



Παράλληλος Προγραμματισμός

Εργασία 1

Καραπιέρης Αχιλλεύς(Π2013023)

p13kara@ionio.gr

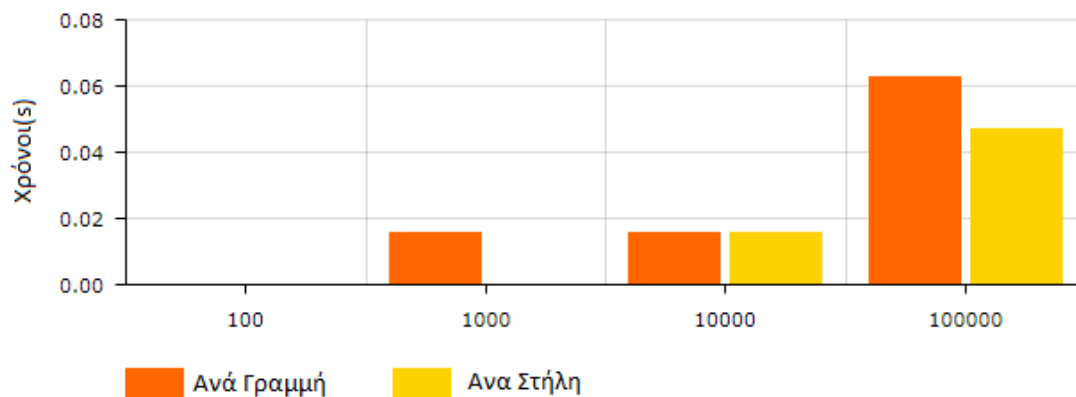
13/3/2018

Για την ολοκλήρωση της εργασίας έγινε καταχώριση μιας τυχαίας τιμής(πάντα η ίδια) σε κάθε κελί του πίνακα. Σε κάθε πείραμα, ο αριθμός των στηλών παρέμεινε σταθερός και άλλαξε ο αριθμός των γραμμών(100, 1000, 10000, 100000).

Παρακάτω παρουσιάζεται ο χρόνος που χρειάστηκε για να ολοκληρωθεί το κάθε πείραμα.

	Ανα Γραμμή	Ανα Στήλη
100	0.000000	0.000000
1000	0.015615	0.000000
10000	0.015627	0.015623
100000	0.062503	0.046880

Για την καλύτερη σύγκριση δημιουργήθηκε το παρακάτω διάγραμμα



Η διαφορά που παρατηρείται ανάμεσα στα δύο πειράματα οφείλονται στο τρόπο που έχουν τοποθετηθεί τα στοιχεία του πίνακα στη μνήμη.

Παράρτημα 1

Προσπέλαση ανα γραμμή (matrix1.c)

```
#include <stdio.h>

#include <stdlib.h>

#include <sys/time.h>

#include <time.h>

// compile like: gcc -Wall -O2 -DNROWS=10000 matrix1.c -o matrix1

#define NCOLS 100

#define NROWS 100

void get_walltime(double *wct) {

    struct timeval tp; //domh tou system

    gettimeofday(&tp, NULL);

    *wct = (double) (tp.tv_sec + (tp.tv_usec / 1000000.0));

}

int main() {

    double i, j, M, N;

    double ts, te, *table;

    table = (double *)malloc(NROWS*NCOLS*sizeof(double));

    if (table==NULL) {

        printf("alloc error!\n");

        exit(1);
```

```
}  
  
M=100;  
  
N=100;  
  
get_walltime(&ts);  
  
for (i = 1; i < M - 1; i++) { //einai oi grammes ?  
    for (j = 1; j < N - 1; j++) { //einai oi sthles ?  
        *table=5;  
    }  
}  
  
get_walltime(&te);  
  
printf("Done on %lf seconds\n", (te - ts)); //emfanish tou xronou pou xriasthke gia  
na treksei to programma  
  
free(table);  
  
return 0;  
  
}
```

Παράρτημα 2

Προσπέλαση ανα στήλη(matrix2.c)

```
#include <stdio.h>

#include <stdlib.h>

#include <sys/time.h>

#include <time.h>

// compile like: gcc -Wall -O2 -DNROWS=10000 matrix1.c -o matrix1

#define NCOLS 100

#define NROWS 100

void get_walltime(double *wct) {

    struct timeval tp; //domh tou system

    gettimeofday(&tp, NULL);

    *wct = (double) (tp.tv_sec + (tp.tv_usec / 1000000.0));

}

int main() {

    double i, j, M, N;

    double ts, te, *table;

    table = (double *)malloc(NROWS*NCOLS*sizeof(double));

    if (table==NULL) {

        printf("alloc error!\n");

        exit(1);
```

```
}

M=100;

N=100;

get_walltime(&ts);

    for (j = 1; j < N - 1; j++) { //einai oi sthles ?

        for (i = 1; i < M - 1; i++) { //einai oi grammes ?

            *table=5;

        }

    }

}

get_walltime(&te);

    printf("Done on %lf seconds\n", (te - ts)); //emfanish tou xronou pou xriasthke gia
na treksei to programma

free(table);

return 0;

}
```