

## Project 1

Μπιλέτσιου Φλωρίνα 2114033

Θωμάς Αγγελόπουλος 2113043

### Δομή

**2114033\_2113043\_proj1.h** : Αρχείο δηλώσεων και αρχικοποιήσεων

**2114033\_2113043\_proj1.c** : Κυρίως πρόγραμμα, θα αναλυθεί παρακάτω

**proj1** : Εκτελέσιμο αρχείο. Παράγωγο της μεταγλώττισης

**test\_proj1.sh** : shell αρχείο testing για 100 πελάτες με seed 1000

### Εκτέλεση

Παρακαλώ να εκτελεστεί σε οποιοδήποτε περιβάλλον Unix το οποίο περιέχει το gcc εγκατεστημένο .

Για custom εκτέλεση:

```
gcc 2114033_2113043_proj1.c -o proj1 -lpthread
```

```
./proj1 «number of customers» «seed»
```

Για την testing εκτέλεση :

```
sh test_proj1.sh
```

### Ανάλυση Κυρίως Προγράμματος

**Γραμμές 5-14** : Βιβλιοθήκες που θα χρειαστούν . Συμπεριλαμβάνουμε και το αρχείο που δημιουργήσαμε με τις δηλώσεις και τις αρχικοποιήσεις κάποιων κοινών μεταβλητών. Περιληπτικά οι βιβλιοθήκες «unistd.h» «pthread.h» χρησιμοποιούνται για την δημιουργία και χρήση των mutex και condition variables , οι «time.h», «sys/time.h» και «signal.h» για την διαχείριση των "ρολογιών", σημάτων κτλ.

**Γραμμες 17-24 και 227-253** : Δήλωση και initialization των mutex και condition variable για τον συγχρονισμό των τηλεφωνητών , της μέσου χρόνου αναμονής και εξυπηρέτησης των πελατών, του πλάνου θέσεων και των οικονομικών πληροφοριών αντίστοιχα.

**Γραμμές 26-28, 35-36 και 312-314:** Μετάβλητές και συνάρτηση για την καταμέτρηση και υπολογισμό του μέσου χρόνου αναμονής και εξυπηρέτησης των πελατών συνολικά.

**Γραμμές 38-41:** Συνάρτηση για το μήνυμα αναμονής , χρήση write για την πιο stable λειτουργία του προγράμματος.

### Στη main():

**Γραμμές 208-215:** Έλεγχος του πλήθους των παραμέτρων του προγράμματος και ανάθεση των πληροφοριών σε ανάλογες μεταβλητές.

**Γραμμές 218-225:** Ελευθερώνουμε τις θέσεις του θεάτρου . Όταν η θέση έχει την τιμή 0 σημαίνει πως είναι ελεύθερη.

**Γραμμές 255-267:** Δημιουργία των νημάτων των πελατών. Τα thread ids αποθηκεύονται στον ανάλογο πίνακα tid[] , η συνάρτηση με το action του thread είναι η client\_func(), η οποία θα αναλυθεί παρακάτω και περνάμε σαν παράμετρο τον αριθμό της επανάληψης +1 , αρραυσιαστικά τον αριθμό πελάτη τύπου 1,2,3,4,.....,N\_cust.

**Γραμμές 268-270:** Περιμένουμε όλα τα threads να κάνουν return με τιμή NULL.

**Γραμμές 272-297:** Εμφάνιση τελικών αποτελεσμάτων . Το πλάνο των θέσεων αρχικά πιο εικονικά σαν πλάνο θέσεων, μετά αναλυτικά θέση/πελάτη και τέλος το balance, τον αριθμό των συναλλαγών και τους μέσους χρόνους αναμονής και εξυπηρέτησης.

