

## ΕΡΓΑΣΙΑ ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ 2021-2022

- Πέτρος Χάνας [3170173]
- Αλέξανδρος-Ιωάννης Δουνάκης [3170044]
- Ελευθέριος Μιχαηλίδης [3170110]

Σε αυτήν την εργασία υλοποιήσαμε ένα σύστημα κράτησης θέσεων θεάτρου με την χρήση της βιβλιοθήκης POSIX. Η εργασία υλοποιεί όλα τα ζητούμενα για τις δοσμένες παραμέτρους των 100 πελατών με αρχικό σπόρο 1000 και εμφανίζει ακολούθως τις κάτωθι τιμές:

Για κάθε Χρήστη

1. Μήνυμα επιτυχούς κράτησης με την εμφάνιση της αντίστοιχης ζώνης, σειράς και αριθμού θέσης όπως και το συνολικό κόστος της συναλλαγής
2. Εμφάνιση αποτυχίας κράτησης λόγω μη ύπαρξης κατάλληλων θέσεων
3. Αποτυχία Συναλλαγής λόγω κωλύματος στην πιστωτική κάρτα

Στο τέλος του προγράμματος εμφανίζουμε:

1. Το συνολικό πλάνο θέσεων
2. Τα συνολικά έσοδα από τις πωλήσεις
3. Το ποσοστό των συναλλαγών που ολοκληρώνεται με καθέναν από τους άνωθι 3 τρόπους
4. Μέσο χρόνο αναμονής πελατών
5. Μέσο χρόνο εξυπηρέτησης πελατών

**Τρόπος Εκτέλεσης Προγράμματος:**

Το πρόγραμμα αποτελείται από τα εξής αρχεία:

- p3170173-p3170044-p3170110-res.h
- p3170173-p3170044-p3170110-res.c
- test-res.sh

Χρησιμοποιώντας τον compiler gcc έχουμε την ακόλουθη σειριακή ακολουθία εντολών

```
gcc -o theater -pthread p3170173-p3170044-p3170110-res.c p3170173-  
p3170044-p3170110-res.h  
./theater 100 1000
```

Εναλλακτικά, μπορείτε απλά να τρέξετε το αρχείο test-res.sh σε περιβάλλον bash με την εντολή ./test-res.sh και το πρόγραμμα θα μεταγλωττιστεί και θα εκτελεστεί.

Αρχικά, στο header έχουμε δηλώσει όλες τις απαραίτητες σταθερές που δίνονταν στην εκφώνηση της εργασίας. Έπειτα στο .c αρχείο εκτελούμε το πρόγραμμά μας. Στην αρχή αρχικοποιούμε τα mutexes και δηλώνουμε κάποια conditions για τα ίδια mutexes συγκεκριμένα για την διαθεσιμότητα τηλεφώνητών και ταμιών. Μετά, δηλώνουμε κάποιες καθολικές μεταβλητές όπως και την δήλωση των συναρτήσεων που υλοποιούμε μετά την main().

Μετά το προγράμμα μας εισέρχεται στην main όπου και κάνουμε έλεγχο εγκυρότητας των παραμέτρων εισαγωγής και την ορθότητα αυτών και μετά αρχικοποιούμε το πλάνο του θεάτρου για κάθε ζώνη και κάθε θέση ως κενή. Σε 2 πίνακες αποθηκεύουμε το αναγνωριστικό του πελάτη και το thread για τον κάθε πελάτη που έχει εισαχθεί στο σύστημα. Έπειτα, καλούμε την διαδικασία του thread και μαζί με αυτή την συνάρτησή run που προσομοιώνει το τηλεφώνημα μεταξύ πελάτη και τηλεφωνητή αλλά και εν γένει την διαδικασία της κράτησης μίας θέσης.

Εντός της run, επιλέγουμε τυχαίο αριθμό θέσεων και χρόνου για κάθε πελάτη στα δοθέντα διαστήματα. Μετά, υπολογίζουμε την πιθανότητα και το συμπλήρωμά της για την εκάστοτε ζώνη που θα πάει ο πελάτης και ψάχνουμε τις θέσεις σειριακά μέχρις ότου να εμφανιστούν οι απαιτούμενες θέσεις στην σειρά. Αφ'ότου ολοκληρωθεί αυτή η διαδικασία υπολογίζουμε το κόστος κράτησης όπως και την πιθανότητα επιτυχίας πίστωσης της κάρτας του πελάτη και μετά υπολογίζουμε τον χρόνο αναμονής και τον χρόνο εξυπηρέτησης για τον κάθε πελάτη. Για να τα πραγματοποιήσουμε όλα αυτά χρησιμοποιούμε και κάποιες βοηθητικές συναρτήσεις που καλούνται εντός της συνάρτησης run. Αφού ο πελάτης ολοκληρώσει την κράτησή του επιστρέφει στην main() όπου γίνεται κατάλληλος χειρισμός για την αναμονή των υπόλοιπων πελατών.

Ολοκληρώνοντας, το πρόγραμμα εμφανίζει τα ζητούμενα αποτελέσματα και απελευθερώνει την όποια μνήμη είχε δεσμεύσει κατά την διάρκεια της εκτέλεσης του.

Να προσθέσουμε επίσης ότι στο πρόγραμμα γίνεται ο σωστός χειρισμός λαθών σχετικά με το κλείδωμα/ξεκλείδωμα των mutexes όπου αυτά εμφανίζονται.

Αναλυτικά, η λειτουργία της κάθε συνάρτησης και του κώδικα περαιτέρω έχει αναλυθεί σε σχόλια εντός του κώδικα.