

Εργασία 1 (υποχρεωτική) - Προγραμματισμός με Pthreads

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2022 - 2023

(ΕΚΦΩΝΗΣΗ) ΤΡΙΤΗ 8 ΝΟΕΜΒΡΙΟΥ 2022

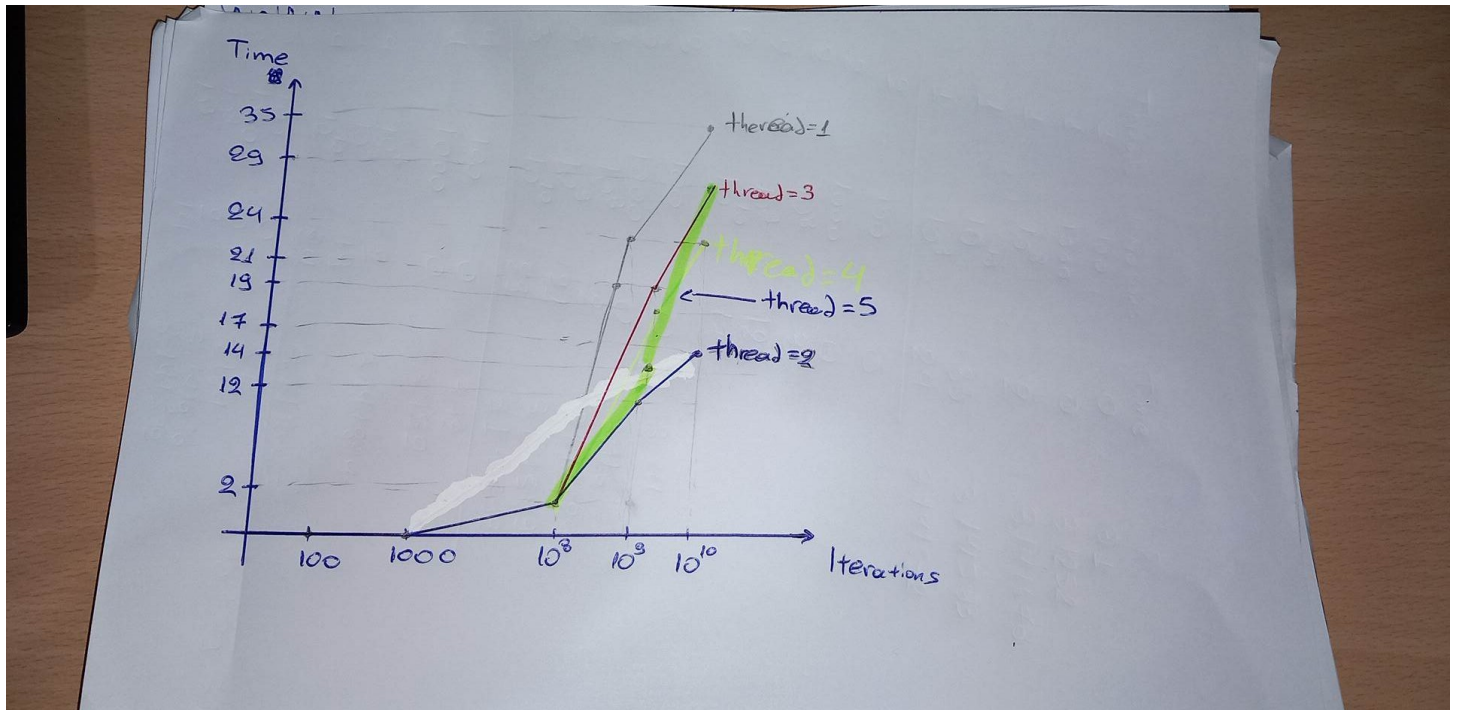
(ΠΑΡΑΔΟΣΗ ΣΤΟ ECLASS ΜΕΧΡΙ) **ΤΡΙΤΗ 6 ΔΕΚΕΜΒΡΙΟΥ 2022**

Επώνυμο	Όνομα	Αριθμός Μητρώου	Email
Χατζόπουλος	Άγγελος Δωρόθεος	1115201900217	sdi1900217@di.uoa.gr
Μουλκιώτης	Γρηγόριος	1115201900117	sdi1900117@di.uoa.gr

Άσκηση 1.1

Τα αρχεία αυτής της άσκησης είναι τα `main.c` και `pi_approc.c` στο `src` και στο `includes` το `pi_approx.h`. Στη αρχική μας προσέγγιση της άσκησης χρησιμοποιήσαμε `mutex` στο σημείο της `pi_approc.c` όπου προστείνονται όλα τα εύστοχα βέλη όμως για να αποφύγουμε την επικοινωνία μεταξύ των νημάτων αφαιρέσαμε τα `mutexes` και κάθε `thread` αλληλεπιδρά με δικό της χώρο που και μόνο στο τέλος όλες η κύρια διεργασία προσθέτει των αριθμό των βελών μέσω του πίνακα των `structs`. Σε κάθε περίπτωση χωρίζουμε κατάλληλα των φόρτο μεταξύ των `threads`. Παρατηρούμε ότι όσο αυξάνονται τα `threads` έχουμε και μια αύξηση των επιδόσεων.

