

HY-345 ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ
ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2021
ΑΣΚΗΣΗ 2

Ανάθεση 5/11
Παράδοση 19/11

Σκοπός της συγκεκριμένης άσκησης είναι η εξοικείωση με νήματα (Posix Threads). Οι βασικές λειτουργίες των προγραμμάτων που θα υλοποιήσετε είναι δημιουργία, συγχρονισμό και καταστροφή νημάτων.

Ερώτημα 1 - **Narrow road**

Σε αυτήν την άσκηση καλείστε να λύσετε το πρόβλημα του “στενού δρόμου” χρησιμοποιώντας **νήματα** (threads) και **σημαφόρους**.

Θεωρήστε έναν αρκετά στενό δρόμο, χτισμένο στην κορυφή ασταθών θεμελίων στο Ηράκλειο της Κρήτης. Ο δρόμος είναι τόσο στενός ώστε να χωράει μόνο ένα αυτοκίνητο στο πλάτος. Επίσης, αν δυο αυτοκίνητα εισέλθουν οδηγώντας προς αντίθετες κατευθύνσεις, εγκλωβίζονται για πάντα. Επιπρόσθετα, τα θεμέλια είναι τόσο ασταθή όπου αν περισσότερα από τρία αυτοκίνητα διασχίσουν τον δρόμο, θα καταρρεύσουν. Υλοποιήστε μια λύση, η οποία να επιτρέπει στα αυτοκίνητα να οδηγούν ακίνδυνα στο δρόμο. Συνοπτικά, η λύση σας πρέπει να ακολουθεί τα ακόλουθα σενάρια :

- Ο δρόμος καταρρέει αν περισσότερα από **τρία** αυτοκίνητα βρίσκονται στον δρόμο ταυτόχρονα
- Η κυκλοφορία σταματάει για πάντα αν δυο αυτοκίνητα εισέλθουν στον δρόμο με διαφορετική κατεύθυνση
- Starvation free. Η διέλευση αυτοκινήτων από την μια πλευρά να μην εμποδίζει (για πάντα) αυτοκίνητα από την άλλη πλευρά να διασχίζουν τον δρόμο

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

Ακόμα, η αρχικοποίηση των αυτοκινήτων-thread περιλαμβάνει τον ορισμό της κατεύθυνσης τους. Κάθε αυτοκίνητο κινείται προς μια κατεύθυνση κάθε φορά, ανατολικά ή δυτικά. Ορίστε την κατεύθυνση των αυτοκινήτων κατά την δημιουργία τους τυχαία. Μπορείτε να χρησιμοποιήσετε την βιβλιοθήκη random(3) για ευκολία.

Εξασφαλίστε την ομαλή κυκλοφορία του δρόμου τυπώνοντας κατάλληλα μηνύματα κατά την είσοδο των αυτοκινήτων αλλά και την έξοδο με τον αναγνωριστικό τους **id** καθώς και τον αριθμό των αυτοκινήτων που βρίσκονται στον δρόμο.

Το πρόγραμμα σας θα πρέπει να δέχεται από την γραμμή εντολών την παράμετρο `c`, τον αριθμό των αυτοκινήτων.

Παράδειγμα εκτέλεσης

```
./nr -c 50
```

Ερώτημα 2 - Ferry boat

Σας έχει ανατεθεί να αναπτύξετε ένα πρόγραμμα που ρυθμίζει την φόρτωση και μεταφορά αυτοκινήτων σε ένα ferry που κάνει την διαδρομή Ρίο-Αντίρριο.

Πιο συγκεκριμένα το ferry όταν φτάσει στο Ρίο επιτρέπει την είσοδο σε **έως 20 αυτοκίνητα** που περιμένουν με σκοπό να τα μεταφέρει στο Αντίρριο. Το ferry μένει για κάποιο διάστημα στο λιμάνι προκειμένου να επιβιβαστούν όσα αυτοκίνητα προλάβουν. Αν κάποια αυτοκίνητα φτάσουν ενώ κάποια βρίσκονται στην διαδικασία επιβίβασης ακόμα, τότε πρέπει να **περιμένουν** για το επόμενο ferry. Μπορείτε να χρησιμοποιήσετε την συνάρτηση `sleep`.

Δώστε μια λύση για το παραπάνω πρόβλημα συγχρονισμού, το πρόγραμμά σας θα πρέπει να δέχεται ως είσοδο από την γραμμή εντολών τον συνολικό αριθμό των αμαξιών. Όπως φαίνεται και στο παράδειγμα:

```
./ferry -c 76
```

Κάθε αυτοκίνητο θα αποτελεί ένα **thread** στο πρόγραμμά σας και στην αρχή θα περιμένει για ένα τυχαίο χρόνο ώστε να προσομοιώσει τα αυτοκίνητα που φτάνουν σε διαφορετικές στιγμές στο λιμάνι του Ρίο. Μπορείτε να χρησιμοποιήσετε τις συναρτήσεις **`sleep(3)`** και **`rand(3)`**.

Για την καλύτερη διευκόλυνση όλων το πρόγραμμα σας θα πρέπει να τυπώνει κατάλληλα μηνύματα κατά την άφιξη αλλά και την επιβίβαση των αυτοκινήτων με το αναγνωριστικό τους **`id`** καθώς και τον **αριθμό** των αυτοκινήτων που βρίσκονται στο ferry.

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. Η μεταγλώττιση πρέπει να γίνει με την παράμετρο **-pthread** ώστε ο linker να μπορεί να βρει τις συναρτήσεις στην pthread βιβλιοθήκη.
3. 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.
4. Γράψτε ένα αρχείο README.txt, το πολύ 30 γραμμών, με επεξηγήσεις για τον τρόπο υλοποίησης.
5. 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)
6. Το πρόγραμμά σας μπορεί να γραφεί μόνο για λειτουργικά συστήματα που υπάρχουν στα εργαστήρια και σε γλώσσα προγραμματισμού C χρησιμοποιώντας system calls της ομάδας threads (POSIX Threads).
7. Για καλύτερη κατανόηση - ανάγνωση του προγράμματος μπορείτε να χρησιμοποιήσετε την συνάρτηση sleep(3) π.χ. προσπέλαση δρόμου, επιβίβαση, μεταφορά αυτοκινήτων.
8. Ο έλεγχος και ο συγχρονισμός ταυτόχρονα εκτελούμενων διεργασιών μπορεί να επιτευχθεί με την χρήση σημαφόρων (semaphore).
9. Για να γίνει το πρόβλημα πιο ρεαλιστικό, χρησιμοποιήστε sleep για την παραγωγή. πχ Το ferry περιμένει ένα χρονικό διάστημα X second για να προλάβουν να επιβιβαστούν κάποια αυτοκίνητα.