Εικόνα που περιέχει κείμενο, δωμάτιο, καζίνο

Περιγραφή που δημιουργήθηκε αυτόματα

**Πανεπιστήμιο Ιωαννίνων**

**Πολυτεχνική Σχολή**

**Τμήμα Μηχανικών Η/Υ και Πληροφορικής**

**Προπτυχιακό Μάθημα: «Παράλληλα Συστήματα κ’ Προγραμματισμός»**

**Πρώτο Σετ Προγραμματιστικών Ασκήσεων**

**Όνομα Φοιτητή – Α.Μ.:**

**Γεώργιος Κρομμύδας – 3260**

**E-mail Φοιτητή:**

[**cs03260@uoi.gr**](mailto:cs03260@uoi.gr)**,** [**cse63260@cs.uoi.gr**](mailto:cse63260@cs.uoi.gr)

**Εικόνα που περιέχει κείμενο, υπογραφή, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα**

*ΙΩΑΝΝΙΝΑ,*

*2022*

Πίνακας περιεχομένων

[**1. Εισαγωγή:** 3](#_Toc100933250)

[**2. Άσκηση-1:** 3](#_Toc100933251)

[**2.1. Το πρόβλημα:** 3](#_Toc100933252)

[**2.2. Μέθοδοι Παραλληλοποίησης:** 3](#_Toc100933253)

[**2.3. Πειραματικά Αποτελέσματα – Μετρήσεις:** 4](#_Toc100933254)

[**2.4. Σχόλια:** 8](#_Toc100933255)

[**3. Άσκηση-2:** 9](#_Toc100933256)

[**3.1. Το πρόβλημα:** 9](#_Toc100933257)

[**3.2. Μέθοδοι Παραλληλοποίησης:** 9](#_Toc100933258)

[**3.3. Πειραματικά Αποτελέσματα – Μετρήσεις:** 9](#_Toc100933259)

[**3.4. Σχόλια:** 9](#_Toc100933260)

[**4. Άσκηση-3:** 9](#_Toc100933261)

[**4.1. Το πρόβλημα:** 9](#_Toc100933262)

[**4.2. Μέθοδοι Παραλληλοποίησης:** 9](#_Toc100933263)

[**4.3. Πειραματικά Αποτελέσματα – Μετρήσεις:** 9](#_Toc100933264)

[**4.4. Σχόλια:** 9](#_Toc100933265)

[**Βιβλιογραφία** 10](#_Toc100933266)

# **1. Εισαγωγή:**

Το πρώτο σετ προγραμματιστικών ασκήσεων αφορά στον παράλληλο προγραμματισμό με το μοντέλο κοινόχρηστου χώρου διευθύνσεων μέσω του προτύπου ***OpenMP***. Ζητείται η παραλληλοποίηση δύο εφαρμογών οι οποίες αφοράν τον υπολογισμό των πρώτων αριθμών που εμφανίζονται στο σύνολο αριθμών , όπου ο είναι ο ένας ακέραιος αριθμός. Η δεύτερη εφαρμογή αφορά το φιλτράρισμα εικόνων με χρήση του φίλτρου ***Gaussian Blur*** και δεδομένης ακτίνας . Επιπλέον, από την έκδοση 4.5 του ***OpenMP*** υποστηρίζεται η εντολή ***taskloop*** η οποία επιτρέπει επαναλήψεις ενός βρόχου ***for*** να εκτελεστούν μέσω ***tasks*** και θα μελετηθεί με ένα απλό πρόγραμμα πολλαπλασιασμού πινάκων.

