

Εργασία #1 (OpenMP)

Α. Παραλληλοποίηση εύρεσης πρώτων αριθμών (50%)

Σας δίνεται ένα σειριακό πρόγραμμα (**primes.c**) στο οποίο, δεδομένου του N , η συνάρτηση *serial_primes()* υπολογίζει το πλήθος των πρώτων αριθμών καθώς και τον μεγαλύτερο πρώτο αριθμό μέχρι και το N . Σας ζητείται να συμπληρώσετε τη συνάρτηση *openmp_primes()* ώστε να κάνει τους ίδιους υπολογισμούς παράλληλα, χρησιμοποιώντας το OpenMP. Δεν επιτρέπεται να αλλάξετε τον αλγόριθμο, απλά να μοιράσετε σωστά τη δουλειά.

- Πρέπει να χρονομετρήσετε και να συγκρίνετε με τη σειριακή έκδοση.
- Πρέπει να δοκιμάσετε εναλλακτικούς τρόπους διαμοιρασμού της δουλειάς μεταξύ των νημάτων και να καταλήξετε (αιτιολογώντας) στον καλύτερο δυνατό.

Β. Θόλωση εικόνων (50%)

Σας δίνεται ένα σειριακό πρόγραμμα (**gaussian-blur.c**) το οποίο εφαρμόζει Gaussian blur προκειμένου να θολώσει (ή να ομαλοποιήσει) μία εικόνα. Η συνάρτηση που κάνει τη θόλωση είναι η *gaussian_blur_serial()*, η οποία παίρνει μία εικόνα **imgin** και παράγει τη θολωμένη της έκδοχή **imgout**, βάσει μίας ακτίνας θόλωσης **radius** (όσο μεγαλύτερη η ακτίνα, τόσο πιο έντονο το θόλωμα). Σας ζητείται να γίνεται η θόλωση παράλληλα χρησιμοποιώντας το OpenMP ως εξής (χωρίς να αλλάξετε τον αλγόριθμο, απλά να μοιράσετε σωστά τη δουλειά):

- να συμπληρώσετε τη συνάρτηση *gaussian_blur_omp_loops()*, παραλληλοποιώντας κατάλληλα loop(s).
- να συμπληρώσετε τη συνάρτηση *gaussian_blur_omp_tasks()* όπου για την παραλληλοποίηση γίνεται χρήση tasks. Θεωρείστε ότι μία γραμμή της εικόνας αποτελεί ένα task.

Για τη θόλωση εικόνων δίνεται εικόνα με ανάλυση 500x500, 1000x1000 και 1500x1500. Πειραματιστείτε με όλες, αλλά δώστε αποτελέσματα μόνο για την μεγαλύτερη εικόνα και για ακτίνα θόλωσης ίση με 8.

Απαιτούμενα

- Ο **πηγαίος κώδικας** που δίνετε για τις υλοποιήσεις σας θα πρέπει να είναι σωστά δομημένος, στοιχισμένος και σχολιασμένος (προτεινόμενη γλώσσα τα Αγγλικά).
- Θα πρέπει να παραδώσετε **πλήρη αναφορά**, περιλαμβάνοντας και γραφικές παραστάσεις χρονομετρήσεων καθώς και συζήτηση γύρω από τα αποτελέσματα.
- Τα προγράμματά σας (πηγαίοι κώδικες + αναφορά) θα πρέπει να τα παραδώσετε στο eclass του μαθήματος σε μορφή **zip αρχείου**. Στο όνομα του αρχείου θα πρέπει να περιλαμβάνεται ο **αριθμός μητρώου** του φοιτητή.
- Οι ασκήσεις ελέγχονται για κοινό κώδικα και **αντιγραφή**. Τέτοιες περιπτώσεις φυσικά θα **μηδενίζονται** και δεν θα υπάρχει δικαίωμα εξέτασης στις επόμενες εξεταστικές περιόδους.
- Για τη **χρονομέτρηση** μπορείτε να χρησιμοποιήσετε κλήσεις χρονομέτρησης, *omp_get_wtime()*, που παρέχει το ίδιο το OpenMP () ή την *gettimeofday()*.
- Τα προγράμματά σας να τα δοκιμάσετε με **1, 2, 3 και 4 νήματα** (και να τα συγκρίνετε με τον καθαρό σειριακό κώδικα).
- Για κάθε περίπτωση, ένα πρόγραμμα θα εκτελείται τουλάχιστον **4 φορές** και ο τελικός χρόνος θα είναι ο μέσος όρος των τεσσάρων χρόνων.

Παρατηρήσεις

- Η ανάπτυξη των προγραμμάτων σας μπορεί να γίνει οπουδήποτε αλλά η εκτέλεση και χρονομέτρηση των πειραμάτων σας θα **πρέπει να γίνει σε υπολογιστές του τμήματος** οι οποίοι διαθέτουν 4-πύρηνους επεξεργαστές.
- Η τελική βαθμολογία θα παρθεί μετά από **προφορική εξέταση**. Σχετικό πρόγραμμα εξέτασης θα βγει εγκαίρως μετά την παράδοση της εργασίας στην ιστοσελίδα του μαθήματος.
- Η παράδοση της εργασίας είναι **υποχρεωτική** και χρειάζεται βαθμολογία $> 50\%$ για να μπορείτε να δώσετε στις τελικές γραπτές εξετάσεις. Προσοχή, η επίλυση μόνο ενός ερωτήματος δεν αρκεί!

Προθεσμία παράδοσης: Κυριακή, 7 Μαΐου 2022

Καλή Επιτυχία!

Ανδρέας-Αλέξανδρος Βασιλάκης

24/04/2023