιβαίο αποκλεισμό καθώς και διάφορες τακτικές συγχρονισμού για δομές δεδομένων.

2 Λογαριασμοί Τράπεζας

Στο πρώτο μέρος της άσκησης σας δίνεται ένα πολυνηματικό πρόγραμμα όπου κάθε νήμα εκτελεί ένα σύνολο πράξεων πάνω σε συγκεκριμένο στοιχείο ενός πίνακα που αντιπροσωπεύει τους λογαριασμούς των πελατών μίας τράπεζας. Το πρόγραμμα χρησιμοποιεί την βιβλιοθήκη Posix Threads (pthreads) για την δημιουργία και την διαχείριση πολλαπλών νημάτων. Ο σχετικός κώδικας βρίσκεται στο φάκελο /home/parallel/rps/2017-2018/a5/z1/. Παρέχεται και το αντίστοιχο Makefile με το οποίο μπορείτε να μεταγλωττίσετε την εφαρμογή. Με την μεταγλώττιση παράγεται το εκτελέσιμο accounts.

Για τη ρύθμιση του αριθμού των νημάτων η εφαρμογή accounts χρησιμοποιεί τη μεταβλητή περιβάλλοντος (environment variable) MT_CONF. Μέσω της μεταβλητής περιβάλλοντος MT_CONF ρυθμίζεται και ο αριθμός των νημάτων αλλά και οι πυρήνες στους οποίους θα εκτελεστούν τα πολλαπλά νήματα. Για παράδειγμα η εκτέλεση της παρακάτω εντολής θα δημιουργήσει 4 νήματα τα οποία θα εκτελεστούν στους πυρήνες 0, 8, 16, 24:

```
user@sandman:~$ MT_CONF=0,8,16,24 ./accounts
```

¹mail to: jimsiak@cslab.ece.ntua.gr, cc: goumas@cslab.ece.ntua.gr, filename: a5-parlabXX-final.pdf

2.1 Ερωτήσεις - Ζητούμενα

1. Υπάρχει ανάγκη για συγχρονισμό ανάμεσα στα νήματα της εφαρμογής;
2. Πώς περιμένετε να μεταβάλλεται η επίδοση της εφαρμογής καθώς αυξάνετε τον αριθμό των νημάτων;
3. Εκτελέστε την εφαρμογή με 1,2,4,8,16,32,64 νήματα χρησιμοποιώντας τις τιμές για την `MT_CONF` που δίνονται στον πίνακα 1. Δώστε ένα διάγραμμα όπου στον άξονα x θα είναι ο αριθμός των νημάτων και στον άξονα y το αντίστοιχο throughput. Το διάγραμμα θα περιέχει δύο καμπύλες, μία για κάθε εκτέλεση του πίνακα 1. Ποια είναι η συμπεριφορά της εφαρμογής για κάθε μία από τις δύο εκτελέσεις; Εξηγήστε αυτήν την συμπεριφορά και τις διαφορές ανάμεσα στις δύο εκτελέσεις.

Αριθμός Νημάτων	MT_CONF	
	Εκτέλεση 1	Εκτέλεση 2
1	0	0
2	0,1	0,8
4	0-3	0,8,16,24
8	0-7	0-1,8-9,16-17,24-25
16	0-7,32-39	0-3,8-11,16-19,24-27
32	0-15,32-47	0-31
64	0-63	0-63

Πίνακας 1: Οι τιμές της μεταβλητής `MT_CONF` για τις δύο εκτελέσεις της εφαρμογής `accounts`.