Όλες οι μετρήσεις έγιναν στο παρακάτω σύστημα:

|  |  |
| --- | --- |
| **Όνομα Υπολογιστή** | opti3060ws10 |
| **Επεξεργαστής** | Intel i3-8300 3.7Ghz |
| **Πλήθος Πυρήνων** | 4 |
| **Μεταγλωττιστής** | gcc v.7.5.0 |

Πίνακας 1: Λεπτομέρειες Συστήματος

# **2. Άσκηση-1:**

## **2.1. Το πρόβλημα:**

Σε αυτή την άσκηση ζητείται να παραλληλοποιηθεί ο αλγόριθμος υπολογισμού πρώτων αριθμών που είναι μικρότεροι από το και να συγκριθούν οι χρόνοι εκτέλεσης του σειριακού προγράμματος με το παράλληλο με διαφορετικό πλήθος νημάτων. Αυτό θα γίνει παραλληλοποιώντας τον βρόχο ***for*** του αλγορίθμου. Τέλος, ζητείται να δοκιμαστούν διαφορετικές μέθοδοι διαμοίρασης των επαναλήψεων με χρήση των ***schedules***.

## **2.2. Μέθοδοι Παραλληλοποίησης:**

Για την παραλληλοποίηση, χρησιμοποιήθηκε το σειριακό πρόγραμμα που υπήρχε στην ιστοσελίδα του μαθήματος ([primes.c](https://www.cse.uoi.gr/~dimako/media/coursestuff/code/primes.c)). Το πρώτο βήμα για την παραλληλοποίηση του προγράμματος είναι να προστεθεί η παρακάτω οδηγία

και στη συνέχεια προστέθηκε και η οδηγία

πριν από το βρόχο του ***for*** του ***i***. Αρχικά, οι μεταβλητές *num, divisor, quotient* και *remainder* πρέπει να είναι ιδιωτικές, έτσι ώστε κάθε νήμα να κάνει τους υπολογισμούς στο δικό του χώρο χωρίς να επηρεάζει τα υπόλοιπα νήματα, ενώ οι μεταβλητές *count* και *lastprime* πρέπει να είναι κοινόχρηστες. Δεν απαιτείται αμοιβαίος αποκλεισμός, καθώς το κάθε νήμα επηρεάζει διαφορετικά το *count* και το *lastprime* πρέπει να έχει μόνο την τελευταία μεγαλύτερη τιμή.

## **2.3. Πειραματικά Αποτελέσματα – Μετρήσεις:**

Το πρόγραμμα εκτελέστηκε στο σύστημα που αναφέρθηκε προηγουμένως στην εισαγωγή και η χρονομέτρηση έγινε με την συνάρτηση ***gettimeofday(struct timeval \*, struct tzp \*)***. Αρχικοποιήθηκαν δύο μεταβλητές τύπου ***struct timeval start, end*** για την αρχή της χρονομέτρησης και μετά στο τέλος. Αρχικά καλούνται οι εντολές ***gettimeofday(&start, NULL),*** ***serial\_primes(UPTO)*** και ***gettimeofday(&end, NULL)***. Αντίστοιχα, υπολογίζονται και για την συνάρτηση ***openmp\_primes(UPTO)***. Στη συνέχεια υπολογίζονται οι χρόνοι ***exectime exectimepar*** για να μετατραπούν οι χρόνοι από ***struct*** σε ***double*** με τον τύπο

Χρησιμοποιήσαμε από 1 μέχρι 4 νήματα μέσω της shell εντολής

για όλα τα πειράματα τα οποία εκτελέστηκαν.

Κάθε πείραμα εκτελέστηκε τέσσερις φορές και υπολογίστηκαν οι μέσοι χρόνοι. Οι χρόνοι αυτοί υπολογίζουν μονό την εκτέλεση του εκάστοτε αλγορίθμου (σειριακός ή παράλληλος). Τα αποτελέσματα δίνονται στους παρακάτω πίνακες (οι χρόνοι είναι σε sec) και διαχωρίζονται ανάλογα με το ***schedule***. Συνολικά θα έχουμε 5 πίνακες και 5 γραφικές παραστάσεις.

