

Σειρά Εργασιών 2

2.1 Δυαδικοί σηματοφόροι

Υλοποιήστε μια δική σας «βιβλιοθήκη» που παρέχει δυαδικούς σηματοφόρους με τις λειτουργίες:

<code>int mysem_init(mysem_t *s, int n);</code>	Αρχικοποίηση σηματοφόρου με τιμή n. Επιστρέφει 1 για επιτυχία, 0 αν το $n < 0$ ή -1 αν ο σηματοφόρος είναι ήδη αρχικοποιημένος.
<code>int mysem_down(mysem_t *s);</code>	Μείωση σηματοφόρου κατά 1. Επιστρέφει 1 για επιτυχία ή -1 αν ο σηματοφόρος δεν έχει αρχικοποιηθεί.
<code>int mysem_up(mysem_t *s);</code>	Αύξηση σηματοφόρου κατά 1. Επιστρέφει 1 για επιτυχία, 0 αν ο σηματοφόρος είναι ήδη 1 ή -1 αν ο σηματοφόρος δεν έχει αρχικοποιηθεί.
<code>int mysem_destroy(mysem_t *s);</code>	Καταστρέφει τον σηματοφόρο. Επιστρέφει 1 για επιτυχία ή -1 αν ο σηματοφόρος δεν έχει αρχικοποιηθεί.

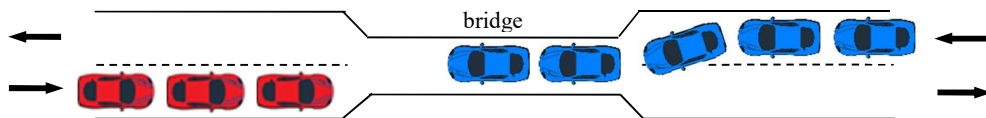
Η υλοποίησή σας πρέπει (εσωτερικά) να χρησιμοποιεί τους σηματοφόρους του system V (υποθέστε ότι είναι δίκαιοι).

2.2 Αναγνώριση πρώτων αριθμών

Αλλάξτε το πρόγραμμα που αναπτύξατε στην εργασία 1.2 έτσι ώστε ο επιθυμητός συγχρονισμός ανάμεσα στο κυρίως νήμα και τα νήματα εργάτες να γίνεται αποκλειστικά με τους δικούς σας δυαδικούς σηματοφόρους.

2.3 Στενή γέφυρα

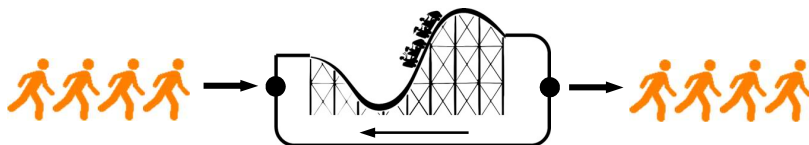
Αναπτύξτε κώδικα ελέγχου για τη ρύθμιση της κυκλοφορίας οχημάτων πάνω από μια στενή γέφυρα, έτσι ώστε να διασφαλίζονται οι εξής ιδιότητες: (1) Δεν επιτρέπεται να υπάρχουν στη γέφυρα οχήματα που κινούνται προς αντίθετες κατευθύνσεις. (2) Δεν επιτρέπεται να υπάρχουν στη γέφυρα περισσότερα από N οχήματα. (3) Δεν επιτρέπεται ένα όχημα να περιμένει για πάντα να περάσει τη γέφυρα, ακόμα και αν καταφθάνουν συνεχώς οχήματα στην άλλη πλευρά.



Υλοποιήστε τον επιθυμητό συγχρονισμό ανάμεσα στα οχήματα, μέσω κατάλληλου κώδικα «εισόδου»/«εξόδου» που εκτελεί κάθε όχημα όταν φτάνει και αφού περάσει τη γέφυρα, αντίστοιχα. Ο συγχρονισμός πρέπει να γίνεται απ' ευθείας ανάμεσα στα οχήματα, χωρίς κάποιο νήμα που να παίζει τον ρόλο του μεσάζοντα/τροχονόμου. Η υλοποίησή σας πρέπει να βασίζεται στους δικούς σας δυαδικούς σηματοφόρους. Για να δοκιμάσετε/επιδείξετε τη λύση σας, αναπτύξτε ένα απλό πρόγραμμα προσομοίωσης που δημιουργεί νήματα/οχήματα σε κάθε πλευρά της γέφυρας με βάση πληροφορία που διαβάζει από την είσοδό του.

2.4 Τρενάκι

Το τρενάκι ενός λούνα-παρκ χωράει N επιβάτες. Το τρενάκι αρχίζει την επόμενη διαδρομή του μόνο όταν γεμίσει, ενώ οι επιβάτες αποβιβάζονται από το τρενάκι αφού αυτό ολοκληρώσει την τρέχουσα διαδρομή, και προτού αρχίσει η επιβίβαση των επόμενων επιβατών.



Υλοποιήστε τον επιθυμητό συγχρονισμό ανάμεσα στους επιβάτες και το τρενάκι, χωρίς κάποιο άλλο βοηθητικό νήμα. Η υλοποίησή σας πρέπει να βασίζεται στους δικούς σας δυαδικούς σηματοφόρους. Για να δοκιμάσετε/επιδείξετε τη λύση σας, αναπτύξτε ένα απλό πρόγραμμα προσομοίωσης που αρχικά δημιουργεί ένα νήμα για το τρενάκι και στην συνέχεια επιπλέον νήματα-επιβάτες με βάση πληροφορία που διαβάζει από την είσοδο του. Ο αριθμός επιβατών που μπορεί να χωρέσει το τρενάκι δίνεται ως όρισμα του προγράμματος.

Η υλοποίηση πρέπει να γίνει σε C με χρήση της βιβλιοθήκης pthreads. Ο συγχρονισμός πρέπει να υλοποιηθεί αποκλειστικά με τους δικούς σας δυαδικούς σηματοφόρους και χωρίς να υπάρχει ενεργή αναμονή.

Παράδοση: Σάββατο 27 Νοεμβρίου 2021, 23:59