

# Αναφορά - Εργασία 1/3 ΕΣΠΧ

Ραφαήλ Μπουλογεώργος (9186) – 29/3/20

## ΣΥΝΔΕΣΜΟΣ ΕΡΓΑΣΙΑΣ – GITHUB

<https://github.com/rafampou/Real-Time-Embedded-Systems>

Περιέχει τον κώδικα `prod-cons_9186.c` τα δεδομένα των δοκιμών `data{4,32,64}_9186.txt` το `script.sh` που τρέξαμε τις δοκιμές και το `prod-cons_9186.m` για την επεξεργασία των αποτελεσμάτων στο matlab

## ΣΥΝΑΡΤΗΣΗ WORK

Επεκτείνοντας το βασικό παράδειγμα `prod-cons.c` η ουρά FIFO κρατάει δεδομένα τύπου struct με όνομα `workFunction`. Ως συνάρτηση ορίζουμε την `workfunc`, η οποία είναι σταθερή και ορισμένη για όλα τα στοιχεία της ουράς, αλλά ο όγκος εργασία εξαρτάται από το όριο της συνάρτησης.

Για συγκεκριμένα η ουρά υπολογίζει το συννημίτονο  $10+b$  γωνιών με  $b$ , το όριο της συνάρτησης. Οι γωνίες που υπολογίζονται είναι πολλαπλάσια του  $3,14+b$ .

Το όριο της συνάρτησης ορίζεται όταν ένα στοιχείο μπαίνει στην ουρά και είναι ο αύξον αριθμός  $i$  στον βρόγχο του κάθε producer.

## ΣΤΑΤΙΣΤΙΚΗ ΑΝΑΛΥΣΗ ΑΝΑΜΟΝΗΣ

Για την λήψη των αποτελεσμάτων έγιναν δοκιμές με τα εξής δεδομένα

- Μέγεθος ουράς `QUEUE_SIZE = 20`
- Επαναλήψεις για κάθε producer `LOOP = 40,20`
- Αριθμός producers  $P = 4,32,64$
- Αριθμός consumers  $Q = [1\ 2\ 4\ 16\ 32\ 64\ 128\ 256\ 512\ 1024\ 2048\ 2096]$

Για κάθε αριθμό  $Q$ , επαναλάβαμε την εκτέλεση **200 φορές**.

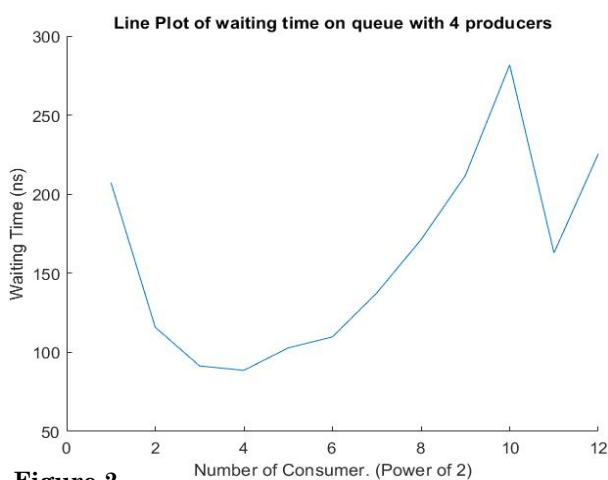


Figure 2

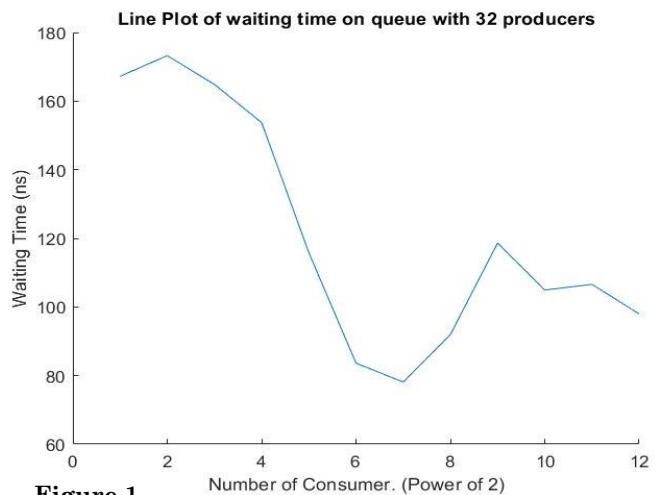
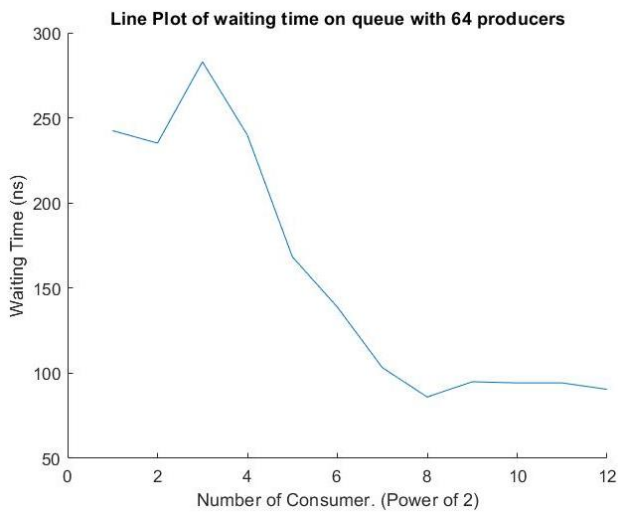