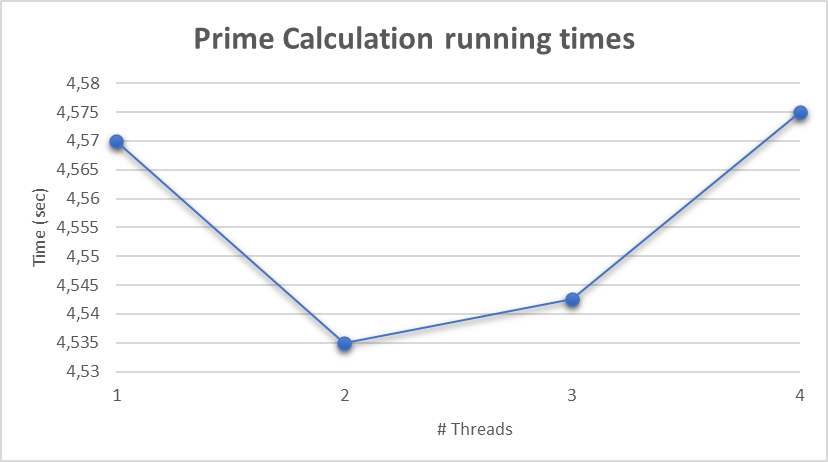
Το πρώτο πείραμα έχει ***schedule static***. Οι μετρήσεις φαίνονται παρακάτω:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Νήματα** | **1η Εκτέλεση** | **2η Εκτέλεση** | **3η Εκτέλεση** | **4η Εκτέλεση** | **Μέσος Χρόνος** |
| 1 | 4.60 | 4.55 | 4.56 | 4.57 | 4.57 |
| 2 | 4.53 | 4.53 | 4.54 | 4.54 | 4.535 |
| 3 | 4.53 | 4.53 | 4.57 | 4.54 | 4.5425 |
| 4 | 4.54 | 4.54 | 4.64 | 4.58 | 4.575 |

Πίνακας 2: Static Schedule Πείραμα

**Σειριακός Χρόνος Προγράμματος** = 13.56 sec

Με βάση τον παραπάνω πίνακα το αποτέλεσμα που προκύπτει είναι η εξής γραφική παράσταση:



Εικόνα 1: Static Schedule Γράφημα

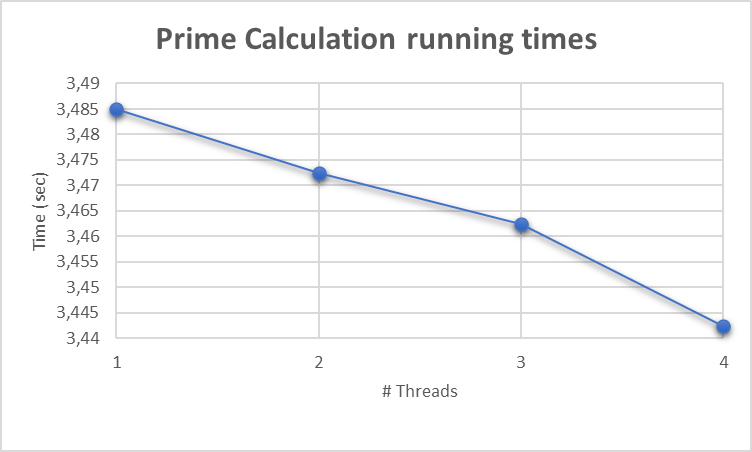
Το δεύτερο πείραμα έχει ***schedule dynamic***. Οι μετρήσεις φαίνονται παρακάτω:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Νήματα** | **1η Εκτέλεση** | **2η Εκτέλεση** | **3η Εκτέλεση** | **4η Εκτέλεση** | **Μέσος Χρόνος** |
| 1 | 3.47 | 3.56 | 3.44 | 3.47 | 3.485 |
| 2 | 3.44 | 3.45 | 3.55 | 3.45 | 3.4725 |
| 3 | 3.43 | 3.46 | 3.50 | 3.46 | 3.4625 |
| 4 | 3.46 | 3.43 | 3.44 | 3.45 | 3.445 |

Πίνακας 3: Static Schedule Πείραμα

**Σειριακός Χρόνος Προγράμματος** = 13.56 sec

Με βάση τον παραπάνω πίνακα το αποτέλεσμα που προκύπτει είναι η εξής γραφική παράσταση:



