ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ Ι

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2011-2012

2η Άσκηση

6 Δεκεμβρίου 2011

Στόχος της άσκησης είναι η δημιουργία ενός απλού ηλεκτρονικού συστήματος κράτησης θέσεων σε πάρκινγκ. Υποθέτουμε ότι υπάρχουν δύο κατηγορίες χρηστών, ιδιοκτήτες πάρκινγκ και οδηγοί. Οι ιδιοκτήτες μπορούν να προσθέτουν νέα πάρκινγκ στο σύστημα, η περιγραφή των οποίων θα αποτελείται από τα εξής στοιχεία:

- Όνομα πάρκινγκ (15 χαρακτήρες μέγιστο)
- Περιγραφή πάρκινγκ (80 χαρακτήρες μέγιστο)
- Τιμή για χρήση θέσης πάρκινγκ
- Για ευκολία υποθέτουμε ότι κάθε πάρκινγκ διαθέτει 10 θέσεις
- Για ευκολία επίσης υποθέτουμε ότι μπορούμε να κάνουμε κράτηση μόνο για τα επόμενα 30 δευτερόλεπτα

Οι ταξιδιώτες διατηρούν επίσης λογαριασμό στο σύστημα, στον οποίο θεωρούμε ότι υπάρχει και ένα χρηματικό ποσό για να πληρώνονται τα πάρκινγκ. Κάθε τέτοιος λογαριασμός έχει τα ακόλουθα στοιχεία:

- Όνοματεπώνυμο οδηγού (40 χαρακτήρες μέγιστο)
- Υπόλοιπο λογαριασμού
- Τελευταίες 4 κρατήσεις

Για ευκολία υποθέτουμε ότι μπορούμε να έχουμε το πολύ 1000 οδηγούς που να χρησιμοποιούν το σύστημα. Οι οδηγοί μπορούν να κάνουν κράτηση σε μία θέση πάρκινγκ, βάσει του υπολοίπου που έχουν στο λογαριασμό τους. Εσείς θα δημιουργήσετε ένα πρόγραμμα-πελάτη (client) που θα χρησιμοποιείται από τους ιδιοκτήτες και τους οδηγούς και ένα πρόγραμμα-εξυπηρέτη (server) που θα αποθηκεύει τις πληροφορίες για τα πάρκινγκ και τις κρατήσεις.
Έτσι, στο πρόγραμμα-πελάτης:

- Θα επιλέγετε αν είστε ιδιοκτήτης ή οδηγός
- Αν είστε ιδιοκτήτες μπορείτε να κάνετε
    - Δήλωση νέου πάρκινγκ, ή
    - Προβολή τελευταίων 20 κρατήσεων στο παρκινγκ. Οι κρατήσεις θα αναφέρουν όνομα οδηγού και χρονική στιγμή.
- Αν είστε πελάτης μπορείτε να κάνετε
    - Δημιουργία νέου λογαριασμού
    - Δημιουργία νέας κράτησης – η κράτηση θα γίνεται επιλέγοντας από τις διαθέσιμα πάρκινγκ και τις αντίστοιχες θέσεις πάρκινγκ
    - Προβολή στοιχείων λογαριασμού

- Κατάθεση χρημάτων στο λογαριασμό

Η χρέωση και απελευθέρωση θέσεων πάρκινγκ που έχουν κρατηθεί θα γίνεται αποκλειστικά από τον εξυπηρέτη.

Θα πρέπει να δώσετε σημασία σε σημεία όπως:

- Να μην γίνεται κράτηση αν δεν υπάρχουν διαθέσιμα χρήματα στο λογαριασμό του αντίστοιχου οδηγού, κτλ. Επίσης, θα πρέπει να κάνετε τους αντίστοιχους ελέγχους για να αποτρέπετε την αποστολή μη αποδεκτών στοιχείων στον εξυπηρέτη (π.χ., χαρακτήρες αντί για αριθμούς, κ.ο.κ.).
- Να μην μπορεί να δημιουργηθεί νέο πάρκινγκ με όνομα ίδιο με ήδη υπάρχον στο σύστημα.
- Επιπλέον, για τη γρήγορη μεταγλώττιση του κώδικα που θα στείλετε, απαιτείται η ύπαρξη Makefile ανάμεσα στα αρχεία που θα ανεβάσετε στη σελίδα του μαθήματος.
- Τέλος, θα πρέπει να υπάρχει δυνατότητα εισαγωγής ορισμάτων από τη γραμμή εντολών για την εισαγωγή στοιχείων στο σύστημα, και για τα δύο μέρη του συστήματος (πελάτης, εξυπηρέτης) καθώς και την εκτέλεση των ενεργειών που ζητούνται. Θα εκτιμηθεί η χρήση αρχείων για την αρχικοποίηση του συστήματος.

Σχολιασμός:

Ο στόχος της άσκησης είναι η εξοικείωση στην χρήση νημάτων του προτύπου POSIX (POSIX Threads ή PThreads). Οι λειτουργίες που θα χρειαστείτε περιλαμβάνουν τις:

1) pthread_create() και pthread_exit() για την δημιουργία και διαχείριση των νημάτων (πιθανόν και την pthread_detach(), ανάλογα με το πως θα χρησιμοποιήσετε την pthread_create())

2) pthread_mutex_init(), pthread_mutex_lock(), pthread_mutex_unlock() και pthread_mutex_destroy() για τον συγχρονισμό νημάτων.

