



Περιγραφή Δομής Κώδικα

Το αρχείο κεφαλίδας είναι το p3150173-p3150090-p3120120-res1.h

Γραμμές 5-6 & 23:

```
#ifndef P3150173_P3150090_P3120120_RES1_H
#define P3150173_P3150090_P3120120_RES1_H

#endif
```

Έχουμε ένα #include guard. Αυτό αποτρέπει τη διπλή δήλωση του αρχείου κεφαλίδας σε περίπτωση που ο κώδικας συμπεριληφθεί ξανά.

Γραμμές 8-12:

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <pthread.h>
#include <stdbool.h>
```

Ενσωματώνουμε όλα τα αρχεία κεφαλίδας που χρειαζόμαστε στο .c αρχείο μας. Πιο συγκεκριμένα, το stdio.h περιέχει την δήλωση και την υλοποίηση της printf(). Το unistd.h παρέχει πρόσβαση στο API του POSIX. Το stdlib.h περιέχει την δήλωση και την υλοποίηση των exit(), rand_r(), atoi(), exit() και του macro RAND_MAX. Το pthread.h περιέχει την δήλωση και την υλοποίηση των pthread_mutex_init(), pthread_cond_init(), pthread_create(), pthread_join(), pthread_mutex_lock(), pthread_mutex_unlock(), pthread_cond_wait(), pthread_cond_broadcast(), pthread_mutex_destroy(), pthread_exit(), clock_gettime() και του macro CLOCK_REALTIME. Το stdbool.h περιέχει την δήλωση και την υλοποίηση του macro bool για τον τύπο δεδομένων boolean.

Γραμμές 14-21:

```
#define N_SEATS 250
#define N_TEL 8
#define N_SEAT_LOW 1
#define N_SEAT_HIGH 5
#define T_SEAT_LOW 5
#define T_SEAT_HIGH 10
#define P_CARD_SUCCESS 0.9
#define C_SEAT 20
```

Ορίζουμε τις σταθερές που ζητούνται από την εκφώνηση.

Το αρχείο κώδικα C είναι το p3150173-p3150090-p3120120-res1.c

Γραμμή 5:

