Εντολή μεταγλώτισσης:

make

Εντολή εκτέλεσης:

exe 1ο\_όρισμα 2ο\_όρισμα

όπου 1ο\_όρισμα=αριθμός παραγωγών και 2ο\_όρισμα=αριθμός βημάτων

Προτείνεται να δίνεται μεγάλος αριθμός βημάτων ,

δηλαδή σίγουρα μεγαλύτερος απο τον αροθμό παραγωγών,

ετσι ώστε ο consumer να προλαβαίνει να γίνεται

διαθέσιμος, διότι αλλιώς καθυστερεί αρκετά να τρέξει.

Στο structs.h:

Περιλαμβάνεται το struct για την πρόταση που θα παρθεί απο το αρχείο,

με το id της και έναν ακέραιο useful, που θα παίρνει τιμές 1 ή 0, ανάλογα

με το αν είναι διαθέσιμος ο πίνακας ή όχι, αντιστοίχως.

Περιλαμβάνει επίσης το struct της διαμοιραζόμενης μνήμης με πίνακες

outds και inds καθώς και το pidmatch που αναφέρεται στην εκφώνηση.

Στο functions.h:

Περιλαβάνονται οι συναρτήσεις των σεμαφόρων up και down ,

καθώς και η συνάρτηση void removeallchars(char\* str, char c)

που πάρθηκε απο τον σύνδεσμο:

http://stackoverflow.com/questions/5457608/how-to-remove-the-character-at-a-given-index-from-a-string-in-c

για να αφαιρεί το '/n' από κάθε γραμμή του αρχείου (για αισθητικούς λόγους).

Στο main.c:

Γίνεται η υλοποίηση του ζητουμένου.

Αρχικά μετράω τις γραμμές που έχει το αρχείο για να πάιρνω τυχαίες κάθε φορά.

Έπειτα δημιουργώ την κοινή μνήμη, την κάνω attach και αρχικοποιώ τα στοιχεία της.

Κάθε θέση των inds και outds έχει προς το παρόν useful 0 , καθώς επίσης και το pidmatch

είναι 0.

Στη συνέχεια δημιουργώ σεμαφόρους που θα χρησιμοποιήσω με semget και τους αρχικοποιώ με semctl.

Διαβάζω από command line τα ορίσματα.

Δεσμεύω μνήμη για τους παραγωγούς και τον καταναλωτή και δημιουργώ τις διεργασίες με fork().

Οι διεργασίες βρίσκονται στον πινακα pids. Αυθαίρετα, ορίζω ο καταναλωτής, επειδή είναι ένας να βρίσκεται

στην 1η θέση του πίνακα(pids[0]).

Και έχω λοιπόν μια if (βρίσκομαι στον καταναλωτή), else if (βρίσκομαι στη γονική διεργασία, main, η οποία

κάνει και ό,τι ζητάει η εκφώνηση), else (βρίσκομαι σε κάποιον παραγωγό).

Αν βρίσκομαι στον καταναλωτή, αρχίζει η φάση διαβάσματος από την μνήμη. Γι' αυτο και κάνω down

στον σεμαφόρο indsfull και semshared. Ψάχνει να βρει διαθέσιμη θεση στον inds και βγαινει απο την επανάληψη

όταν βρει. Αποθηκέυει στο p τη γραμμή απο το αρχείο και ύστερα κάνει overwrite στον πίνακα για να βγάλει το

στοιχείο που πήρε. Με τη συνάρτηση toupper κεφαλαιοποιεί έναν έναν τους χαρακτήρες του p και κάνει up. Τέλος,

όταν λάβει τόσα Term όσα οι παραγωγοί, τότε αυτό είναι το ειδικό σήμα που θα ειδοποιήσει οτι οι επαναλ'ηψεις

τελέιωσαν.

Για τη φάση που ο καταναλωτής γράφει στην μνήμη, όσο δεν έχουν τελειώσει τα βήματα ,γράψε στην μνήμη και μείωσε τα,

αλλιώς γράψε το μήνυμα TERM στο p.

Αν βρίσκομαι σε κάποιον παραγωγό,στο γράψιμο στη μνήμη, με την fseek πηγαίνω πάνω στο αρχείο και παίρνω μια τυχαία

γραμμή του. Βγάζω το '/n' και συμπληρώνω τα πεδία του struct