Ο server θα είναι μια διεργασία που θα εκτελείται στο παρασκήνιο (background). Ο server θα δεσμεύει ένα τμήμα μνήμης, στο οποίο θα αποθηκεύονται όλες οι πληροφορίες για κάθε λογαριασμό. Για κάθε αίτηση που θα λαμβάνει ο server θα δημιουργεί ένα νέο νήμα, το οποίο θα εξυπηρετεί την αίτηση. Ο server αμέσως θα μπαίνει πάλι σε κατάσταση αναμονής, περιμένοντας την επόμενη αίτηση. Στην υλοποίηση αυτή δεν χρειάζεται να δημιουργήσετε ρητά κοινή μνήμη, αφού τα νήματα εξ ορισμού μοιράζονται τον χώρο διευθύνσεων της διεργασίας. Τα νήματα που εξυπηρετούν τις αιτήσεις θα πρέπει να τερματίζονται κατάλληλα, ώστε στο σύστημα να μην παραμένουν άχρηστα νήματα.

Ο client θα είναι μια διεργασία που θα εκτελείται στο προσκήνιο (foreground) και θα επικοινωνεί με τον server μέσω sockets. Μόλις ο χρήστης συμπληρώνει τις απαραίτητες πληροφορίες, αυτές θα αποστέλλονται στον server και θα ενημερώνονται οι αντίστοιχες πληροφορίες.

Προσοχή απαιτείται λόγω του γεγονότος ότι θα είναι δυνατόν να εκτελούνται ταυτόχρονα πολλοί clients. Επομένως, η ενημέρωση των πληροφοριών από τα νήματα που θα δημιουργεί ο server θα πρέπει να προστατεύεται από κάποια μέθοδο συγχρονισμού. Επειδή οι μεταβλητές που θα χρησιμοποιηθούν για τον συγχρονισμό θα πρέπει να είναι προσπελάσιμες από όλα τα νήματα που δημιουργεί ο server, οι μεταβλητές αυτές θα πρέπει να κοινές.

Έλεγχος ορθότητας:

Για να βεβαιωθείτε ότι οι λειτουργίες του server έχουν υλοποιηθεί σωστά (και ιδιαίτερα ο συγχρονισμός), μπορείτε να δημιουργήσετε clients οι οποίοι δεν θα περιμένουν είσοδο από τον χρήστη, αλλά θα στέλνουν προκαθορισμένα στοιχεία σε έναν λογαριασμό στον server. Για παράδειγμα, ο client μπορεί να κάνει 1000 καταθέσεις και 1000 αναλήψεις εναλλάξ του ίδιου ποσού σε έναν μόνο λογαριασμό. Με ένα script μπορείτε να ξεκινήσετε 10 clients μαζί. Αν εκτελέσετε το πείραμα πολλές φορές και στο τέλος κάθε εκτέλεσης το ποσό που μένει στον λογαριασμό είναι το ίδιο με το αρχικό ποσό, τότε πιθανότατα το πρόγραμμά σας είναι σωστό. Ο κώδικας του client που θα παραδώσετε όμως θα πρέπει να δέχεται τα δεδομένα από τον εκάστοτε χρήστη.

Διαδικαστικά:

- Η άσκηση θα υλοποιηθεί είτε ατομικά, είτε σε ομάδες των 2 ή 3 ατόμων. Οι ομάδες θα πρέπει να είναι οι ίδιες με αυτές της προηγούμενης άσκησης.
- Η ημερομηνία παράδοσης της άσκησης είναι η Παρασκευή, 30 Δεκεμβρίου 2011 και ώρα 23:59.
- Ο κώδικας που θα παραδώσετε θα πρέπει να είναι καλά δομημένος.
- Ο κώδικας που θα παραδώσετε θα πρέπει να είναι καλά σχολιασμένος.
- Ο κώδικας θα παραδοθεί ηλεκτρονικά μέσω της σελίδας του μαθήματος.
- Δεν χρειάζεται να παραδώσετε αναφορά, αλλά μόνο τα προγράμματα που υλοποιούν τον server και τον client.
- Ο κώδικας θα πρέπει να χρησιμοποιεί τα εργαλεία που αναφέρθηκαν νωρίτερα (pthread_create, pthread_exit, pthread_detach, pthread_mutex_init, pthread_mutex_lock, pthread_mutex_unlock, pthread_mutex_destroy).
- Σε περίπτωση καθυστέρησης στην παράδοση της άσκησης θα υπάρχει μείωση 10% για κάθε 24 ώρες καθυστέρησης.
- Σε περίπτωση που εντοπιστεί αντιγραφή, ο βαθμός στο μάθημα θα μηδενίζεται για όλους όσους εμπλέκονται (και αυτούς που έλαβαν την άσκηση και αυτούς που την έδωσαν).