**ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ**
**Εαρινό Εξάμηνο 2022**


- Ελευθερία Ντούλια Α.Μ.: 3180129
- Μαρία Μαγλογιάννη Α.Μ.: 3150094
- Ηλίας Θεοφάνης Γραββάνης Α.Μ.: 3200248


Περιγραφή Εργασίας:

Σκοπός της εργασίας είναι η υλοποίηση ενός συστήματος αγοράς θέσεων σε θεατρικές παραστάσεις, με χρήση του πακέτου νημάτων POSIX threads (pthreads). Η δομή της εργασίας έχει ως εξής:

- **main()** : Αρχικά, ελέγχει αν ο χρήστης έχει δώσει σωστό input, αν είναι δηλαδή θετικοί αριθμοί, διαφορετικά οδηγείται στον τερματισμό του προγράμματος. Στη συνέχεια, αρχικοποιείται το πλάνο θέσεων ως δύο δισδιάστατοι πίνακες, ένας για κάθε ζώνη θέσεων (newSeatArrayA[i][j], newSeatArrayB[i][j]), και γεμίζουμε τα κελιά με -1 (δηλαδή η θέση δεν έχει αποδοθεί σε κανέναν πελάτη). Μετά γίνεται η αρχικοποίηση και η καταστροφή mutex και condition. Τελικά τυπώνει το πλάνο των θέσεων ανά Ζώνη, το συνολικό balance,το average waiting time, το average service time, τον αριθμό των συναλλαγών που ακυρώθηκαν λόγω έλλειψης θέσεων, τον αριθμό των συναλλαγών που ακυρώθηκαν λόγω αποτυχίας πληρωμής και τέλος τον αριθμό των συνολικών συναλλαγών.

- **calcPzoneA()** : Υπολογίζει την πιθανότητα, βάση ενός τυχαίου αριθμού rand(), που ο πελάτης θα επιλέξει θέσεις σε κάθε ζώνη. Για την ζώνη Β επιστρέφει 0 και για τη ζώνη Α επιστρέφει 1.

- **calcPcardSuccess()** : Υπολογίζει την πιθανότητα, βάση ενός τυχαίου αριθμού rand(), που καθορίζει την επιτυχία της πληρωμής με πιστωτική κάρτα. Για την επιτυχία πληρωμής επιστρέφει 1 και για την αποτυχία πληρωμής επιστρέφει -1.

- **rndGen()** : Επιστρέφει ένα τυχαίο int με ορίσματα το min και max και με βάση το seed που έχει δώσει ο χρήστης από την έναρξη της εκτέλεσης του προγράμματος.

- **reserveSeatsZA()** : Πραγματοποιείται ο έλεγχος διαθεσιμότητας των συνεχόμενων θέσεων που έχει δηλώσει ο πελάτης, στη Ζώνη Α.Πρώτα ελέγχουμε αν υπάρχουν θέσεις στην συγκεκριμένη ζώνη, μέσω του ελέγχου της τιμής της θέσης του πίνακα, αφού αν η τιμή είναι -1 σημαίνει ότι η θέση είναι κενή και μπορεί να αποδοθεί στον τρέχοντα πελάτη, ενώ σε διαφορετική περίπτωση προχωράμε στον έλεγχο της επόμενης. Αν έχουμε sold out, δηλαδή αν όλες οι θέσεις στην συγκεκριμένη ζώνη έχουν κρατηθεί, πραγματοποιείται έξοδος από το πρόγραμμα. Στη συνέχεια ελέγχουμε αν οι διαθέσιμες

θέσεις, με τον επιθυμητό αριθμό που έχει δοθεί από τον πελάτη, είναι συνεχόμενες. Αν δεν είναι συνεχόμενες, πραγματοποιείται έξοδος από το πρόγραμμα, διαφορετικά, καταχωρείται στις θέσεις του πίνακα το id του πελάτη και προχωράμε για την κράτηση.

- **reserveSeatsZB()** : Όμοια με το **reserveSeatsZA(),** πραγματοποιείται ο έλεγχος διαθεσιμότητας των συνεχόμενων θέσεων στη Ζώνη Β

