Ενσωματωμένα Συστήματα Πραγματικού Χρόνου

Εργασία 1

Αντώνης Φάββας, ΑΕΜ:8675, e-mail : afavvask@auth.gr

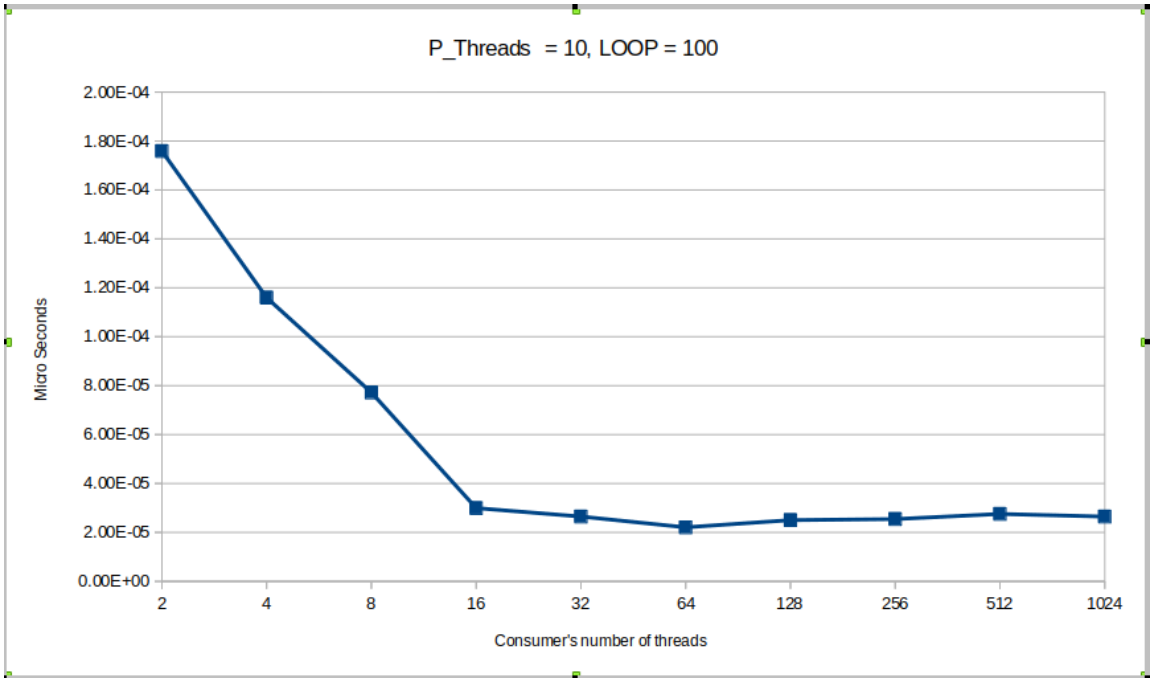
Στην παρούσα εργασία κληθήκαμε να τροποποιήσουμε τον συγκεκριμένο δοσμένο αλγόριθμο της εκφώνησης, ο οποίος σαν βασικές λειτουργίες έχει τις εξής:

- Δημιουργεί 2 νήματα, το ένα εκ των οποίων προσθέτει τιμές ακεραίων σε μία ουρά (producer), και το άλλο έπειτα λαμβάνει αυτές τις τιμές από την ουρά (consumer).
- Η ουρά έχει συγκεκριμένη δομή (FIFO), κατά την οποία όποιο στοιχείο εισέλθει πρώτο στην ουρά θα εξέλθει πρώτο και από αυτήν.

