Παράλληλος Προγραμματισμός 2017
Προγραμματιστική Εργασία #1

(Προσοχή: η παράδοση της άσκησης θα γίνει μέσω github. Διαβάστε τις οδηγίες στο τέλος της εκφώνησης)

Θέμα

Στην επεξεργασία εικόνας συχνά εμφανίζεται ο μετασχηματισμός, κατά τον οποίο κάθε **pixel** αντικαθίσταται από μια νέα τιμή υπολογισμένη από **τα 8 γειτονικά pixels** (και το ίδιο). Κατά τον μετασχηματισμό, κάθε αρχικό pixel πολλαπλασιάζεται με μια σταθερά **float**.

Έτσι, αν έχουμε τα γειτονικά pixels:
P0 P1 P2
P3 P4 P5
P6 P7 P8

και τις σταθερές μετασχηματισμού:
K0 K1 K2
K3 K4 K5
K6 K7 K8

τότε η νέα τιμή για το pixel P4 ισούται με:
newP4 = P0*K0+P1*K1+P2*K2+P3*K3+P4*K4+
+P5*K5+P6*K6+P7*K7+P8*K8

Η διαδικασία επαναλαμβάνεται για όλα τα pixel, για όλες τις γραμμές της εικόνας.

Για τις ανάγκες της άσκησης θεωρήστε ότι:

- Η εικόνα είναι **μονόχρωμη** (1 float αριθμός ανά pixel).
- Η εικόνα έχει ήδη μετατραπεί και φορτωθεί σε έναν πίνακα float NxM (αρχικοποιήστε με **τυχαίες τιμές** τον πίνακα πριν αρχίσετε τη μέτρηση του χρόνου).
- Οι νέες τιμές που υπολογίζετε θα αποθηκεύονται **σε δεύτερο πίνακα** με τις ίδιες διαστάσεις (αρχικοποιήστε και αυτόν τον πίνακα για να μεταφερθεί στην κρυφή μνήμη πριν αρχίσετε να μετράτε χρόνο).
- Τα M και N είναι τέτοια ώστε να βολεύουν στις πράξεις σας, π.χ. μπορείτε να θεωρήσετε ότι το N είναι πολλαπλάσιο του 4 ή οτιδήποτε άλλο σας βολεύει.
- Μπορείτε, αν θέλετε, να αγνοήσετε τον υπολογισμό των pixels στο περίγραμμα της εικόνας.
- Για τις ανάγκες της άσκησης, τα K0, K1, K2, K3, K5, K6, K7, K8 ισούνται με 0.5, ενώ το K4 είναι ίσο με 5.0 (μπορείτε δηλαδή να χρησιμοποιήσετε **2 σταθερές μόνο**!).

Ζητείται η επιλογή όσο το δυνατόν αποδοτικότερων SIMD πράξεων (από το σετ SSE/SSE2) για την υλοποίηση του μετασχηματισμού. Σε αντίθεση με παραδείγματα που θα βρείτε on-line, τα οποία χρησιμοποιούν 3 ανεξάρτητες συνιστώσες χρώματος (RGB) και εξάγουν την παραλληλία από αυτό ακριβώς το γεγονός, εδώ η εικόνα είναι μονόχρωμη και ο παραλληλισμός δυσκολότερος! **Βεβαιωθείτε ότι το τροποποιημένο πρόγραμμά σας δεν είναι αργότερο από το αρχικό!**

Ζητούμενο

Ο στόχος της άσκησης είναι:

α) Να κατασκευάσετε δοκιμαστικό πρόγραμμα σε C χωρίς εντολές SSE, έτσι ώστε να μελετήσετε τον χρόνο εκτέλεσης χωρίς πρόσθετες βελτιώσεις. Χρησιμοποιήστε πίνακες που χωράνε στην κρυφή μνήμη (L2) του συστήματός σας, έως π.χ. 1000x1000.

β) Κατασκευάστε το αντίστοιχο πρόγραμμα με εντολές SSE. Χρησιμοποιήστε ως βάση τον κώδικα που δόθηκε στο εργαστήριο. Μετρήστε τους αντίστοιχους χρόνους εκτέλεσης και εισάγετε περαιτέρω βελτιώσεις, αν χρειάζεται. **Βεβαιωθείτε ότι οι αναγνώσεις και εγγραφές είναι ευθυγραμμισμένες στα 16 bytes**!

Μπορείτε να αναζητήσετε on-line πληροφορία για τα intrinsics που θα χρησιμοποιήσετε, π.χ. στο "Intel® 64 and IA-32 Architectures Software Developer's Manual" της Intel, Vol.2 (Instruction Set Reference, A-Z):
http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html

Παραδοτέο
Η παράδοση θα γίνει μέσω github. Οδηγίες:

1. Αντιγράψτε (fork) το repository https://github.com/mixstef/parprog1617a1 στο δικό σας repository.
2. Τροποποιήστε κατάλληλα τα αρχεία που περιέχονται στο repository σας με το δικό σας περιεχόμενο:
   - Συμπληρώστε τα στοιχεία σας στο αρχείο README.md .
   - Βάλτε τον κώδικά σας στα αρχεία no-sse.c και sse.c .
   - Προσθέστε την αναφορά σας ως report.pdf .
   - **Προσοχή: πρέπει να διατηρήσετε τα ονόματα των παραπάνω αρχείων!**
3. Ενημερώστε το repository σας στο github.
4. Στείλτε με e-mail τη διεύθυνση του repository σας εντός προθεσμίας.

Η εργασία είναι αυστηρά ατομική. Για την εγκυρότητα της υποβολής σας θα χρησιμοποιηθεί η χρονοσήμανση των αλλαγών (commits) των αρχείων σας.

Προθεσμία παράδοσης: Τετάρτη 29/3/2017 13:00.