**Γραμμές 299-309:** Απελευθέρωση μνήμης και εξόδος.

### Στην client\_func:

**Γραμμή 45:** Ξεκινάει ο χρόνος εξυπηρέτησης εφόσον ξεκινάνε οι ενέργειες του πελάτη

**Γραμμές 47-49:** Ανάθεση του client id για να απελευθερώσουμε την μνήμη που χρησιμοποιήσαμε για να το περάσουμε ως παράμετρο και να είναι πιο ευχρηστο.

**Γραμμές 51-64:** Δήλωση του timer για το μήνυμα αναμονής με χρήση SIGALRM . Με την μεταβλητή .it\_value.tv\_sec δηλώνουμε πως θέλουμε να εμφανίζεται το μήνυμα ανά T\_ihitiko δευτερόλεπτα και με την it\_interval.tv\_sec πως θέλουμε να επαναλαμβάνεται.

**Γραμμές 68-86:** Αν στο thread εκείνη την στιγμή δεν υπάρχει διαθέσιμος τηλεφωνητής "παγιδεύεται" μέσα στο while μέχρι κάποιος τηλεφωνητής να απελευθερωθεί.

**Γραμμές 70-74:** Εκκίνηση του timer για την εμφάνιση του μηνύματος αναμονής και παράλληλα έλεγχος για σφάλματα.

**Γραμμή 78:** Το νήμα "παγώνει" στο συγκεκριμένο σημείο μέχρι να γίνει signal για το condition variable.

**Γραμμή 79:** Εφόσον βρισκόμαστε στο σημείο αυτό σημαίνει πως έχει "ξεπαγώσει" η εκτέλεση του νήματος και πιθανότατα να υπάρχει διαθέσιμος τηλεφωνητής αρρα αυτό σημαίνει πως ο χρόνος αναμονής τελείωσε.

**Γραμμές 81-83:** Πρόσθεση του χρόνου αναμονής του συγκεκριμένου νήματος στο συνολικό χρόνο αναμονής με τον ανάλογο συγχρονισμό για να μην υπάρξει το φαινόμενο του deadlock , με τον ανάλογο mutex.

**Γραμμές 88-91:** Είτε βρισκόμαστε σε αυτό το σημείο του κώδικα μετά από τον βρόγχο του while των μη διαθέσιμων τηλεφωνητών, είτε εξ αρχής υπήρχε διαθέσιμος τηλεφωνητής, σε αυτό στο σημείο πρέπει να δηλώσουμε πως ένας από τους τηλεφωνητές δεν είναι διαθέσιμος πια γιατί θα εξυπηρετεί το συγκεκριμένο πελατη-νήμα. Με τον ανάλογο συγχρονισμό με το mutex . Το κλειδώνουμε , κάνουμε την αλλαγή , και το ξεκλειδώνουμε για να μπορεί να το

χρησιμοποιήσει το οποιοδήποτε άλλο thread. Η ίδια στρατηγική ακολουθείται σε κάθε συγχρονισμό με mutex (κλείδωμα,επεξεργασία,ξεκλείδωμα).

**Γραμμές 93-104:** Κάνουμε έλεγχο για την πληρότητα του θεάτρου πριν ξεκινήσουμε την εξυπηρέτηση του πελάτη . Αν είναι πλήρες δεν υπάρχει νόημα να συνεχίσει ο πελάτης αφού οι επιλογές και να γίνουν δεν θα μπορέσει να αγοράσει εισιτήρια τελικά. Επομένως ενημερώνουμε τον χρήστη , μετά “απελευθερώνουμε” τον τηλεφωνητή που χρησιμοποιούμε και τέλος κάνουμε signal το condition variable ώστε να ξεκινήσει και το νήμα που είναι στην αναμονή , αν υπάρχει κάποιο. Τελικά το thread αφού κάνει τα παραπάνω τερματίζει.

**Γραμμές 106-114 :** Ο τηλεφωνητής ρωτάει τον πελάτη το πλήθος των εισιτηρίων, ο πελάτης επιλέγει τυχαίο αριθμό εισιτηρίων ανάμεσα στα επιτρεπτά όρια (1-6 εισιτήρια).Αργότερα ο τηλεφωνητής προσομοιώνει τον έλεγχο της διαθεσιμότητας των εισιτηρίων με την συνάρτηση sleep() για τυχαίο αριθμό δευτερολέπτων απο 4 μέχρι 11.

