



Όνομα: Δημήτρης Βάσιος
Α.Μ. : el19404

Όνομα: Παύλος Καζάκος
Α.Μ. : el18403

Άσκηση 1

(1)

Χρησιμοποιώντας την εντολή `time` για να μετρήσουμε τον χρόνο εκτέλεσης των εκτελέσιμων διαπιστώσαμε ότι ο χρόνος εκτέλεσης για τα προγράμματα με συγχρονισμό είναι μεγαλύτερος. Αυτό συμβαίνει διότι κάθε φορά που εκτελείται μία πράξη από ένα νήμα, είτε αύξηση του κοινού αριθμού είτε μείωση του, το άλλο νήμα περιμένει να εκτελεστεί η πράξη και στη συνέχεια να ενεργήσει πάνω στον αριθμό. Αυτό αυξάνει τον χρόνο εκτέλεσης των προγραμμάτων.

(2)

Μπορούμε να διαπιστώσουμε πολύ εύκολα με την εντολή `time` ότι η μέθοδος συγχρονισμού με ατομικές λειτουργίες είναι γρηγορότερη από αυτή με την χρήση POSIX Mutexes. Αυτό συμβαίνει καθώς οι ατομικές εντολές υλοποιούνται σε επίπεδο υλικού ενώ τα Mutexes υλοποιούνται με τη βοήθεια του λειτουργικού συστήματος και χρειάζονται περισσότερο χρόνο.

(3)

Παρακάτω φαίνεται ο κώδικας σε επίπεδο assembly με τον οποίο υλοποιούμε τα atomic operations:

```
lock addq  
$1, -16($rbp)
```

```
lock subq  
$1, -16(%rbp)
```

(4)

Παρακάτω φαίνεται ο κώδικας σε επίπεδο assembly με τον οποίο υλοποιούμε τα POSIX Mutexes:

```
movl  $mutex1, %edi  
call  pthread_mutex_lock  
testl  %eax, %eax  
je     .L3
```

```
movl  $mutex1, %edi  
call  pthread_mutex_unlock  
testl  %eax, %eax  
je     .L4
```



Άσκηση 2

(1)

Για το σχήμα συγχρονισμού που υλοποιούμε χρειάζονται τόσοι σεμοφόροι όσα είναι τα threads. Δημιουργούμε ένα πίνακα από σεμαφόρους που κάθε έναν από αυτούς τους ενεργοποιούμε κυκλικά, δηλαδή μόλις ένα νήμα εκτυπώσει μία γραμμή κάνουμε post στον αμέσως επόμενο σεμαφόρο, ανεβαίνει κατά 1 και εκτυπώνεται η επόμενη γραμμή. Αυτή η διαδικασία συνεχίζεται έως ότου εκτυπωθούν όλες οι γραμμές.

(2)

Ο χρόνος που απαιτείται με τον σειριακό υπολογισμό είναι ο εξής: 0.997s

Ο χρόνος που απαιτείται με τον παράλληλο υπολογισμό για δύο νήματα είναι ο εξής: 0.504s

Παρατηρούμε ότι ο χρόνος έπεσε ακριβώς στο μισό, κάτι που είναι απόλυτα λογικό καθώς τα νήματα καταμερίζονται παράλληλα σε δύο πυρήνες.

(3)

Το κρίσιμο τμήμα που έχουμε υλοποιήσει είναι αυτό της εκτύπωσης των γραμμών. Το πρόβλημα που υπάρχει με την συγκεκριμένη υλοποίηση είναι ότι κάθε φορά που εκτυπώνεται μία γραμμή πρέπει να περιμένει την εντολή post από άλλο νήμα για να προχωρήσει με την φάση υπολογισμού. Αυτό καθυστερεί σε ένα βαθμό το πρόγραμμά μας, αλλά παραμένει πιο αποτελεσματικό από την σειριακή υλοποίηση.

(4)

Το ctrl-c αυτό που κάνει είναι να στέλνει ένα σήμα SIGINT στη διεργασία που τρέχει. Αυτό σταματάει την διεργασία. Θα μπορούσαμε να διαχειριστούμε αυτή τη κατάσταση με τη προσθήκη συναρτήσεων που διαχειρίζονται τα σήματα και εξασφαλίζουν ότι το τερματικό θα επαναφερθεί στη προηγούμενη κατάστασή του.