

ΕΡΓΑΣΙΑ 1

Οι εργασίες του μαθήματος θα είναι θεωρητικές και πρακτικές. Οι πρακτικές (προγραμματιστικές) εργασίες αφορούν σε εφαρμογές πολυνηματικού (multithreaded) προγραμματισμού πρέπει να υλοποιηθούν σε Python. Οι φοιτητές καλούνται να μελετήσουν τις διαφάνειες που υπάρχουν στην ιστοσελίδα του μαθήματος, τις ασκήσεις, καθώς και τα σχετικά tutorial.

Επισημαίνεται ότι οι εργασίες αυτές δεν είναι υποχρεωτικές. Παρόλα αυτά, οι φοιτητές που θα υποβάλλουν εργασίες θα ανταμειφθούν με ένα μπόνους στον τελικό τους βαθμό. Πιο συγκεκριμένα, η επιτυχής ολοκλήρωση κάθε εργασίας θα αποφέρει μπόνους μίας μονάδας. Συνεπώς, η επιτυχής υποβολή δύο/τριών σωστών εργασιών θα αποφέρει συνολικό μπόνους δύο/τριών μονάδων αντίστοιχα.

Ως προθεσμία υποβολής για όλες τις εργασίες ορίζεται η 31/12/2020.

ΘΕΜΑ 1

Ένας web server με μία CPU τρέχει τους παρακάτω τύπους προγραμμάτων με τις αντίστοιχες δυνατότητες παραλληλισμού:

Τύπος προγράμματος	Δυνατότητα παραλληλισμού
A	80%
B	60%
C	50%

Επιθυμούμε να αναβαθμίσουμε την επεξεργαστική ισχύ του server ώστε να μπορούμε τουλάχιστον να τετραπλασιάσουμε την ταχύτητα εκτέλεσης των προγραμμάτων αντικαθιστώντας την υπάρχουσα CPU με N παράλληλους επεξεργαστές.

α. Πόσους επεξεργαστές χρειαζόμαστε;

β. Τί επιτάχυνση επιτυγχάνεται για κάθε τύπο προγράμματος με βάση την απάντηση του υποερωτήματος (α).

ΘΕΜΑ 2

Υποθέστε ότι ένα πρόγραμμα πρέπει να εκτελέσει 10^{12} instructions για να λύσει ένα συγκεκριμένο πρόβλημα. Υποθέστε περαιτέρω ότι ένα σύστημα με έναν επεξεργαστή μπορεί να λύσει το πρόβλημα σε 10^6 δευτερόλεπτα (περίπου 11,6 ημέρες). Έτσι, κατά μέσο όρο, το σύστημα αυτό εκτελεί 10^6 , δηλαδή ένα εκατομμύριο instructions ανά δευτερόλεπτο.

Τώρα υποθέστε ότι το πρόγραμμα έχει παραλληλισθεί για εκτέλεση σε ένα σύστημα κατανεμημένης μνήμης. Υποθέστε επίσης ότι αν το παράλληλο πρόγραμμα χρησιμοποιεί p επεξεργαστές, κάθε επεξεργαστής θα εκτελέσει $10^{12}/p$ instructions και κάθε επεξεργαστής πρέπει να στείλει $10^9(p-1)$ μηνύματα. Τέλος, ας υποθέσουμε ότι δεν υπάρχει επιπλέον επιβάρυνση κατά την εκτέλεση του παράλληλου προγράμματος, δηλαδή το πρόγραμμα θα ολοκληρωθεί αφού ο κάθε επεξεργαστής έχει εκτελέσει όλες τις οδηγίες του και έχει στείλει όλα τα μηνύματά του και δεν θα υπάρξουν καθυστερήσεις λόγω αναμονής για μηνύματα.

α. Υποθέστε ότι για την αποστολή ενός μηνύματος απαιτούνται 10^{-9} δευτερόλεπτα. Πόσο χρόνο θα χρειαστεί το πρόγραμμα αν τρέχει με 1000 παράλληλους επεξεργαστές, αν κάθε επεξεργαστής έχει ταχύτητα όση και ο μοναδικός επεξεργαστής στον οποίο εκτελείται το σειριακό πρόγραμμα;

β. Ας υποθέσουμε ότι χρειάζονται 10^{-3} δευτερόλεπτα για την αποστολή ενός μηνύματος. Πόσο χρόνο θα χρειαστεί να τρέξει το πρόγραμμα με 1000 επεξεργαστές;

ΘΕΜΑ 3

Πρέπει να παραδώσετε μια εργασία για κάποιο μάθημα ως τη Δευτέρα 1 Ιουνίου 2020. Για να γνωρίζετε πόσο χρόνο έχετε μέχρι την προθεσμία αναπτύξτε ένα πρόγραμμα στην Python.

Το πρόγραμμα θα λαμβάνει υπόψη την προθεσμία και θα επιστρέφει τον χρόνο που απομένει από τώρα μέχρι την προθεσμία. Επεκτείνετε το πρόγραμμα έτσι ώστε να επιστρέφει τον υπόλοιπο χρόνο με τις ακόλουθες μορφές: (i) μόνο ώρες, (ii) μόνο λεπτά και (iii) μόνο δευτερόλεπτα.

[Hint: για να βρίσκετε την τωρινή ώρα χρησιμοποιήστε το `datetime` module της Python]

ΘΕΜΑ 4

Θέλετε να υπολογίσετε πόσα χρήματα κερδίζετε ανά ημέρα γνωρίζοντας τον ετήσιο μισθό σας. Η μόνη ανησυχία που έχετε είναι εάν το έτος είναι δίσεκτο ή όχι.

Αναπτύξτε ένα πρόγραμμα που θα λαμβάνει ως input από το πληκτρολόγιο το έτος και τον ετήσιο μισθό, θα αποφασίζει αν το έτος είναι δίσεκτο και στη συνέχεια θα υπολογίζει το χρηματικό ποσό ανά ημέρα για το συγκεκριμένο έτος.