Ευφραίμ Αναστασιάδης

Π2016157

Αναφορά

Παράλληλος Προγραμματισμός 2020

Προγραμματιστική Εργασία #1

Αναφορα:

Χωρίς ΣΣΕ

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#define N 1000
#define M 1000
```

Δηλώνουμε τις απαραίτητες βιβλιοθήκες μαζί με το M και το N τα οποία θα αλλάζουμε για να δούμε την αποδοτικότητα του προγράμματος.

```
//initialization

float *pic, *newpic;
float k0 = 0.5, k1 = 0.5, k2 = 0.5, k3 = 0.5, k5 = 0.5, k6 = 0.5, k7 = 0.5, k8 = 0.5, k4 = 5.0;

int i = 0;
int j = 0;

double ts,te;
```

Εδώ βλέπουμε την δήλωση τον απαραίτητων μεταβλητών και κατά επέκταση την αρχικοποίηση τους αν χρειάζεται.

Με τις pic και newpic να είναι οι πίνακες που θα χρησιμοποιήσουμε για να βάλουμε τα pixels, τα K από 1 μεχρι 3 και 5 μεχρι 8 τα αρχικοποιούμε με τιμή 0.5 και το k 4 με τιμή 5 σύμφωνα με την εκφώνηση και οι μεταβλητές ts,te θα αντιπρωσοπεύουν την αρχή και το τέλος του χρόνου.

```
//allocate the memory
pic = (float*)malloc(N* M* sizeof(float));
newpic = (float*) malloc(N* M* sizeof(float));
```

Δυναμική δέσμευση των πινάκων

```
//init of the arrays
for(i = 0; i < N * M; i++){
        pic[i] = i * (i + 3) ;
        newpic[i] = 1.0;
}
```

Και αρχικοποίηση τους με τυχαίους αριθμούς(του πρώτου πίνακα γιατί ο δεύτερος θα αλλάξει με την δική μας επεξεργασία)

Με την

```
get_walltime(&ts);
```

Ξεκινάμε το ρολόι..

```
for(i = 1; i < N - 1; i++){
        for (j = 1; j < M - 1; j++){
                newpic[i] = (pic[(i - 1) + (j - 1)] * k0) + (pic[(i - 1) + j] * k1) + (pic[(i - 1) + (j + 1)] * k2) +
        }
}
```

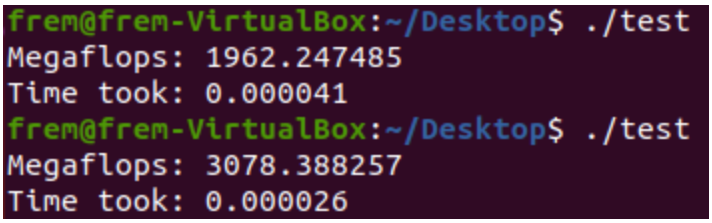
Στην συνέχεια υπολογίζουμε τα pixels της καινούργιας εικόνας με την βοήθεια των K.

Και με την

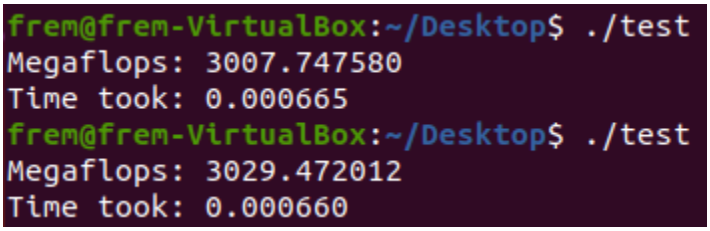
```
get_walltime(&te);
```

Σταματάμε το ρολόι

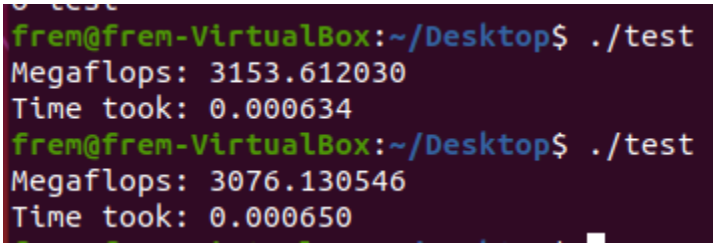
Τέλος εκτυπώνουμε τα αποτελέσματα και ελευθερόνουμε την μνήμη.



Αποτελέσματα για M = 300 και N = 300



Αποτελέσματα για M = 1000 και N = 1000



Αποτελέσματα για διαφορετικό αριθμό στην δήλωση του πίνακα πάλι με N και M με τιμές 1000.(που προφανώς δεν επηρεάζει το πρόγραμμα)

Με ΣΣΕ:

Δυστηχώς δεν κατάφερα να φτάσω σε ένα πρόγραμμα το οποίο λειτουργεί αλλά θα περιγράψω πως θεωρητικά θα δούλευε το πρόγραμμα. Το SSE το οποίο σημαίνει Streaming SIMD Extensions μας επιτρέπει να εκτελέσουμε μια εντολή σε πολλά δεδομένα με την χρήση δεδομένων __m128 .

Αρχικά στην δήλωση των μεταβλητών θα χρειαστούμε έναν sse pointer στον πίνακα με τις σταθερές τιμές.

Για την δέσμευση της μνήμης θα χρησιμοποιήσουμε την posim_memalign αντι για την malloc για να τα κάνουμε 16bit.

Το ρολοίο μας παραμένει ίδιο φυσικά και ξεκινάει μετά την δέσμευση.

Στη συνέχεια όταν γεμίσουμε το φορτίο μέσα στην επανάληψη του προηγούμενου χρησιμοποιούμε την εντολή _mm_mul_ps για να πολλαπλασιάσουμε τον pointer που δείχνει στον πίνακα με τις σταθερές τιμές και τις τιμές απο το array, με τις πράξεις που έχουμε απο την εκφώνηση ,με την βοήθεια του _mm_set_ps που θα τα μετατρέψει σε vectors που είναι συμβατά με __m128 και έπειτα θα τα βάλουμε όλα σε μια μεταβλητή sum.

Τέλος γεμίζουμε τπ array της τελικής εικόνας και μετά την επανάληψη κλείνουμε το ρολόι.