

Λειτουργικά Συστήματα Ι
Χειμερινό εξάμηνο, ακαδημαϊκή περίοδος 2013-2014

Αναφορά άσκησης 1^{ης}:
Διεργασίες, σημαφόροι, σήματα, κοινή μνήμη και sockets.

ΕΠΩΝΥΜΟ:ΠΑΤΡΩΝΗ
ΟΝΟΜΑ:ΣΩΤΗΡΙΑ
ΑΜ:5399

Στο πρόγραμμα που ζητείται στην παρούσα άσκηση έχουν χρησιμοποιηθεί όλα τα απαιτούμενα εργαλεία που προσφέρει το λειτουργικό σύστημα, όπως η δημιουργία και η διαχείριση διεργασιών (fork, exit), η επικοινωνία διεργασιών (sockets, shared memory, signals) και ο συγχρονισμός (semaphores).

Αναλυτικά έχουν χρησιμοποιηθεί 3 semaphores στο πρόγραμμα του server, ο ένας είναι για την προσπέλαση στην κοινή μνήμη που πρέπει να γίνεται υποχρεωτικά κάθε φορά μόνο από μία διεργασία για να γίνεται τροποποίηση στην κοινή μνήμη, ο δεύτερος είναι για την διεργασία που υλοποιεί το σύστημα της τράπεζας γιατί μπορεί να εξυπηρετεί ταυτόχρονα το πολύ 4 αιτήσεις ελέγχου, και ο τρίτος χρησιμοποιείται για τη διεργασία που υλοποιεί τους 10 τηλεφωνητές, αφού μπορούν να εξυπηρετήσουν μόνο 10 αιτήσεις από clients.

Επίσης χρησιμοποιήθηκε μία shared memory για τα δεδομένα που χρειάζονται για τις κρατήσεις του θεάτρου (διαθέσιμες θέσεις σε κάθε ζώνη, χρηματικό ποσό που συγκεντρώθηκε, συνολικός αριθμός πελατών κλπ.). Χρησιμοποιήθηκε άλλη μία shared memory για την μετάβαση της εγκυρότητας της πιστοποιητικής κάρτας ή όχι από την διεργασία τράπεζας στην διεργασία τηλεφωνητή (δημιουργείται για κάθε αίτηση client μία shared memory που περιέχει την πληροφορία εγκυρότητας).

Χρησιμοποιούνται signals για τον τερματισμό του server (ctrl+C), για την αποφυγή των zombie processes και alarm signals στον server για την μεταφορά των χρημάτων από τον λογαριασμό της εταιρίας κρατήσεων (company_account) στον τραπεζικό λογαριασμό του θεάτρου (theater_account) και την συμπλήρωση του πίνακα μεταφορών, και στον client για να τυπώνεται ένα μήνυμα «συγγνώμης» για την καθυστέρηση λόγω αναμονής.

Τα sockets που χρησιμοποιούνται είναι για την επικοινωνία μεταξύ server και client για την αποστολή του αιτήματος για τα εισητήρια

που επιθυμεί ο client για την κράτηση και για την ενημέρωση του client από τον server σχετικά με την πρόοδο της καράτησης.

Οι συναρτήσεις fork και exit χρησιμοποιούνται για την παράλληλη εκτέλεση κάποιων πραγμάτων που απαιτεί το πρόγραμμα και για την έξοδο από το πρόγραμμα αντίστοιχα.

Η επιλογή των εισητηρίων γίνεται με τυχαίο τρόπο με την κλήση μίας συνάρτησης, όταν ο client δίνει αυτόματα δεδομένα στον server (στο αρχείο client.c), ενώ έχει δημιουργηθεί και ένα δεύτερο αρχείο client (clientinput.c) όπου τα δεδομένα δίνονται στο πρόγραμμα από τον χρήστη.

Όσον αφορά τις παραγγελίες, στο πρόγραμμα έχουν θεωρηθεί αποτυχημένες παραγγελίες οι παραγγελίες οι οποίες είτε ο client που έχει στείλει αίτημα για εξυπηρέτηση δεν έχει έγκυρη πιστωτική κάρτα, δεν υπάρχουν διαθέσιμες θέσεις στην ζώνη που ζητήθηκε από τον πελάτη ή δεν υπάρχουν καθόλου διαθέσιμες θέσεις στο θέατρο.