**Γραμμές 116-119:** Πραγματικός έλεγχος διαθεσιμότητας εισιτηρίων για την συγκεκριμένη αγορά του πελάτη, αν δεν υπάρχει ο επιθυμητός αριθμός εισιτηρίων που ο πελάτης θέλει τότε η μόνη ενέργεια είναι η ενημέρωση του χρήστη και αργότερα η ενημέρωση χρόνου και η απελευθέρωση του τηλεφωνητή στην γραμμή 191 , η οποία βρίσκεται εκτός ελεγχου if καθώς πρόκειται για κοινή ενέργεια και στις δύο περιπτώσεις και δεν υπάρχει λόγος να γίνεται πολλές φορές.

**Γραμμές 120-189:** Περίπτωση που υπάρχουν τα επιθυμητά εισιτήρια , οπότε προχωράμε στην εξυπηρέτηση.

**Γραμμές 124-151:** Ενημέρωση του χρήστη , κλειδώνουμε τον ανάλογο mutex , αναζήτηση ελεύθερων θέσεων και όταν βρεθεί κάποια τότε αναθέτουμε στην θέση αυτή το id του πελάτη ώστε να γνωρίζουμε ποιες θέσεις του ανήκουν. Οι επαναλήψεις σταματάνε όταν “κλείσουμε” τον επιθυμητο αριθμό θέσεων με τον έλεγχο temp\_count == N\_cust\_tickets . Ο συγκεκριμένος βρόγχος θα μπορούσε να βρίσκεται και στην γραμμή 133 , αλλά έτσι θα επρεπε να περιμέναμε να ελένξουμε ολο τον πίνακα ενώ τώρα απλά σταματάμε όταν κλείσουμε τον αριθμο εισιτηρίων που ζήτησε ο πελάτης. Ξεκλείδωμα του mutex, και ένας προληπτικός , σχετικά περριτός , έλεγχος για τον αριθμό των εισιτηρίων που κρατήθηκαν. Τέλος αφού ενημερώσαμε για τις θέσεις που κρατήθηκαν ενημερώνουμε και για το κόστος αυτών.

**Γραμμές 153-155:** Επιλογή τυχαία αν η πληρωμή με κάρτα θα επιτύχει απο 1 μέχρι 100.

**Γραμμές 157-166:** Αν η πιθανότητα πληρωμής είναι απο 1 μέχρι 85 αυτό σημαίνει πως βρισκόμαστε στην πιθανότητα 85% που η πληρωμή είναι επιτυχής επομένως ενημερώνουμε τον χρήστη , με τον ανάλογο συγχρονισμό αλλάζουμε το Balance και τον αριθμό των συναλλαγών , ξεκλειδώνουμε και συνεχίζουμε στην γραμμή 191 που εξηγήσαμε παραπάνω τι συμβαίνει.

**Γραμμές 168-188:** Αυτή είναι η περίπτωση που η πληρωμή με κάρτα αποτύχει ενημερώνουμε τον χρήστη και με τον ανάλογο συγχρονισμό απελευθερώνουμε τις θέσεις που “κρατήσαμε” για τον συγκεκριμένο χρήστη.

**Γραμμές 191-194:** Εφόσον βρισκόμαστε σε αυτό το σημείο του κώδικα σημαίνει πως η εξυπηρέτηση του πελάτη τελείωσε επομένως με τον ανάλογο συγχρονισμό προσθέτουμε τον χρόνο εξυπηρέτησης του πελάτη στον γενικό μέσο χρόνο εξυπηρέτησης.

**Γραμμές 196-200:** Απελευθέρωση του τηλεφωνητή, signal to condition variable με τον ανάλογο συγχρονισμό και τέλος το νήμα τερματίζει με το return NULL .