4. Η εφαρμογή έχει την συμπεριφορά που αναμένετε; Αν όχι, εξηγήστε γιατί συμβαίνει αυτό και προτείνετε μία λύση. Τροποποιήστε κατάλληλα τον κώδικα και δώστε και πάλι τα αντίστοιχα διαγράμματα για τις δύο εκτελέσεις.
Υπόδειξη: πώς αποθηκεύεται ο πίνακας με τους λογαριασμούς στα διάφορα επίπεδα της ιεραρχίας της μνήμης και τι προκαλεί αυτό ανάμεσα στα νήματα της εφαρμογής;

3 Αμοιβαίος Αποκλεισμός - Κλειδώματα

Στο δεύτερο μέρος της άσκησης θα υλοποιήσετε και θα αξιολογήσετε διαφορετικούς τρόπους υλοποίησης κλειδωμάτων για αμοιβαίο αποκλεισμό. Για τους σκοπούς της άσκησης το κρίσιμο τμήμα που προστατεύεται μέσω των κλειδωμάτων που θα αξιολογήσετε περιλαμβάνει την αναζήτηση τυχαίων στοιχείων σε μία ταξινομημένη συνδεδεμένη λίστα. Το μέγεθος της λίστας δίνεται σαν όρισμα στην εφαρμογή και καθορίζει και το μέγεθος του κρίσιμου τμήματος.

Μπορείτε να βρείτε τον αντίστοιχο κώδικα στο φάκελο `/home/parallel/pps/2017-2018/a5/z2/`. Σας δίνεται το αρχείο `main.c` το οποίο αναλαμβάνει τη δημιουργία και τη διαχείριση των νημάτων μέσω της βιβλιοθήκης `Pthreads`. Επίσης, δίνεται το αρχείο `ll_serial.h` στο οποίο παρέχεται η υλοποίηση μίας ταξινομημένης συνδεδεμένης λίστας. Τέλος, στα αρχεία της μορφής `<lock_type>_lock.c` υπάρχει σκελετός για τις υλοποιήσεις των κλειδωμάτων. Με τη μεταγλώττιση (`make`) παράγονται τα εκτελέσιμα `linked_list` και `test`. Το πρώτο είναι αυτό που θα χρησιμοποιήσετε για τα ζητούμενα της άσκησης ενώ το δεύτερο δίνεται για να μπορείτε να ελέγχετε την ορθότητα των κλειδωμάτων που θα υλοποιήσετε.

Συγκεκριμένα, θα μελετήσετε τα παρακάτω είδη κλειδωμάτων (με έντονα γράμματα αυτά που καλείστε να υλοποιήσετε μόνοι σας):

- *nosync_lock*: Η συγκεκριμένη υλοποίηση δεν παρέχει αμοιβαίο αποκλεισμό και θα χρησιμοποιηθεί ως άνω όριο για την αξιολόγηση της επίδοσης των υπόλοιπων κλειδωμάτων.
- *pthread_lock*: Η συγκεκριμένη υλοποίηση χρησιμοποιεί ένα από τα κλειδώματα που παρέχεται από την βιβλιοθήκη Pthreads (*pthread_spinlock_t*).
- *tas_lock*: Το *test-and-set* κλειδίωμα όπως έχει παρουσιαστεί στις διαλέξεις του μαθήματος.
- *ttas_lock*: Το *test-and-test-and-set* κλειδίωμα όπως έχει παρουσιαστεί στις διαλέξεις του μαθήματος.
Υπόδειξη: Για την υλοποίηση των κλειδωμάτων *tas* και *ttas* μπορείτε να χρησιμοποιήσετε τις υλοποιήσεις ατομικών εντολών που παρέχονται από τον gcc [1]. Πιο συγκεκριμένα η εντολή *test-and-set* παρέχεται ως `__sync_lock_test_and_set(...)`.
- *array_lock*: Το array-based κλειδίωμα όπως περιγράφεται και στις διαφάνειες του μαθήματος.
- *clh_lock*: Ένα είδος κλειδώματος που στηρίζεται στη χρήση μίας συνδεδεμένης λίστας. Αναλυτικές πληροφορίες μπορείτε να βρείτε στο Κεφάλαιο 7 του βιβλίου "The Art of Multiprocessor Programming".

