

ΑΣΚΗΣΗ 2

HY-345 ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2024

Ανάθεση : 17/10

Παράδοση : 6/11

Ώρες γραφείου : Δευτέρα και Πέμπτη στις 14:00-16:00 στην αίθουσα B210

Περιγραφή

Σε αυτήν την άσκηση θα εξοικειωθείτε με τη δημιουργία και το χειρισμό threads, καθώς και semaphores (POSIX semaphores) για τον συγχρονισμό των διαφορετικών threads.

Άσκηση - Δρομολόγια προς σχολή

Κάθε πρωί, οι φοιτητές που σπουδάζουν Ηράκλειο πηγαίνουν προς την στάση της Χανιώπορτας (Stop A) και περιμένουν να έρθει το λεωφορείο για να πάνε στη σχολή τους.

Ο κάθε φοιτητής μπορεί να ανήκει:

- Στο Μαθηματικό
- Στο Φυσικό
- Στο Χημικό
- Στο CSD

Επιπλέον ο κάθε φοιτητής θα κάτσει στη σχολή για να παρακολουθήσει τα μαθήματά του και να διαβάσει για ένα ορισμένο χρονικό διάστημα ξεχωριστό για τον καθένα. Μόλις φτάσει στη στάση της σχολής (Stop B), μπει στη σχολή του και περάσει ο χρόνος αυτός, κατευθύνεται πάλι πίσω, προς την στάση του λεωφορείου της Πανεπιστημιούπολης και περιμένει ξανά το λεωφορείο για να γυρίσει πίσω στο σπίτι του.

Το λεωφορείο θα ξεκινάει από την στάση A (όπου εκεί θα αρχικοποιούνται και οι φοιτητές), και θα ξεκινάει για το Πανεπιστήμιο μόνο όταν δεν υπάρχουν άλλα άτομα στην στάση, ή δεν μπορεί να πάρει άλλα άτομα σύμφωνα με τους κανόνες που εξηγούνται παρακάτω.

Στο λεωφορείο υπάρχει χώρος για N άτομα. Επιπλέον, για να υπάρχει ισότητα μεταξύ των φοιτητών διαφορετικών σχολών, το λεωφορείο μπορεί να δεχτεί έως $(N/4)$ φοιτητές από κάθε σχολή. Αυτό σημαίνει πως για $N = 20$, αν υπάρχουν 5

φοιτητές του Φυσικού μέσα στο λεωφορείο, και ο επόμενος φοιτητής που θέλει να εισέλθει είναι από το Φυσικό, δεν θα μπορέσει να μπει, ακόμα και αν υπάρχουν θέσεις κενές από τα άλλα τμήματα. (Κάθε student thread, πρέπει να ελέγχει εάν έχει χώρο στο λεωφορείο σύμφωνα με το department όπου ανήκει.) Επίσης το λεωφορείο αυτό αφού φύγει από μία στάση για να φτάσει στην επόμενη, χρειάζεται ένα χρονικό διάστημα (T) δευτερολέπτων.

Γράψτε κώδικα σε C που να επιτρέπει την μεταφορά των φοιτητών στην Πανεπιστημιούπολη και πίσω στο σπίτι τους, σύμφωνα με τους παραπάνω κανόνες, χρησιμοποιώντας σημαφόρους για τον συγχρονισμό τους.

Η λύση θα πρέπει να:

- Χρησιμοποιεί ένα διαφορετικό thread για κάθε φοιτητή.
- Χρησιμοποιεί ένα thread για το λεωφορείο.
- Αποφεύγει το starvation (κανένας φοιτητής να μην περιμένει για πάντα σε κάποια στάση).
- Τηρεί την σειρά προτεραιότητας των φοιτητών (FIFO). Οι φοιτητές πρέπει να μπαίνουν στο λεωφορείο με την σειρά που έφτασαν στη στάση (αν τους επιτρέπεται η είσοδος στο λεωφορείο με βάση τα προηγούμενα κριτήρια) Τερματίζει όταν όλοι οι φοιτητές φτάσουν σπίτια τους.

Αρχικοποίηση φοιτητών

1. Η αρχικοποίηση των φοιτητών-threads περιλαμβάνει ανάθεση αριθμού μητρώου (AM)
2. Ορίστε τον χρόνο μελέτης των φοιτητών τυχαία από 5 μέχρι 15 δευτερόλεπτα. Μπορείτε να χρησιμοποιήσετε την βιβλιοθήκη random(3) για ευκολία.
3. Κάθε στιγμή, οι φοιτητές μπορούν να βρίσκονται:
 - Στη στάση της Χανιώπορτας (**Stop A**)
 - Στο λεωφορείο (**Bus**)
 - Στη σχολή (για απλοποίηση της άσκησης θεωρούμε πως ο χώρος της σχολής είναι ίδιος για όλους και δεν πάει ο κάθε φοιτητής σε διαφορετική σχολή, αν και ανήκει σε διαφορετική σχολή) (**University**)
 - Στη στάση της Πανεπιστημιούπολης (**Stop B**)

Αρχικοποίηση λεωφορείου

Το λεωφορείο χωράει μέχρι N φοιτητές, N/4 από κάθε σχολή και στην αρχή είναι άδειο.