- **customerServe()** : Αποτελεί τον κορμό των διεργασιών που εκτελεί το κάθε thread. Ελέγχει αρχικά, μέσω του mutex TelCounter για το εκάστοτε thread-πελάτη αν υπάρχει διαθέσιμος τηλεφωνητής. Χρησιμοποιούμε έναν βρόχο με while για να ελέγχουμε τη συνθήκη αναμονής και περιμένουμε με pthread_cond_wait μέχρι να γίνει κάποιος διαθέσιμος (με χρήση του **pthread_cond_signal(&telThresholdCond),** ειδοποιούμε τα νήματα που περιμένουν να ξεκινήσουν). Όταν βρεθεί ο πρώτος διαθέσιμος, κλειδώνουμε με mutex για να τον χρησιμοποιήσουμε σε αυτό το νήμα. Σε αυτό το σημείο κρατάμε τον χρόνο αναμονής που χρειάστηκε μέχρι να συνδεθεί ο πελάτης με τον τηλεφωνητή για να χρησιμοποιηθεί και να προστεθεί στον ολικό χρόνο αναμονής του πελάτη (totalWaitingTime). Έπειτα, γίνεται χρήση της **calcPzoneA()** για να δούμε, με βάση τις δοθείσες από την εκφώνηση πιθανότητες, σε ποια ζώνη επιθυμεί ο πελάτης να κάνει κράτηση.Έπειτα, επιλέγουμε τυχαία έναν αριθμό μεταξύ T_SEAT_LOW, T_SEAT_HIGH και η διεργασία “κοιμάται” - περιμένει για ανάλογα δευτερόλεπτα.. Στη συνέχεια κλειδώνουμε με το seatsMutex τις θέσεις του θεάτρου, έτσι ώστε να επεξεργάζονται αποκλειστικά από αυτό μόνο το Thread, και ανάλογα με το αποτέλεσμα της **calcPzoneA** καλείται είτε η **reserveSeatsZA()** είτε η **reserveSeatsZB(),** ώστε να επιλέξουμε σε ποια ζώνη θα εξετάσουμε την ύπαρξη διαθέσιμων θέσεων. Αμέσως μετά την επιλογή της επιθυμητής ζώνης, καλούμε την ανάλογη συνάρτηση για προσδιορισμό της ύπαρξης συνεχόμενων θέσεων. Ανεξάρτητα με το αποτέλεσμα της συνάρτησης, όταν επιστρέφεται το αποτέλεσμα, κάνουμε signal το condition variable seatsThresholdCond έτσι ώστε να ειδοποιηθούν τα υπόλοιπα Threads που ίσως περιμένουν για να χρησιμοποιήσουν - κλειδώσουν το Mutex seatsMutex, για να ξεκινήσουν και αυτά την αναζήτηση θέσεων, και κάνουυμε έπειτα unlock το seatsMutex, για να μπορεί να το κλειδώσει κάποιος άλλος. Αν η απάντηση της συνάρτησης υπολογισμού θέσεων είναι θετική (return 2 από την συνάρτηση), τότε κρατάμε εκείνη τη χρονική στιγμή στη μεταβλητή telDisconnect τοου timespec struct και ο πελάτης προωθείται στον διαθέσιμο ταμία. Περιμένει να γίνει κάποιος διαθέσιμος με ανάλογο τρόπο με τον τηλεφωνητή, και όταν συνδεθεί κρατάμε πάλι εκείνη τη χρονική στιγμή στο cashConnect, και προσθέτουμε τον χρόνο cashConnect-telDisconnect στο totalWaitingTime. Περιμένει ο ταμίας για τυχαίο διάστημα μεταξύ T_CASH_LOW, T_CASH_HIGH, και μετά για να γίνει η διαδικασία της πληρωμής καλείται η **calcPcardSuccess()**. Εάν αυτή αποτύχει, με πιθανότητα 0.1, απελευθερώνουμε στον αντίστοιχο πίνακα τις θέσεις που είχαμε κάνει κράτηση για αυτόν τον πελάτη (απόδοση τιμής -1 ξανά) και αυξάνουμε την μεταβλητη creditCancellation που δείχνει τις ακυρώσεις που έχουν συμβεί λόγω του ταμία. Εάν επιτύχει, κλειδώνουμε το mutex addToBalance και προσθέτουμε στο ολικό μας χρηματικό υπόλοιπο, το σύνολο της τιμής των εισιτηρίων αυτού του πελάτη. Αφού κάνουμε αυτον τον υπολογισμό, το ξεκλειδώνουμε για να το χρησιμοποιήσει στη συνέχεια όποιο Thread το χρειάζεται.

Επίσης εδώ αυξάνουμε τον αριθμό των επιτυχών συναλλαγών (που έχουν ολοκληρωθεί, οι θέσεις έχουν κρατηθεί και οι πελάτες έχουν πληρώσει). Επίσης μετά την ολοκλήρωση της συνεργασίας με τον ταμία κλειδώνουμε και αυξάνουμε την μεταβλητή totalSupportTime η οποία δείχνει τον ολικό χρόνο εξυπηρέτησης των πελατών.

- waitingTime: Double στο οποίο αποθηκεύεται το άθροισμα χρόνων αναμονής για κάθε πελάτη.

- serviceTime: Double στο οποίο αποθηκεύεται το άθροισμα χρόνων εξυπηρέτησης για κάθε πλάτη.

- currentTelInUsed: Αριθμός τηλεφωνητών που είναι ήδη απασχολημένοι.

- currentCashInUsed: Αριθμός ταμείων που είναι ήδη απασχολημένοι.

- balance: Σύνολο εσόδων από πώληση εισιτηρίων.