Εικόνα 2: Dynamic Schedule Γράφημα

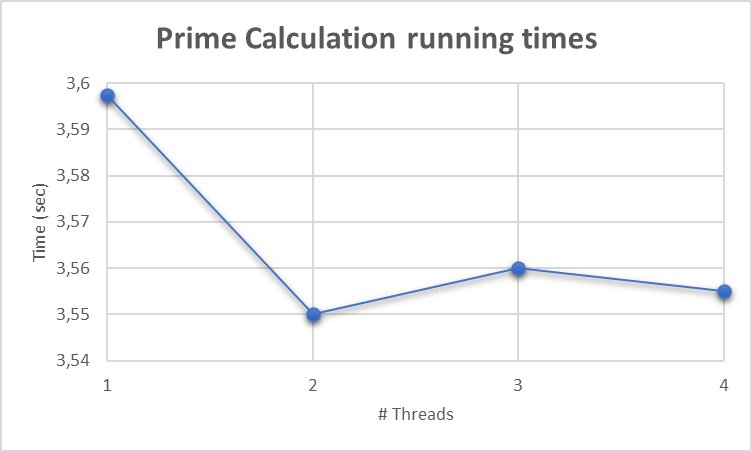
Το τρίτο πείραμα έχει ***schedule static*** με ***NumOfPckts = 5***. Οι μετρήσεις φαίνονται παρακάτω:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Νήματα** | **1η Εκτέλεση** | **2η Εκτέλεση** | **3η Εκτέλεση** | **4η Εκτέλεση** | **Μέσος Χρόνος** |
| 1 | 3.52 | 3.56 | 3.58 | 3.73 | 3.5975 |
| 2 | 3.51 | 3.64 | 3.53 | 3.52 | 3.55 |
| 3 | 3.44 | 3.53 | 3.53 | 3.75 | 3.56 |
| 4 | 3.50 | 3.55 | 3.55 | 3.62 | 3.555 |

Πίνακας 4: Static(NumOfPckts 5) Schedule Πείραμα

**Σειριακός Χρόνος Προγράμματος** = 13.56 sec

Με βάση τον παραπάνω πίνακα το αποτέλεσμα που προκύπτει είναι η εξής γραφική παράσταση:



Εικόνα 3: Static Schedule(NumOfPckts 5) Γράφημα

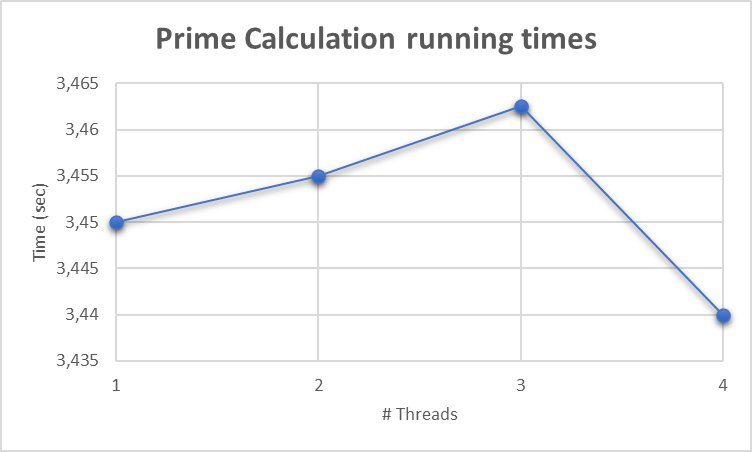
Το τέταρτο πείραμα έχει ***schedule dynamic*** με ***NumOfPckts = 5***. Οι μετρήσεις φαίνονται παρακάτω:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Νήματα** | **1η Εκτέλεση** | **2η Εκτέλεση** | **3η Εκτέλεση** | **4η Εκτέλεση** | **Μέσος Χρόνος** |
| 1 | 3.45 | 3.45 | 3.46 | 3.44 | 3.45 |
| 2 | 3.45 | 3.48 | 3.44 | 3.45 | 3.455 |
| 3 | 3.44 | 3.45 | 3.53 | 3.45 | 3.4625 |
| 4 | 3.48 | 3.43 | 3.42 | 3.43 | 3.44 |

