

ONOMA: Πέτρος  
ΕΠΩΝΥΜΟ: Βασιλόπουλος  
Α.Μ.: 1115201000218

Το πρόγραμμα υλοποιεί όλα τα ζητούμενα της άσκησης. Πιο αναλυτικά, στην αρχή παίρνουμε σαν όρισμα το P και το K τα οποία είναι ο αριθμός των παραγωγών και ο αριθμός των μηνυμάτων που θα πάρει ο καταναλωτής αντίστοιχα. Δηλώνουμε δύο σημαφόρους και μία shared memory στην οποία της δίνουμε μνήμη ίση με το InMessage (in-ds), το OutMessage (out-ds) τα οποία είναι δομές μέσα στη shared memory και P\*int χώρο καθώς ο κάθε παραγωγός θα έχει το δικό του “ντουλαπάκι” για να κρατάμε το pid\_match που στην συνέχεια θα το επιστρέψει στην πατρική διεργασία. Έπειτα δημιουργούμε με fork() τους παραγωγούς και καλούμε για κάθε έναν από αυτούς την p\_main η οποία παίρνει σαν όρισμα τον αριθμό της διεργασίας που είναι και το id του, την διαμοιραζόμενη μνήμη και τους δύο σημαφόρους.

Η p\_main φτιάχνει δύο αντικείμενα InMessage και OutMessage και προσθέτει στο αντικείμενο της InMessage το pid της διεργασίας παραγωγός. Έπειτα, ανοίγει ένα αρχείο και διαβάζει όλες τις γραμμές για να δει πόσες είναι, διαβάζει μια τυχαία γραμμή και με το rewind(fp) ξαναγυρνάει τον κέρσορα στην αρχική του θέση και διαβάζει γραμμές απ την αρχή μέχρι να φτάσει στην ζητούμενη. Στη συνέχεια καλεί την queue\_in για να βάλει μέσα το αντικείμενο της InMessage που έχει το pid της διεργασίας αλλά και την γραμμή σε χαρακτήρες του αρχείου. Η queue\_put\_in κατεβάζει και ανεβάζει τους σημαφόρους για να κάνει ένα memcpy το αντικείμενο στο shared memory. Ύστερα καλεί την queue\_get\_out για να τραβήξει το αντικείμενο της OutMessage που έχει στείλει ο καταναλωτής και περιέχει τη χασαρισμένη γραμμή καθώς και το pid του παραγωγού που το δημιούργησε. Συγκρίνει το pid που πήρε με το δικό του και αν είναι ίδια τότε αυξάνει το pid\_match του. Όταν θα λάβει το signal για να τερματίσει θα καλέσει την exit\_p η οποία καλεί την queue\_write που γράφει το pid\_match στο ντουλάπι του συγκεκριμένου id δηλαδή στη θέση μνήμης pmem + sizeof (InMessage) + sizeof (OutMessage) + pos \* sizeof (int).

Η c\_main παίρνει σαν όρισμα το id του καταναλωτή που είναι 1, τη διαμοιραζόμενη μνήμη, τους δύο σημαφόρους, το K για το πόσα μηνύματα θα πάρει, τον πίνακα με τα pid των παραγωγών και τον αριθμό των παραγωγών. Φτιάχνει δύο αντικείμενα InMessage και OutMessage και έπειτα μέχρι να τελειώσει το K βγάζει μηνύματα απο την InMessage με την queue\_get\_in, έπειτα το pid που τράβηξε το βάζει στο αντικείμενο της OutMessage, χασάρει τη γραμμή που τράβηξε και την βάζει και αυτή στο αντικείμενο της OutMessage και τα στέλνει στην shared memory με το queue\_put\_out. Στο τέλος κάνει ένα μικρό usleep για να προλάβουν όλοι οι καταναλωτές να πάρουν τα μηνύματα πριν πεθάνουν και έπειτα στέλνει σήμα σε όλους τους παραγωγούς να τερματίσουν.

Η πατρική διεργασία περιμένει να τελειώσουν όλα τα παιδιά της και μόλις γίνει αυτό, παεί και κάνει memcpy με την queue\_read στα ντουλαπάκια των εκάστοτε pid δηλαδή στη διεύθυνση pmem = pmem + sizeof (InMessage) + sizeof (OutMessage) +

`pos * sizeof (int);` όπου  $1 < pos < P$

Κάνει sum τα `pid_match` και το εκυπώνει. Τέλος κάνει `detach` την shared memory καθώς και τους σημαφόρους και ελευθερώνει το `pid_p`.