Οι δομές που χρησιμοποιήθηκαν στο πρόγραμμα είναι οι εξής:

- struct zones, η οποία χρησιμοποιείται για την αποθήκευση των δεδομένων στη shared memory και τα πεδία της είναι ένας πίνακας 4 θέσεων που περιέχει το σύνολο των διαθέσιμων θέσεων που περιέχει το θέατρο (κάθε ζώνη αντιστοιχεί σε μία θέση του πίνακα, Α ζώνη->κελί [0] κλπ.), έναν πίνακα με τις τιμές που έχει κάθε ζώνη με αντίστοιχο τρόπο, μία μεταβλητή που περιέχει το σύνολο των πελατών, μία που περιέχει τον λογαριασμό της εταιρίας κρατήσεων (company_account), μία που περιέχει τις αποτυχημένες παραγγελίες, μία που περιέχει τις συνολικές παραγγελίες, δύο μεταβλητές για τον χρόνο αναμονής και εξυπηρέτησης των παραγγελιών αντίστοιχα, 4 πίνακες που αντιστοιχούν στις ζώνες του θεάτρου και περιέχουν ουσιαστικά το «πλάνο των θέσεων» του θεάτρου με το id του κάθε πελάτη στην θέση την οποία έχει κάνει κράτηση, και τέλος έναν πίνακα 4 θέσεων (κάθε ζώνη αντιστοιχεί σε μία θέση του πίνακα) όπου περιέχει τον συνολικό αριθμό των τρέχουσων κλεισμένων θέσεων του θεάτρου.
- struct krathsh, η οποία χρησιμοποιείται για την αποθήκευση των στοιχείων κάθε πελάτη (δημιουργείται μία δομή για κάθε πελάτη) και τα πεδία της είναι ένας αναγνωριστικός αριθμός για τον εκάστοτε πελάτη, μία μεταβλητή με το κόστος της παραγγελίας, μία μεταβλητή η οποία χρησιμοποιείται σαν έλεγχος αν η παραγγελία θεωρείται αποτυχημένη ή όχι με βάση την παραπάνω θεώρηση (η μεταβλητή παίρνει τις τιμές: μηδέν αν η παραγγελία είναι επιτυχημένη, 2 αν δεν υπάρχουν θέσεις στην συγκεκριμένη ζώνη, και 3 αν το θέατρο έχει γεμίσει).

- struct aithsh (που περιέχεται και στον client), η οποία χρησιμοποιείται για να περαστούν τα δεδομένα που στέλνει ο client στον server με την χρήση sockets, και το μοναδικό της πεδίο είναι ένας πίνακας 4 θέσεων με τις ζώνες (κάθε ζώνη αντιστοιχεί σε μία θέση του πίνακα) που περιέχεται ο αριθμός των εισητηρίων που επιθυμεί ο πελάτης σε κάθε κελί (μόνο ένα κελί θα περιέχει αριθμό διάφορο του μηδενός κάθε φορά γιατί ο πελάτης έχει την δυνατότητα να επιλέξει εισητήρια μόνο από μία ζώνη κάθε φορά).

Οι συναρτήσεις που έχουν χρησιμοποιηθεί για την εκτέλεση του προγράμματος του server είναι οι εξής:

- η συνάρτηση cr_crd_rand για την τυχαία εγκυρότητα πιστωτικής κάρτας
- η συνάρτηση closing_server για το κλείσιμο του server (τυπώνεται το πλάνο των θέσεων και τα ζητούμετα άλλα στοιχεία για το θέατρο)
- ο signal handler sig_chld για την αποφυγή των zombie processes
- ο signal handler catch_int που τερματίζει τον server καλώντας την συνάρτηση closing_server
- ο signal handler catch_alarm για την μεταφορά των χρημάτων από από τον λογαριασμό της εταιρίας κρατήσεων (company_account) στον τραπεζικό λογαριασμό του θεάτρου (theater_account) και την συμπλήρωση του πίνακα μεταφορών

Οι συναρτήσεις που έχουν χρησιμοποιηθεί για την εκτέλεση του προγράμματος του client είναι οι εξής:

- η συνάρτηση rand_select η οποία επιλέγει τυχαία σε ποιά ζώνη του θεάτρου θα γίνει η κράτηση
- ο signal handler catch_alarm για να τυπώνεται ένα μήνυμα «συγγνώμης» για την καθυστέρηση λόγω αναμονής

Σημείωση: αρχικά έχει αρχικοποιηθεί με μηδέν στο πρόγραμμα (του server) μία μεταβλητή που συγκεντώνει το σύνολο των παραγγελιών που έχουν γίνει (αποτυχιμένες ή μη) και για την εύρεση του ποσοστού αποτυχιμένων παραγγελιών $(\text{αποτυχιμένες} / \text{συνολικές παραγγελίες} * 100 \%)$ γίνεται διαίρεση με αυτή τη μεταβλητή. Αυτό σημαίνει ότι αν τερματιστεί το πρόγραμμα πριν ο server προλάβει να δεχθεί αίτημα προς εξυπηρέτηση η μεταβλητή αυτή θα παραμείνει μηδέν και θα δημιουργηθεί error κατά την εκτέλεση του προγράμματος. Δεν έχει δημιουργηθεί exception για αυτή την περίπτωση διότι θεωρούμε ότι ο server δεν θα τερματιστεί χωρίς να δεχθεί αιτήματα!

Η άσκηση έχει υλοποιηθεί ατομικά και ο φάκελος που έχει επισυναπτεί περιέχει τα παρακάτω αρχεία:

Για πρακτικούς (μεγάλος χώρος στην αναφορά) λόγους δεν γίνεται εμφάνιση του τελικού πλάνου του θεάτρου όταν έχει γεμίσει πλήρως.