Πίνακας 5: Dynamic(NumOfPckts 5) Schedule Πείραμα

**Σειριακός Χρόνος Προγράμματος** = 13.56 sec

Με βάση τον παραπάνω πίνακα το αποτέλεσμα που προκύπτει είναι η εξής γραφική παράσταση:



Εικόνα 4: Dynamic Schedule(NumOfPckts 5) Γράφημα

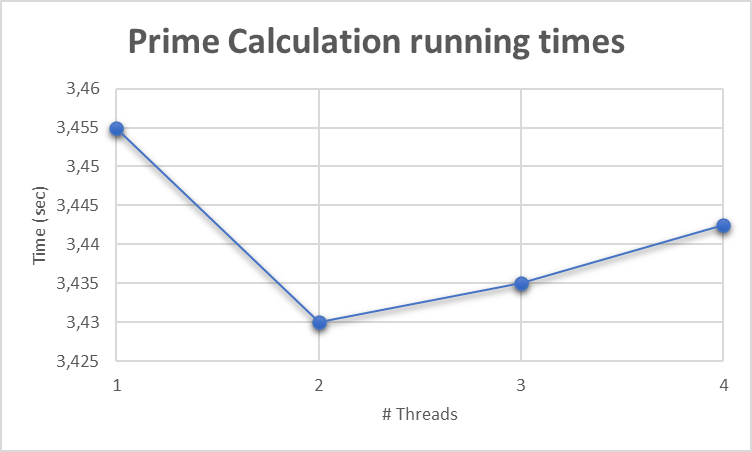
Το πέμπτο πείραμα έχει ***schedule guided*** με ***NumOfPckts = 5***. Οι μετρήσεις φαίνονται παρακάτω:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Νήματα** | **1η Εκτέλεση** | **2η Εκτέλεση** | **3η Εκτέλεση** | **4η Εκτέλεση** | **Μέσος Χρόνος** |
| 1 | 3.43 | 3.42 | 3.55 | 3.42 | 3.455 |
| 2 | 3.43 | 3.43 | 3.43 | 3.43 | 3.43 |
| 3 | 3.43 | 3.42 | 3.44 | 3.45 | 3.4535 |
| 4 | 3.44 | 3.44 | 3.44 | 3.45 | 3.4425 |

Πίνακας 6: Guided(NumOfPckts 5) Schedule Πείραμα

**Σειριακός Χρόνος Προγράμματος** = 13.56 sec

Με βάση τον παραπάνω πίνακα το αποτέλεσμα που προκύπτει είναι η εξής γραφική παράσταση:



Εικόνα 5: Guided(NumOfPckts 5) Schedule Γράφημα

## **2.4. Σχόλια:**

# **3. Άσκηση-2:**

## **3.1. Το πρόβλημα:**

## **3.2. Μέθοδοι Παραλληλοποίησης:**

## **3.3. Πειραματικά Αποτελέσματα – Μετρήσεις:**

## **3.4. Σχόλια:**

# **4. Άσκηση-3:**

## **4.1. Το πρόβλημα:**

## **4.2. Μέθοδοι Παραλληλοποίησης:**

## **4.3. Πειραματικά Αποτελέσματα – Μετρήσεις:**

## **4.4. Σχόλια:**

# **Βιβλιογραφία**

1. Β. Β. Δημακόπουλος, “*Παράλληλα Συστήματα και Προγραμματισμός*”, (1η Αναθεωρημένη Έκδοση), Εκδόσεις Κάλλιπος, 2017.
2. *OpenMP API 4.5 Complete Specifications,* November 2015.
3. *OpenMP API 4.5 Reference Guide – C/C++*, November 2015.
4. P. Pacheco, “*An Introduction to Parallel Programming*”, Morgan Kaufmann Publishers - Elsevier, 2015.
5. R. Chandra, L. Dagun, D. Kohr, D. Maydan, J. McDonald, R. Menon, “*Parallel Programming in OpenMP*”*,* Morgen Kaufmann Publishers,2001.