

ΑΝΑΦΟΡΑ 1^{ης} ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗΣ ΑΣΚΗΣΗΣ

Μέλη ομάδας

- 1) Ανδρέας Πολυχρονάκης (Αριθμός Μητρώου: 3170140)
- 2) Άγγελος Τσελές (Αριθμός Μητρώου: 3170160)
- 3) Χρήστος Μπουκουβάλας (Αριθμός Μητρώου: 3170118)

Στο header file, αρχικά δηλώθηκαν όλες οι απαραίτητες βιβλιοθήκες, τις οποίες πρόκειται να χρησιμοποιήσουμε στο κυρίως πρόγραμμα. Ακολούθως, ορίστηκαν οι απαραίτητες σταθερές, όπως υποδεικνύει η εκφώνηση. Τέλος, ορίζεται η βασική συνάρτηση customer, την οποία πρόκειται να τροποποιήσουμε εντός του αρχείου .c .

Στο αρχείο .c , κάνουμε include το παραπάνω header καθώς και την βιβλιοθήκη time, την οποία χρειαζόμαστε για τον υπολογισμό των χρόνων. Ακολούθως, δηλώνουμε και σε ορισμένες περιπτώσεις ,κίόλας, αρχικοποιούμε όλες τις απαραίτητες μεταβλητές.

Η συνάρτηση customer αποτελεί τον βασικό κορμό πάνω στον οποίο βασίζεται σχεδόν εξ 'ολοκλήρου το πρόγραμμα μας. Η συνάρτηση αυτή παίρνει ως όρισμα την μεταβλητή id , η οποία είναι διαφορετική για κάθε πελάτη. Αρχικά, κλειδώνουμε το mutex και με μία while συνθήκη εξετάζουμε τον αριθμό των διαθέσιμων τηλεφωνητών εκείνη την στιγμή. Αν δεν υπάρχουν διαθέσιμοι τηλεφωνητές, τότε ο πελάτης περιμένει στην αναμονή το οποίο επιτυγχάνεται με την pthread_cond_wait. Προφανώς, αφότου εξυπηρετηθεί ένας πελάτης, τότε ο επόμενος θα μπορέσει να εξυπηρετηθεί από τον διαθέσιμο τηλεφωνητή. Μόλις ολοκληρωθεί το βήμα της αναμονής για τον κάθε πελάτη, τότε υπολογίζεται ο χρόνος αναμονής και προστίθεται κάθε φορά στην μεταβλητή totalWaiting. Ο χρόνος αναμονής υπολογίζεται καλώντας δύο φορές την μέθοδο clock_gettime, μία φορά πριν και μετά την διαδικασία που αναφέρθηκε παραπάνω.

Στην συνέχεια, υπολογίζουμε τον αριθμό εισιτηρίων που θέλει ο πελάτης να αγοράσει, κάνοντας χρήση της μεθόδου rand_r .Ο τηλεφωνητής ψάχνει να δει αν υπάρχουν διαθέσιμες θέσεις, για το οποίο χρειάζεται χρόνο t, ο οποίος υπολογίζεται και αυτός με την χρήση της rand_r.

Ακολούθως, διακρίνουμε περιπτώσεις:

1^η περίπτωση

Αν ο αριθμός των αγορασμένων εισιτηρίων είναι μικρότερος από 250 και ο αριθμός αγορασμένων συν τον αριθμό των εισιτηρίων που θέλει να κλείσει ο συγκεκριμένος πελάτης, είναι πάνω από 250. Στην περίπτωση αυτή, εκτυπώνεται το κατάλληλο μήνυμα, το mutex ξεκλειδώνει και το thread τερματίζει.

2^η περίπτωση

Αν ο αριθμός των αγορασμένων εισιτηρίων είναι 250 και πάνω. Τότε εκτυπώνεται το κατάλληλο μήνυμα, το mutex ξεκλειδώνει και το thread τερματίζει.

3^η περίπτωση

Στην περίπτωση αυτή, δεσμεύονται τα εισιτήρια που επέλεξε ο πελάτης και διακρίνουμε δύο υποπεριπτώσεις.

ΑΝΑΦΟΡΑ 1^{ης} ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗΣ ΑΣΚΗΣΗΣ

- 1) Η πληρωμή απέτυχε. Στην περίπτωση αυτή, εκτυπώνεται το κατάλληλο μήνυμα και τα εισιτήρια του πελάτη αποδεσμεύονται. Στο τέλος το mutex ξεκλειδώνει και το thread τερματίζει.
- 2) Η πληρωμή έγινε με επιτυχία. Στην περίπτωση αυτή, εκτυπώνονται τα κατάλληλα μηνύματα και γίνονται οι απαραίτητοι υπολογισμοί. Τέλος, το mutex ξεκλειδώνει και το thread τερματίζει.

Σημειώνεται ότι στο τέλος όλων των παραπάνω περιπτώσεων υπολογίζεται ο χρόνος εξυπηρέτησης, ακριβώς με τον ίδιο τρόπο αντίστοιχα που περιγράψαμε παραπάνω για τον χρόνο αναμονής.

Στην main, παίρνουμε ως ορίσματα των αριθμό των πελατών και τον σπόρο της γεννήτριας από το terminal και γίνονται οι απαραίτητοι έλεγχοι. Μετά, με μία for loop δημιουργείται thread για κάθε πελάτη, ενώ γίνεται και απαραίτητη χρήση της pthread_join .

Τέλος, γίνονται οι υπολογισμοί και οι εκτυπώσεις των μηνυμάτων που ζητάει η εκφώνηση, αποδεσμεύεται μνήμη, τα mutex καταστρέφονται και το πρόγραμμα τερματίζει.