THREADS	ITERATIONS	TIME	Pi
1	100	0	3.1
1	1000	0	3.1252
1	10 ⁸	2	3.141678
1	10 ⁹	24	3.141592
1	10 ¹⁰	35	3.141593
2	100	0	3.2
2	1000	0	3.084
2	10 ⁸	2	3.141595
2	10 ⁹	12	3.141612
2	10 ¹⁰	17	3.141600
3	100	0	3.24
3	1000	0	3.116
3	10 ⁸	2	3.141724
3	10 ⁹	21	3.141589
3	10 ¹⁰	29	3.141592
4	100	0	3.12
4	1000	0	3.152
4	10 ⁸	2	3.141844
4	10 ⁹	21	3.141596
4	10 ¹⁰	29	3.141607
5	100	0	3.12
5	1000	0	3.164
5	10 ⁸	2	3.141916
5	10 ⁹	14	3.141592
5	10 ¹⁰	29	3.1416001



Εδώ αξίζει να σημειωθεί πως επειδή τα προγράμματα τρέχουν στο cluster της σχολής έχουμε πτώση αποδόσεων για 3,4 και 5 threads και οι χρόνοι παραμένουν ουσιαστικά σταθεροί, παρόλα αυτά βλέπουμε πως από το 1 στα 2 threads ο χρόνος μειώθηκε στο μισό.

Άσκηση 1.2

Τα αρχεία αυτής της άσκησης είναι τα `matrix.c`, `matrix2.c` και `pth_mat_vect_rand_split.c`

στο `src` και στο `includes` το `pi_approx.h` όπου το `pth_mat_vect_rand_split.c` αποτελεί την αρχική υπολοποίηση και τα `matrix` αποτελούν υλοποιήσεις που αξιοποιούν τα `optimization` της εργασίας. Στο `matrix.c` δεσμεύουμε μεγαλύτερο πίνακα και εισάγουμε εικονικά στοιχεία ανάμεσα στις θέσεις που πειράζουν τα threads έτσι ώστε να αποφύγουμε το `write back` στην κύρια μνήμη από την `cache`.

Στο `matrix2.c` δίνουμε σε κάθε thread δικό του χώρο μεταβλητών και έτσι αντιμετωπίσουμε το πρόβλημα του `false sharing`, τα αποτελέσματα είναι καλύτερα σε κάθε thread με εξαίρεση ελάχιστες περιπτώσεις.

Για το default πρόγραμμα έχουμε τις εξής :

```
./pth_mat_vect_rand_split 8 8000 8000
Thread 4 > Elapsed time = 2.718711e-02 seconds
Thread 0 > Elapsed time = 3.865695e-02 seconds
Thread 2 > Elapsed time = 3.890109e-02 seconds
Thread 1 > Elapsed time = 4.329014e-02 seconds
Thread 6 > Elapsed time = 3.438497e-02 seconds
Thread 5 > Elapsed time = 3.851819e-02 seconds
Thread 3 > Elapsed time = 3.931713e-02 seconds
Thread 7 > Elapsed time = 3.456688e-02 seconds
Total Time: 0.059183
```

```
./pth_mat_vect_rand_split 8 8000000 8
Thread 0 > Elapsed time = 5.095601e-02 seconds
Thread 2 > Elapsed time = 5.307102e-02 seconds
Thread 1 > Elapsed time = 5.410910e-02 seconds
```

[Χατζόπουλος - Μουλκιώτης]

```

Thread 3 > Elapsed time = 5.489397e-02 seconds
Thread 4 > Elapsed time = 4.916286e-02 seconds
Thread 7 > Elapsed time = 4.958200e-02 seconds
Thread 6 > Elapsed time = 5.386806e-02 seconds
Thread 5 > Elapsed time = 5.071592e-02 seconds
Total Time: 0.063723

./pth_mat_vect_rand_split 8 8 8000000
Thread 0 > Elapsed time = 8.972287e-02 seconds
Thread 1 > Elapsed time = 9.574103e-02 seconds
Thread 7 > Elapsed time = 1.955969e-01 seconds
Thread 6 > Elapsed time = 2.258968e-01 seconds
Thread 3 > Elapsed time = 2.694101e-01 seconds
Thread 2 > Elapsed time = 2.704432e-01 seconds
Thread 4 > Elapsed time = 2.720921e-01 seconds
Thread 5 > Elapsed time = 2.743111e-01 seconds
Total Time: 0.285172