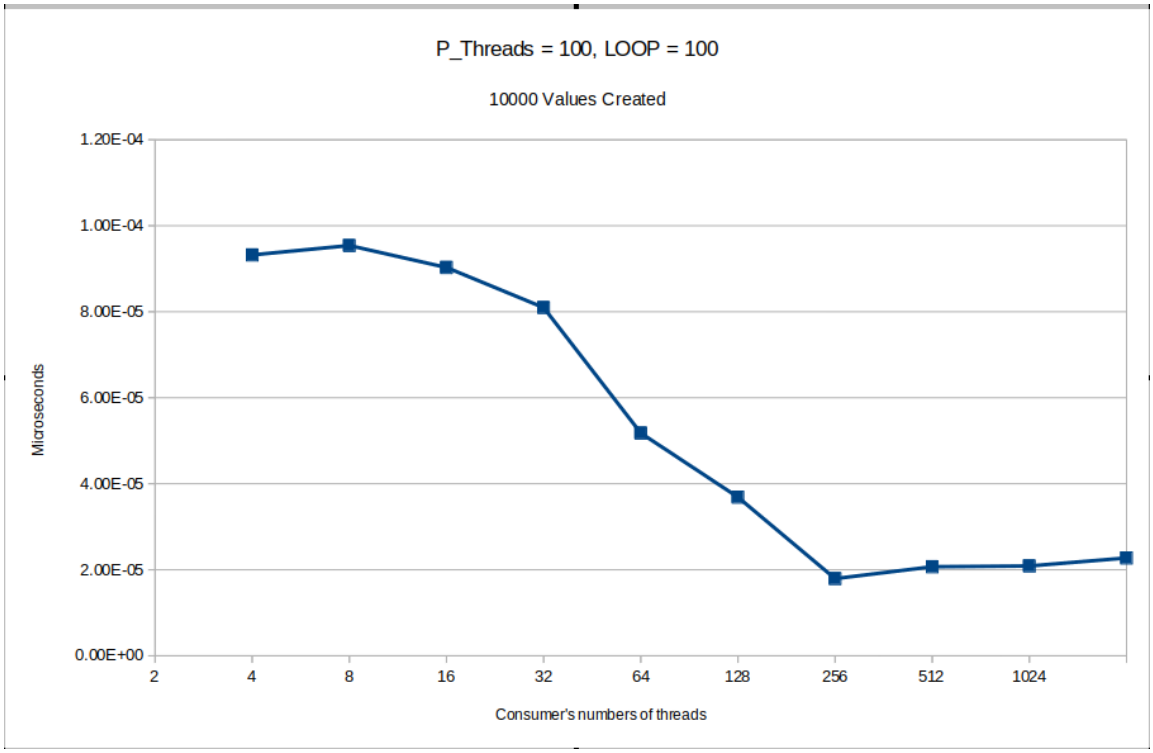
Οι βασικές τροποποιήσεις του κώδικα που πραγματοποιήθηκαν στα πλαίσια της εργσίας ήταν οι εξής:

- Δημιουργούμε πολλά νήματα Producer και Consumer αριθμού **pn** και **qn** αντίστοιχα που καθορίζονται από τον χρήστη.
- Στην κάθε «κελί» πλέον της ουράς δεν αποθηκεύονται τιμές ακεραίων, αλλά ένα ολόκληρο struct τύπου **workFunction.**
- Ο αριθμός των στοιχείων που αποθηκεύονται στην ουρά καθορίζεται από την πράξη: **LOOP*Pn**.
- Το struct τύπου **workFunction** αποτελείται από 2 στοιχεία. Το πρώτο είναι ένας Pointer to function που θα χρησιμοποιηθεί για να καλέσουμε συνάρτηση που εκτελεί μία αρκετά «απλοϊκή» πράξη. Το δέυτερο στοιχείο είναι ένας Void pointer που χρησιμοποιούμε για να καθορίσουμε την αρχή του χρόνου μέτρησης κατά την οποία το συγκεκριμένο στοιχείο struct αποθηκέυτηκε στην ουρά.
- Η μέτρηση του πειράματος ολοκληρώνεται όταν το συγκεκριμένο στοιχείο struct (που περιγράφηκε παραπάνω) λαμβάνεται από την ουρά από ένα οποιοδήποτε thread Consumer. Η μέτρηση του χρόνου έγινε χρησιμοποιώντας την συνάρτηση **gettimeofday()**.
- Τα αποτελέσματα των μετρήσεων χρόνου αποθήκέυονται μέσα σε ένα αρχείο με επέκταση .txt, το όνομα του οποίου προκύπτει παραμαμετροποιημένο ανάλογα με τους αριθμούς **pn**, **qn**.
- Χρησιμοποιώντας το πρόγραμμα Octave λαμβάνουμε τον μέσο όρο των μετρήσεων χρόνου για κάθε εκτέλεση του κώδικα με διαφορετικές παραμέτρους.
- Τέλος τρέχουμε το πρόγραμμα για διαφορετικές τιμές του **pn**,**qn** και θέλουμε να εξετάσουμε κυρίως την επίδραση του αριθμού των νημάτων Consumer στις μετρήσεις του χρόνου.