3.1 Ερωτήσεις - Ζητούμενα

1. Υλοποιήστε τα ζητούμενα κλειδώματα συμπληρώνοντας τα αντίστοιχα αρχεία της μορφής `<lock_type>_lock.c`.
2. Εκτελέστε την εφαρμογή με όλα τα διαφορετικά κλειδώματα που σας παρέχονται και αυτά που υλοποιήσατε. Εκτελέστε για 1,2,4,8,16,32,64 νήματα και για λίστες μεγέθους 16, 1024, 8192. Παρουσιάστε τα αποτελέσματά σας σε διαγράμματα αντίστοιχα με το πρώτο μέρος της άσκησης και εξηγήστε την συμπεριφορά της εφαρμογής για κάθε κλειδίωμα.
Σημείωση: σε όλες τις εκτελέσεις θα θέσετε κατάλληλα την μεταβλητή περιβάλλοντος `MT_CONF` ώστε τα νήματα να καταλαμβάνουν διαδοχικούς πυρήνες, π.χ. τα 16 νήματα εκτελούνται στους πυρήνες 0-15.

4 Τακτικές συγχρονισμού για δομές δεδομένων

Στο τρίτο και τελευταίο μέρος της άσκησης στόχος είναι η υλοποίηση και η αξιολόγηση των διαφορετικών εναλλακτικών τακτικών για δομές δεδομένων. Η δομή με την οποία θα ασχοληθείτε είναι η ταξινομημένη συνδεδεμένη λίστα. Στο φάκελο `/home/parallel/rps/2017-2018/a5/z3/` σας δίνεται το αρχείο `main.c` το οποίο αναλαμβάνει να δημιουργήσει τα νήματα κάθε ένα εκ των οποίων εκτελεί τυχαίες λειτουργίες στην λίστα.

Εκτελώντας *make* δημιουργείται το εκτελέσιμο `linked_list` το οποίο παίρνει 4 ορίσματα, το εύρος των κλειδιών που θα χρησιμοποιηθούν από τα νήματα και το ποσοστό των λειτουργιών που θα είναι αναζητήσεις (`contains()`), εισαγωγές (`add()`) και διαγραφές (`remove()`). Σας δίνεται το αρχείο `ll.h` όπου ορίζουμε τις συναρτήσεις που πρέπει να υλοποιεί η λίστα καθώς επίσης και η σειριακή υλοποίηση στο αρχείο `ll_serial.h`. Εσείς καλείστε να υλοποιήσετε λίστες με τις παρακάτω τακτικές συγχρονισμού όπως αυτές έχουν παρουσιαστεί στις διαλέξεις του μαθήματος²:

- Fine-grain locking
- Optimistic synchronization
- Lazy synchronization
- Non-blocking synchronization

²Σε όλες υλοποιήσεις απαιτείται η χρήση κλειδωμάτων αυτά θα είναι τύπου `pthread_spinlock_t`.

4.1 Ερωτήσεις - Ζητούμενα

1. Υλοποιήστε τις ζητούμενες λίστες συμπληρώνοντας τα αντίστοιχα αρχεία της μορφής `ll_<sync_type>.c`.
2. Εκτελέστε την εφαρμογή για όλες τις διαφορετικές υλοποιήσεις λίστας. Εκτελέστε για 1,2,4,8,16,32,64 νήματα, για λίστες μεγέθους 1024 και 8192 και για συνδυασμούς λειτουργιών 80-10-10 και 20-40-40. Παρουσιάστε τα αποτελέσματά σας σε διαγράμματα και εξηγήστε την συμπεριφορά της εφαρμογής για κάθε κλείδωμα.

Σημείωση: σε όλες τις εκτελέσεις θα θέσετε κατάλληλα την μεταβλητή περιβάλλοντος `MT_CONF` ώστε τα νήματα να καταλαμβάνουν διαδοχικούς πυρήνες, π.χ. τα 16 νήματα εκτελούνται στους πυρήνες 0-15.

Χρήσιμοι σύνδεσμοι

[1] <https://gcc.gnu.org/onlinedocs/gcc-4.1.2/gcc/Atomic-Builtins.html>