```

Για την πρώτη βελτιστοποίηση στο πρόγραμμα έχουμε τις εξής εκτελέσεις:

```

./matrix 8 8000 8000
Thread 2 > Elapsed time = 3.922486e-02 seconds
Thread 1 > Elapsed time = 4.398513e-02 seconds
Thread 0 > Elapsed time = 5.024314e-02 seconds
Thread 3 > Elapsed time = 5.122399e-02 seconds
Thread 4 > Elapsed time = 4.259300e-02 seconds
Thread 5 > Elapsed time = 4.269505e-02 seconds
Thread 7 > Elapsed time = 3.910804e-02 seconds
Thread 6 > Elapsed time = 4.312181e-02 seconds
Total Time: 0.055318

```

```

./matrix 8 8000000 8
Thread 3 > Elapsed time = 3.088498e-02 seconds
Thread 2 > Elapsed time = 4.323602e-02 seconds
Thread 0 > Elapsed time = 5.051303e-02 seconds
Thread 1 > Elapsed time = 5.162287e-02 seconds
Thread 5 > Elapsed time = 5.136585e-02 seconds
Thread 6 > Elapsed time = 5.093098e-02 seconds
Thread 4 > Elapsed time = 5.508614e-02 seconds
Thread 7 > Elapsed time = 4.917097e-02 seconds
Total Time: 0.069326

```

```

./matrix 8 8 8000000
Thread 3 > Elapsed time = 2.735281e-02 seconds
Thread 2 > Elapsed time = 3.980398e-02 seconds
Thread 5 > Elapsed time = 3.561807e-02 seconds
Thread 1 > Elapsed time = 5.079293e-02 seconds
Thread 0 > Elapsed time = 5.110407e-02 seconds
Thread 6 > Elapsed time = 3.964305e-02 seconds
Thread 4 > Elapsed time = 4.353118e-02 seconds
Thread 7 > Elapsed time = 4.322290e-02 seconds

```

Total Time: 0.063408

Για την δεύτερη βελτιστοποίηση στο πρόγραμμα έχουμε τις εξής εκτελέσεις:

```
./matrix2 8 8000 8000
```

```
Thread 2 > Elapsed time = 2.721810e-02 seconds
Thread 7 > Elapsed time = 3.920197e-02 seconds
Thread 4 > Elapsed time = 3.109193e-02 seconds
Thread 1 > Elapsed time = 4.921293e-02 seconds
Thread 0 > Elapsed time = 5.166698e-02 seconds
Thread 6 > Elapsed time = 4.364109e-02 seconds
Thread 5 > Elapsed time = 3.813887e-02 seconds
Thread 3 > Elapsed time = 4.735780e-02 seconds
Total Time: 0.059030
```

```
./matrix2 8 8000000 8
```

```
Thread 7 > Elapsed time = 3.107715e-02 seconds
Thread 0 > Elapsed time = 4.756594e-02 seconds
Thread 2 > Elapsed time = 5.069900e-02 seconds
Thread 1 > Elapsed time = 5.770493e-02 seconds
Thread 5 > Elapsed time = 4.918790e-02 seconds
Thread 4 > Elapsed time = 5.097890e-02 seconds
Thread 6 > Elapsed time = 4.385114e-02 seconds
Thread 3 > Elapsed time = 5.799294e-02 seconds
Total Time: 0.070928
```

```
./matrix2 8 8 8000000
```

```
Thread 2 > Elapsed time = 4.383492e-02 seconds
Thread 1 > Elapsed time = 5.031610e-02 seconds
Thread 0 > Elapsed time = 5.123091e-02 seconds
Thread 3 > Elapsed time = 5.129004e-02 seconds
Thread 5 > Elapsed time = 4.525685e-02 seconds
Thread 4 > Elapsed time = 4.526281e-02 seconds
Thread 7 > Elapsed time = 4.605389e-02 seconds
Thread 6 > Elapsed time = 4.657388e-02 seconds
Total Time: 0.056263
```

Παρατηρούμε ότι οι δύο βελτιστοποιήσεις βελτιώνουν τους χρόνους ελάχιστα αλλά το πιο σημαντικό είναι ότι έχουμε σταθερούς χρόνους ανάμεσα στα νήματα το οποίο συμβαίνει επειδή αποφεύγετε το false sharing.

Πληροφορίες σχετικά με το λογισμικό και το υλικό:

Υπολογιστής:linux30.di.uoa.gr

Πυρήνες:4

Μεταγλωττιστής:gcc

Οδηγίες μεταγλωττιστής:

Για την πρώτη άσκηση:

[Χατζόπουλος - Μουλκιώτης]

Εκτελούμε την εντολή

```
gcc -o main main.c pi_approx.c -lpthread
```

Και για εκτέλεση

```
./main <threads> <Iterations>
```

Για την δεύτερη άσκηση:

Για το default:

```
gcc -o pth_mat_vect_rand_split pth_mat_vect_rand_split.c -lpthread
```

Για την πρώτη βελτιστοποίηση:

```
gcc -o matrix matrix.c -lpthread
```

Για την δεύτερη:

```
gcc -o matrix2 matrix2.c -lpthread
```