

HY-345 ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ
Εαρινό εξάμηνο 2018
Άσκηση 2

Το πρόβλημα του παραγωγού-καταναλωτή

Στην άσκηση αυτή καλείστε να υλοποιήσετε μια λύση για το πρόβλημα πολλαπλών παραγωγών-πολλαπλών καταναλωτών χρησιμοποιώντας νήματα (threads) και σημαφόρους. Κάθε παραγωγός και κάθε καταναλωτής θα είναι ένα νήμα στη ίδια διεργασία.

Το απλό πρόβλημα παραγωγού-καταναλωτή περιγράφεται στη σελίδα 172 του βιβλίου (TANENBAUM 3η έκδοση). Η διαφορά του πολλαπλού προβλήματος από το απλό είναι ότι αντί ενός παραγωγού και ενός καταναλωτή, υπάρχουν ταυτόχρονα πολλοί - η λύση όμως παραμένει η ίδια. Στην περίπτωση μας ο κάθε παραγωγός και καταναλωτής είναι ξεχωριστά νήματα μέσα στην ίδια διεργασία, τα οποία μοιράζονται μια κοινή μνήμη. Ο παραγωγός παράγει δεδομένα και τα εισάγει στην μνήμη και ο καταναλωτής διαβάζει τα δεδομένα από την μνήμη. Το πρόβλημα είναι να μην επιτραπεί στους παραγωγούς να προσθέσουν δεδομένα στην κοινή μνήμη εάν αυτή είναι γεμάτη, να μην επιτραπεί στους καταναλωτές να αφαιρέσουν δεδομένα όταν η κοινή μνήμη είναι κενή, και τέλος τόσο οι παραγωγοί όσο και οι καταναλωτές θα πρέπει να βεβαιωθούν ότι η πρόσβαση στην κοινή μνήμη δεν δημιουργεί συνθήκες ανταγωνισμού (race conditions).

Για την υλοποίηση θα χρησιμοποιήσετε system calls της ομάδας pthreads (POSIX Threads) για τη δημιουργία κάθε παραγωγού/καταναλωτή, και σημαφόρους για τον συγχρονισμό τους.

Το πρόγραμμά σας θα πρέπει να δέχεται από την γραμμή εντολών τις παρακάτω παραμέτρους:

- p, το πλήθος των παραγωγών
- c, το πλήθος των καταναλωτών

Όλοι οι παραγωγοί και οι καταναλωτές θα επικοινωνούν με την ίδια κοινόχρηστη μνήμη. Κάθε παραγωγός θα περιμένει για ένα τυχαίο χρονικό διάστημα μεταξύ 1-10 microseconds, και θα παράγει έναν τυχαίο αριθμό μεταξύ 1-100 που θα εισάγει στην μνήμη. Κάθε παραγωγός θα σταματάει αφού θα έχει παράξει 1 εκατομμύριο αριθμούς. Κάθε καταναλωτής θα καταναλώνει όσο πιο γρήγορα μπορεί. Το πρόγραμμα θα τυπώνει το σύνολο όλων των αριθμών που καταναλώθηκαν από κάθε consumer.

Παράδειγμα εκτέλεσης με 10 producers και 2 consumers :

```
>cp -p 10 -c 2  
Consumer1: 250,000,001  
Consumer2: 250,000,002
```

Η μεταγλώττιση πρέπει να γίνει με την παράμετρο `-pthread` ώστε ο linker να μπορεί να βρει τα σύμβολα στην pthread βιβλιοθήκη.

Παρατηρήσεις

1. Η άσκηση είναι ατομική. Τυχόν αντιγραφές μπορούν να ανιχνευθούν εύκολα από κατάλληλο πρόγραμμα και θα μηδενιστούν. Συμπεριλάβετε το όνομα σας, τον αριθμό μητρώου σας, και το username σας σε όλα τα αρχεία.
2. Κατασκευάστε ένα αρχείο Makefile, έτσι ώστε πληκτρολογώντας `make all` να γίνεται η μεταγλώττιση (compilation) του προγράμματος και να παράγεται το εκτελέσιμο αρχείο. Επίσης πληκτρολογώντας `make clean` να καθαρίζονται όλα τα περιττά αρχεία, και να μένουν μόνο τα αρχεία που χρειάζονται για τη μεταγλώττιση.
3. Γράψτε ένα αρχείο README.txt, το πολύ 30 γραμμών, με επεξηγήσεις για τον τρόπο υλοποίησης.
4. Τοποθετήστε σε ένα κατάλογο όλα τα αρχεία που χρειάζονται για την άσκηση 2. Παραδώστε τα παραπάνω αρχεία χρησιμοποιώντας το πρόγραμμα `turnin`(πληκτρολογήστε `turnin assignment_2@hy345 directory_name` από τον κατάλογο που περιέχει τον κατάλογο `directory_name` με τα αρχεία της άσκησης)
5. Το πρόγραμμά σας μπορεί να γραφεί μόνο για λειτουργικά συστήματα που υπάρχουν στα εργαστήρια και σε γλώσσα προγραμματισμού C χρησιμοποιώντας `system calls` της ομάδας pthreads (POSIX Threads).