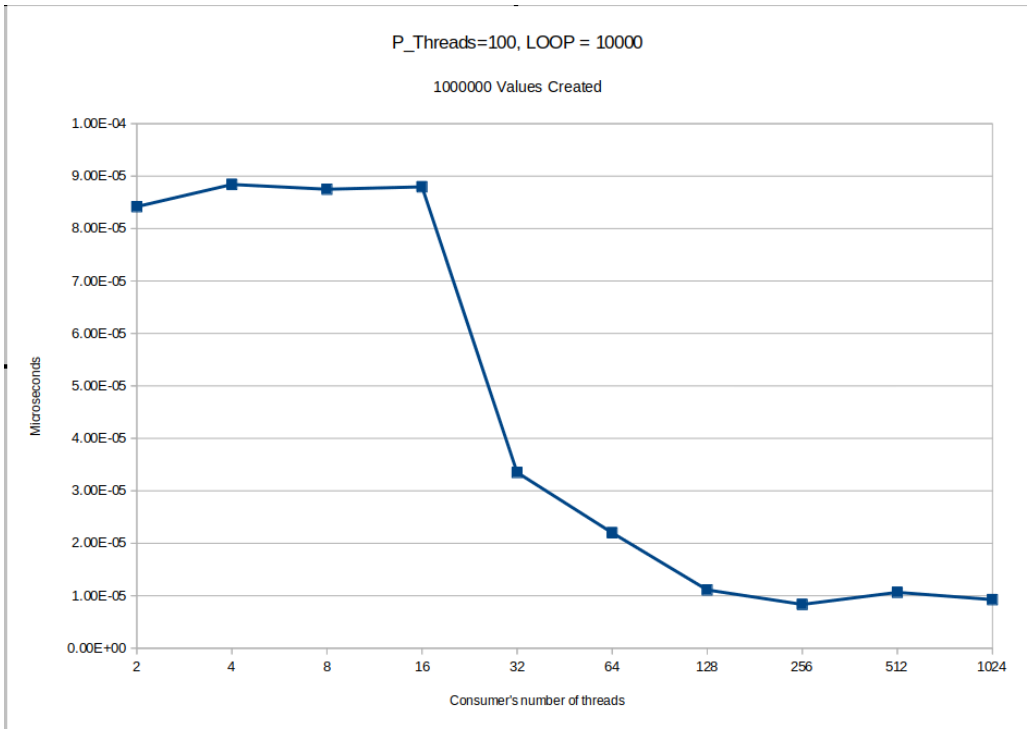
Παρακάτω παραθέτονται τα αποτελέσματα του μέσου χρόνου μέτρησης, για διάφορες εκτελέσεις του κώδικα με διαφορετικές παραμέτρους **pn**, **qn**.



Εικόνα 1.



Εικόνα 2.



Εικόνα 3.

Συμπέρασμα.

Τα πειραματικά αποτελέσματα σχεδόν αποτυπώνουν και επαληθέυουν την θεωρητική υπόθεση η οποία διατυπώνεται ως εξής:

Όσο αυξάνει ο αριθμός των νημάτων που «τραβάνε» στοιχεία από την ουρά, τόσο θα μειώνεται και ο χρόνος που ένα στοιχείο θα παραβρίσκεται σε αυτήν. Χαρακτηριστικά στην εικόνα 1 έχουμε τον ελάχιστο μέσο χρόνο μέτρησης για **qn** (αριθμός νημάτων Consumer) ίσο με 64, ενώ στις εικόνες 2 και 3 έχουμε ελαχιστοποίηση του ίδιου χρόνου για qn ίσο με 256. Ωστόσο αυτό (όπως φαίνεται και από τις εικόνες) δεν συνεπάγεται ότι όσο αυξάνουμε τον αριθμό **qn** θα έχουμε πάντα καλύτερη απόδοση, καθώς πολλές φορές ο χρόνος που απαιτείται για την δημιουργία ενός νήματος είναι μεγαλύτερος, σε σχέση με τον χρόνο που εξοικονομούμε απο την χρήση του (overhead).

Ο τροποποιημένος κώδικας υπάρχει εδώ: https://github.com/antonis-fav/-.git