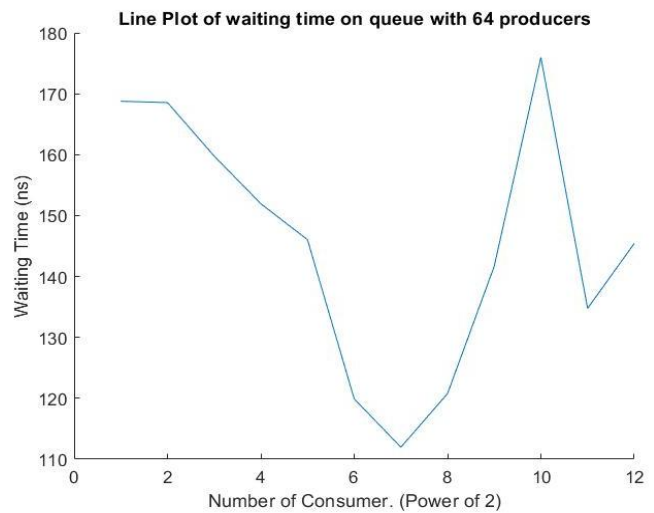
Figure 1

Στα διαγράμματα Figure 1,2,4 βλέπουμε την μέση τιμή των χρόνων για 4,32 και 64 producer threads ως προς τον αριθμό των consumer threads. Κάθε producer εισάγει στην ουρά 40 εργασίες. Για 4 Producer η ελάχιστη μέση αναμονή είναι περίπου 90nsec με  $2^4 = 16$  Consumer threads, ενώ για 32 producer έχουμε συνολικά λιγότερο χρόνο αναμονής με βέλτιστο αριθμό consumer  $2^7 = 128$ . Τέλος για 64 Producers έχουμε 128 Consumer

Μπορούμε να καταλήξουμε στο συμπέρασμα ότι χρειαζόμαστε **4πλάσιο** αριθμό καταναλωτών σε σχέση με τους παραγωγούς για το βέλτιστο χρόνο με αυτά τα δεδομένα.



**Figure 3: 64 Producers with loop=40**



**Figure 4: 64 Producers with loop=20**

Συγκρίνοντας τα figure 3,4 βλέπουμε ότι ο βέλτιστος αριθμός consumer δεν είναι συνάρτηση μόνο των producer αλλά και του LOOP, δηλαδή των εργασιών που βάζει στην ουρά κάθε παραγωγός.

Βλέπουμε ότι για

- $loop = 40$  έχουμε  $2^8 = 256 = 64 * \frac{40}{10}$
- ενώ για  $loop = 20$  έχουμε  $2^7 = 128 = 64 * \frac{20}{10}$

Μπορούμε έτσι να καταλήξουμε σε ένα γενικό συμπέρασμα από τα παραπάνω αποτελέσματα

**Ο βέλτιστος αριθμός νημάτων Consumer είναι  $q = p * \frac{loop}{10}$ .**

Στο διάγραμμα κουτιού που ακολουθεί, για 64 παραγωγούς και loop=40 επιβεβαιώνουμε την βέλτιστη λύση καθώς για 256 καταναλωτές έχουμε τον μικρότερο μέσο όρο σε συνδυασμό με την μικρότερη διακύμανση.

Παρόλα αυτά και οι χρόνοι για 512 ή περισσότερους καταναλωτές είναι πολύ κοντά, όμως στο σύστημα που έγιναν οι μετρήσεις υπήρξαν και καθυστερήσεις που αυξάνουν τον μέσο όρο

