

Να διαμορφωθεί το παράδειγμα `prod-cons.c` ώστε

1. Η `FIFO` ουρά να κρατά στοιχεία με τύπο

```
struct workFunction {  
    void * (*work) (void *);  
    void * arg;  
}
```

2. Να λειτουργεί με `p` νήματα `producer` που βάζουν στην **κοινή** ουρά δείκτες σε συναρτήσεις με την παραπάνω δομή, και `q` νήματα `consumer` που παίρνουν τους δείκτες από την ουρά, και εκτελούν τη συνάρτηση. Οι συναρτήσεις στην ουρά κάνουν κάτι απλό, πχ υπολογίζουν το ημίτονο για δέκα γωνίες, ή εμφανίζουν ένα μήνυμα στην οθόνη.
3. Μη χρησιμοποιείτε τη `sleep()`, δε χρειάζεται η τεχνητή καθυστέρηση, ούτε το δεύτερο βρόγχο. Κάνετε το πρώτο βρόγχο να επαναλαμβάνεται για ένα συγκεκριμένο μεγάλο αριθμό επαναλήψεων. Κάνετε το βρόγχο στο `consumer` while 1.

Να ετοιμάσετε μια αναφορά μέχρι 2 σελίδες όπου θα παρουσιαστούν οι στατιστικές του χρόνου αναμονής που μεσολαβεί από τη στιγμή που ένας `producer` βάζει στην ουρά, μέχρι ένας `consumer` να το παραλάβει αυτό, (αλλά όχι και να το εκτελέσει), σε έναν υπολογιστή με τουλάχιστον δύο πυρήνες. Χρησιμοποιείστε την συνάρτηση `gettimeofday()` για να βρείτε τη χρονική στιγμή.

Ποιος είναι ο αριθμός από νήματα `consumer` στο σύστημά σας που ελαχιστοποιεί τη μέση τιμή του χρόνου αναμονής;

Χρησιμοποιείστε ικανό αριθμό από πειράματα μέχρι να σταθεροποιηθούν οι μετρήσεις σας.

Η εργασία είναι ατομική. Σχόλια και ερωτήσεις στο forum του μαθήματος.

Το παράδειγμα `prod-cons.c` είναι [εδώ](#).

Πληροφορίες για την χρήση `pthread` μπορείτε να βρείτε [εδώ](#).