Αν το λεωφορείο φτάσει σε μία στάση η οποία είναι άδεια, θα φύγει για την επόμενη στάση σύμφωνα με τους χρόνους που του έχετε ορίσει ακόμα και αν είναι άδειο.

Αρχικοποίηση σχολής

Η σχολή χωράει απεριόριστο αριθμό ατόμων, ανεξαρτήτως σχολής και στην αρχή είναι άδεια. Οι συνολικοί φοιτητές που θέλουν να διαβάσουν πρέπει να είναι έως 200 και θα δίνονται κατά την αρχικοποίηση ως είσοδος στο πρόγραμμα.

Σημειώσεις

- Μπορείτε να χρησιμοποιήσετε την βιβλιοθήκη `sleep(1)` για διευκόλυνση.
- Είναι σημαντικό να τυπώνονται τα κατάλληλα μηνύματα που θα δείχνουν την κατάσταση της κάθε στάσης, του λεωφορείου και της σχολής κάθε φορά που:
 - δημιουργείται ένας φοιτητής-thread
 - όταν ένας φοιτητής φτάσει σε κάποια στάση
 - όταν ένας φοιτητής μπαίνει στο λεωφορείο
 - όταν ένας φοιτητής μπει στη σχολή
 - όταν ένας φοιτητής φτάσει σπίτι του
 - όταν ένας φοιτητής δεν χωράει στο λεωφορείο και πρέπει να περιμένει στη στάση.
- Προτεινόμενες αρχικοποιήσεις μεταβλητών:
 - T: 10 sec
 - N: 12
 - Bus waiting time at each stop : 3 sec (Optional)
 - Number of students (input) > 12
- Γενικά προσπαθήστε να ελέγξετε τα σενάρια που φτάνουν στα όρια των δομών σας. Εάν για παράδειγμα το λεωφορείο θα παίρνει 12 άτομα στο σύνολο (3 από κάθε τμήμα), στο stop A θα πρέπει να περιμένουν >12 άτομα για να μπορέσουμε να ελέγξουμε την πλήρη λειτουργικότητα της άσκησης.

Παράδειγμα εκτέλεσης με παραμέτρους (N=4, Number of students = 5)

See attached screenshots

Χρήσιμοι συνδεσμοί

Παρακάτω παρατίθενται οι οδηγίες χρήσης κάποιων εντολών που ίσως χρειαστείτε:

`sem_init(3)` : https://man7.org/linux/man-pages/man3/sem_init.3.html

`sem_wait(3)`: https://man7.org/linux/man-pages/man3/sem_wait.3.html

sem_post(3): https://man7.org/linux/man-pages/man3/sem_post.3.html
sem_destroy(3): https://man7.org/linux/man-pages/man3/sem_destroy.3.html
sleep(1): <https://man7.org/linux/man-pages/man1/sleep.1.html>
rand(3): https://man7.org/linux/man-pages/man3/rand_r.3.html
pthread_barrier_init(3): https://linux.die.net/man/3/pthread_barrier_init
pthread_barrier_wait(3): https://linux.die.net/man/3/pthread_barrier_wait
pthread_join(3): https://man7.org/linux/man-pages/man3/pthread_join.3.html
pthread_create(3): https://man7.org/linux/man-pages/man3/pthread_create.3.html
pthread_mutex_lock: https://man7.org/linux/man-pages/man3/pthread_mutex_lock.3p.html

Παρατηρήσεις

1. Η άσκηση είναι **ατομική**. Τυχόν αντιγραφές μπορούν να ανιχνευθούν εύκολα από κατάλληλο πρόγραμμα και να μηδενιστούν.
2. Συμπεριλάβετε το όνομα σας, τον αριθμό μητρώου σας, και το username σας σε όλα τα αρχεία.
3. Η μεταγλώττιση πρέπει να γίνει με την παράμετρο **-pthread** ώστε ο linker να μπορεί να βρει τις συναρτήσεις στην pthread βιβλιοθήκη.
4. 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 μείνουν μόνο τα αρχεία που χρειάζονται για τη μεταγλώττιση.
5. Γράψτε ένα αρχείο **README.txt**, το πολύ 30 γραμμών, με επεξηγήσεις για τον τρόπο υλοποίησης
6. 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).
7. Το πρόγραμμά σας μπορεί να γραφεί μόνο για λειτουργικά συστήματα που υπάρχουν στα εργαστήρια (μηχανήματα σχολής) και σε γλώσσα προγραμματισμού C χρησιμοποιώντας system calls της ομάδας threads (POSIX Threads).