ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΑΡΑΛΛΗΛΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ 2020

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ #1

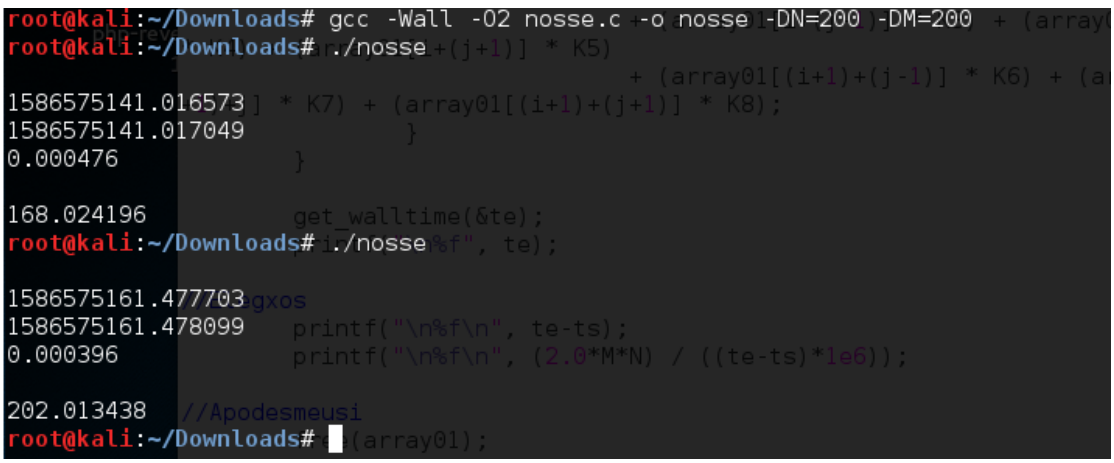
Ονοματεπώνυμο: Τράντη Ελένη

Α.Μ: Π2016106

Περιγραφή κώδικα για το πρόγραμμα no-sse:

Στην αρχή του κώδικα υπάρχουν οι βιβλιοθήκες που θα χρησιμοποιήσουμε και ακριβώς επόμενη είναι μια συνάρτηση που ορίζει τον χρόνο με όρισμα μια τιμή double. Επόμενη είναι η δήλωση των μεταβλητών και συγκεκριμένα: δυο πινάκων τύπου float, 9 σταθερών τιμών τύπου float, 2 τιμών double που αναπαριστούν την αρχική και την τελική τιμή του χρόνου που μετρήθηκε και τέλος οι i, j μεταβλητές που χρησιμοποιούνται στις επαναλήψεις. Έπειτα, είναι η δυναμική δέσμευση των πινάκων με την συνάρτηση malloc και μέγεθος M * N και η αρχικοποίηση τους μέσα στην for με τον πρώτο πίνακα να παίρνει τυχαίες τιμές και τον δεύτερο του οποίου οι τιμές θα αλλάξουν στην συνέχεια με την τιμή 1. Στην συνέχεια είναι η συνάρτηση getwalltime και τοποθετήθηκε πριν τον μετασχηματισμό των pixel ώστε να μας επιστρέψει τον χρόνο που την καλέσαμε και να μπορέσουμε να συγκρίνουμε τις τιμές με την δεύτερο πρόγραμμα. Αμέσως μετά είναι οι πράξεις για τον μετασχηματισμό των pixel. Οι επαναλήψεις ξεκινούν από το 1 μέχρι το N-1 καθώς παραλείπουμε τα γειτονικά pixel και στην επανάληψη for είναι οι πράξεις που χρειάζονται για κάθε pixel και εκχωρούνται στον δεύτερο πίνακα. Τέλος είναι η κλήση της getwalltime οπού θα αποθηκεύσει τον χρόνο που τέλειωσαν οι πράξεις έτσι ώστε στην συνέχεια να βγει ο συνολικός χρόνος που χρειάστηκε για να γίνουν όλες οι πράξεις και το πρόγραμμα να κλείσει με την αποδέσμευση των πινάκων.

Πρώτο παράδειγμα με τιμές 200*200:



Δεύτερο παράδειγμα με τιμές 1000*1000:

```
root@kali:~/Downloads# gcc -Wall -O2 nosse.c -o nosse -DN=1000 -DM=1000
root@kali:~/Downloads# ./nosse
            //Elegxos
1586575268.872016    printf("\n%f\n", te-ts);
1586575268.882330    printf("\n%f\n", (2.0*M*N) / ((te-ts)*1e6));
0.010314
            //Apodesmeusi
193.911419           free(array01);
root@kali:~/Downloads# ./nosse02);
                     return 0;
1586575272.135705
1586575272.151959
0.016254

123.047027   //gia na kano compile grafo :   gcc -Wall -O2 newfile.c -o newfile
```

Παρατηρούμε ότι οι τιμές δεν έχουν μεγάλη απόκλιση μεταξύ τους για διαφορετικές τιμές.

Περιγραφή κώδικα για το πρόγραμμα sse:

Η διαφορά με αυτό το πρόγραμμα είναι η χρήση της μεθόδου mm_set_ps και η δημιουργία του διανύσματος __m128 που χρησιμοποιείται για την διαφορετική προσπέλαση των pixel. Η πρώτη χρήση του γίνεται στην δήλωση των μεταβλητών οπού δηλώνουμε τις μεταβλητές ώστε την συνέχεια να με τους συνδυασμούς μεταξύ τους να έχουμε το επιθυμητό αποτέλεσμα. Οι συνδυασμοί αυτοί γίνονται στην επαναληπτική for στην οποία αφού κάνουμε τις πράξεις για τα pixel στην συνέχεια αποθηκεύουμε την τιμή στον πίνακα και μέσω της _mm_load σε μια μεταβλητή calc. Την μεταβλητή αυτή θα την προσθέσουμε με την _mm_add με στις υπόλοιπες και θα βγάλουμε ένα τελικό sum οπού και θα εισχωρήσουμε στον πίνακα array02.

Το πρόγραμμα δεν καταφέρα να το υλοποιήσω ώστε να τρέχει αλλά υπάρχει στα αρχεία του φακέλου του μαθήματος.

Συμπεράσματα:

Το αποτέλεσμα που περιμένουμε από τους χρόνους εκτέλεσης και των δύο προγραμμάτων είναι το πρόγραμμα με τις εντολές sse να είναι πολύ πιο γρήγορο από το άλλο καθώς οι πράξεις SIMD είναι γρηγορότερες από τις κανονικές μια και γίνονται ταυτόχρονα 4.