

# READ ME

Όνομα: Βασιλοπούλου Θωμαΐς  
Α.Μ: 1115201500016

Εντολή compile: make  
Εντολή εκτέλεσης: ./project3 10 60

Η υλοποίηση περιλαμβάνει τις εξής δομές και τύπους δεδομένων:

1. Shared memory segment:

Η δομή που μοιράζονται οι producers με τον consumer. Περιλαμβάνει μεταβλητές για τις συναλλαγές μεταξύ τους, αλλά και τους σημαφόρους full και empty.

2. Statistics:

Μια δομή με συγκεντρωμένες τις μεταβλητές που αναγράφονται στο τέλος της εκφώνησης προς εκτύπωση.

3. Threads:

Η βασικότερη δομή της υλοποίησης σε συνδυασμό με το shared memory segment, threads τύπου producer και consumer, μαζί με τον mutex.

Η υλοποίηση ξεκινάει με την εισαγωγή των στοιχείων της εκτέλεσης. Αφού ο χρήστης επιβεβαιώσει την ορθότητα των δεδομένων ή επαναλάβει την εισαγωγή, το σύστημα αρχικοποιεί τις δομές (1), (2) με τα κατάλληλα keys και τιμές σημαφόρων.

Στη συνέχεια δημιουργούνται τα thread. Το σύστημα κάνει sleep για δευτερόλεπτα, προκειμένου να ξεκινήσουν τα threads να τρέχουν.

Αφού η εκτέλεση τους ολοκληρωθεί, ελευθερώνεται η δεσμευμένη μνήμη, εκτυπώνονται τα στατιστικά των “συναλλαγών” και το πρόγραμμα τερματίζει.

## PRODUCER

Η εκτέλεση του producer thread θα συνεχίσει για όσο υπάρχουν ακόμα προϊόντα προς παραγωγή. Για αρχή, αποκτά access στο shared memory και διαβάζει από κάποιο τυχαίο αρχείο κειμένου (εδώ το text.txt) κάποια τυχαία γραμμή και την αποθηκεύει στο inText (δομή για αποστολή producer -> consumer) του shared memory segment.

Αν έχει ήδη λάβει κάποιο μήνυμα από τον consumer, ενημερώνει ένα flag ότι ο consumer εξακολουθεί να αποκρίνεται. Σπάει το μήνυμα του consumer (βλ. παρακάτω) στο md5string, που είναι η κωδικοποιημένη προαναφερθείσα γραμμή σε md5 και στο id του producer thread που έστειλε τη γραμμή στον consumer. Ανάλογα με το εάν το thread-αποστολέας ταυτίζεται με το thread παραλήπτης, ενημερώνονται τα στατιστικά και τυπώνονται τα κατάλληλα μηνύματα πριν κλείσει η πρόσβαση από το τρέχον thread στο shared memory.

Για κάθε προϊόν, το πρόγραμμα ενημερώνει το χρήστη πόσα προϊόντα να αναμένει ακόμη.

## CONSUMER

Η εκτέλεση του consumer thread θα συνεχίσει για όσο υπάρχουν ακόμα προϊόντα προς κατανάλωση. Για αρχή, αποκτά access στο shared memory και με βάση τις συναρτήσεις της md5 υπολογίζει για τη γραμμή που έλαβε την hash κωδικοποίηση. Στη συνέχεια κάνει concatenate το id του producer-αποστολέα με το hashαρισμένο μήνυμα, μειώνει τον αριθμό των προϊόντων που απομένουν και ξανακλείνει τη μνήμη.

Όταν η παραγωγή και η λήψη όλων των απαιτούμενων αντικειμένων ολοκληρωθεί, αποστέλεται το pthread\_cancel προκειμένου να τερματίσουν τα threads.

## REFERENCES

1. <http://openwall.info/wiki/people/solar/software/public-domain-source-code/md5>
2. <https://stackoverflow.com/questions/8465006/how-do-i-concatenate-two-strings-in-c>
3. [https://en.wikipedia.org/wiki/Characters\\_per\\_line](https://en.wikipedia.org/wiki/Characters_per_line)

## ΠΑΡΑΔΟΧΕΣ

1. Η ζητούμενη μεταβλητή `steps` ανάγεται στην πορεία της πληροφορίας που μεταφέρεται, καθώς αυτή ερμηνεία έβγαζε το περισσότερο νόημα.
2. Το `busy waiting` που περιλαμβάνεται είναι για την επιτυχία προβολής ~14000 αντικειμένων και προφανώς μπορεί να μειωθεί σημαντικά χωρίς πρόβλημα, παρ'όλα αυτά ο χρόνος εκτέλεσης είναι ιδιαίτερα μικρός, οπότε θεώρησα ότι δε θα αποτελέσει πρόβλημα.
3. Περιλαμβάνονται συναρτήσεις που υπήρχαν σε προηγούμενες εργασίες μου.
4. Η αρχική επιβεβαίωση είναι ιδιαίτερα "dummy" και υπάρχει για το καλύτερο διαδραστικό αποτέλεσμα.
5. Οι `setSignalHandler` και `catchInterrupt` υπάρχουν σε περίπτωση που είναι επιθυμητός ο εναλλακτικός χειρισμός του `SIGSEGV`.

Για οποιαδήποτε απορία, είμαι στη διάθεση σας.