```
#include "p3150173-p3150090-p3120120-res1.h"
```

Συμπεριλαμβάνουμε όλα τα περιεχόμενα του αρχείου p3150173-p3150090-p3120120-res1.h.

Γραμμές 7-69:

Δηλώνουμε όλες τις μεταβλητές και τις μεθόδους που θα χρησιμοποιήσουμε.

Γραμμές 71-126:

Η main συνάρτηση του προγράμματος

- **Γραμμές 73-87:** Ελέγχουμε τα ορίσματα που έχει δώσει ο χρήστης και αρχικοποιούμε τις μεταβλητές customer και seed.
- **Γραμμές 91-92:** Ορίζουμε τον πίνακα threads που περιέχει στοιχεία τύπου pthread_t και τον πίνακα id που αποθηκεύουμε το αναγνωριστικό του κάθε πελάτη.
- **Γραμμές 94-101:** Αρχικοποιούμε τα mutexes και ελέγχουμε εάν ήταν επιτυχείς όλες οι αρχικοποιήσεις με την check_rc().
- **Γραμμές 103-114:**
 - Στην γραμμή 103 ξεκινάμε την χρονομέτρηση με την startTimer().
 - Στις γραμμές 105-108 ο επαναληπτικός βρόχος δημιουργεί τα threads δίνοντας τους σαν όρισμα τον πίνακα id.
 - Στις γραμμές 110-112 ο επαναληπτικός βρόχος συγχρονίζει όλα τα νήματα-παιδιά με το νήμα της main.
 - Στην γραμμή 114 σταματάμε την χρονομέτρηση με την stopTimer().
- **Γραμμές 116-123:** Καταστρέφουμε όλα τα νήματα και ελέγχουμε εάν καταστράφηκαν επιτυχώς.
- **Γραμμή 125:** Καλούμε την συνάρτηση printf η οποία εμφανίζει όλα τα απαραίτητα στοιχεία της κράτησης.

Γραμμές 128-208: Η customer() είναι η start routine που χρησιμοποιούμε για τα νήματα.

Παίρνει ένα όρισμα τύπου void pointer και επιστρέφει έναν void pointer.

- **Γραμμή 130:** Δηλώνουμε και αρχικοποιούμε την μεταβλητή id για να αποθηκεύσουμε το αναγνωριστικό του κάθε πελάτη.
- **Γραμμή 132:** Δηλώνουμε 4 struct τύπου timespec για να κρατάμε τις χρονικές στιγμές που αρχίζει και τελειώνει η αναμονή και η εξυπηρέτηση του κάθε πελάτη.
- **Γραμμές 136-154:** Κλειδώνουμε τους τηλεφωνητές και ελέγχουμε αν υπάρχει κάποιος διαθέσιμος, αν ναι τους ξεκλειδώνουμε και συνεχίζουμε την εκτέλεση της customer, αν όχι περιμένουμε να λάβουμε σήμα για να συνεχίσουμε την εκτέλεση.
- **Γραμμές 158-188:**
 - Στην γραμμή 158 ελέγχουμε εάν υπάρχουν διαθέσιμες θέσεις για κράτηση με την checkRemainingSeats(). Εάν δεν υπάρχουν εμφανίζεται στην κονσόλα πως το θέατρο γέμισε, το if αποτυγχάνει και η εκτέλεση μεταφέρεται στην γραμμή 189. Εάν υπάρχουν η εκτέλεση μεταφέρεται στις γραμμές 159-187.
 - Στην γραμμή 160 καλούμε την choiceRandom() με ορίσματα τα N_SEAT_LOW και N_SEAT_HIGH για να αρχικοποιήσουμε την μεταβλητή seats που αναπαριστά τις θέσεις που επέλεξε ο πελάτης.

- ο Στην γραμμή 162 καλούμε την `sleep()` με όρισμα `sleepRandom(T_SEAT_LOW, T_SEAT_HIGH)` γιατί η τηλεφωνήτρια θέλει κάποιο χρόνο για να βρει εάν υπάρχουν διαθέσιμες θέσεις.
- ο Στην γραμμή 164 καλούμε `checkAvailableSeats()` με όρισμα το `seats` η οποία ελέγχει εάν υπάρχουν αρκετές θέσεις με βάση αυτές που ζήτησε ο πελάτης. Εάν υπάρχουν τότε ο έλεγχος περνάει στις γραμμές 165-186.
- ο Στην γραμμή 166 κάνουμε κράτηση τις θέσεις με την `bookSeats()` η οποία παίρνει τα ορίσματα `seats, id`.
- ο Στην γραμμή 168 καλούμε την `POS()` η οποία ελέγχει εάν η συναλλαγή με την κάρτα ήταν επιτυχής χρησιμοποιώντας την `cardRandom()`. Εάν ήταν τότε ο έλεγχος περνάει στις γραμμές 169-185. Εάν δεν ήταν επιτυχής η συναλλαγή με την κάρτα τότε αποδεσμεύουμε τις θέσεις της κράτησης και εμφανίζουμε στην κονσόλα πως η κράτηση ματαιώθηκε γιατί η συναλλαγή με την πιστωτική κάρτα δεν έγινε αποδεκτή.
- ο Στις γραμμές 169-185 καταγράφουμε την συναλλαγή, τα έσοδα και εμφανίζεται στον πελάτη το αναγνωριστικό της συναλλαγής του, οι θέσεις του και το κόστος κάνοντας χρήση του `logTransaction()`, ενός επαναληπτικού βρόχου και της `Cost()` με όρισμα `seats`.
- **Γραμμές 190-207:** Καταγράφουμε πως ο πελάτης εξυπηρετήθηκε και αποδεσμεύουμε την τηλεφωνήτρια ενημερώνοντας κίόλας τον μέσο χρόνο εξυπηρέτησης. Τέλος στην γραμμή 207 κάνουμε `pthread_exit(NULL)` για να αποδεσμεύσουμε το νήμα μας.

Γραμμές 210-221:

Υλοποιούμε την `sleepRandom`, την `choiceRandom` και την `cardRandom` οι οποίες επιστρέφουν από έναν τυχαίο αριθμό ο οποίος ανήκει σε ένα εύρος αντιστοίχως.

Γραμμές 223-243:

- **Γραμμές 223-249:** Η `startTimer()` αποθηκεύει στην μεταβλητή `requestStart` την τιμή του συστήματος του ρολογιού τη στιγμή που θα κληθεί.
- **Γραμμές 231-237:** Η `stopTimer()` αποθηκεύει στην μεταβλητή `requestEnd` τιμή του συστήματος του ρολογιού τη στιγμή που θα κληθεί.
- **Γραμμές 239-245:** Η `Clock()` εμφανίζει ένα ψηφιακό ρολόι στην κονσόλα.

Γραμμές 245-284:

- **Γραμμές 245-251:** Η `Cost()` δέχεται ως όρισμα την μεταβλητή `numOfSeats` για να υπολογίσει το συνολικό ποσό που πρέπει να πληρώσει ο πελάτης. Αυξάνει την μεταβλητή `profit` με βάση το ποσό και επιστρέφει το ποσό.
- **Γραμμές 253-258:** Η μέθοδος `logTransaction()` δεν παίρνει ορίσματα, ενημερώνει την μεταβλητή `transactions` αυξάνοντας την κατά 1 και επιστρέφει τον κωδικό (αναγνωριστικό) της πληρωμής.
- **Γραμμές 260-271:** Η μέθοδος `bookSeats()` δέχεται ως όρισμα δύο μεταβλητές την `numOfSeats` και την `custID`. Δεσμεύει `numOfSeats` θέσεις στον πίνακα `seatsPlan` για το `custID` και μειώνει την μεταβλητή `remainingSeatsPtr` κατά `numOfSeats`.
- **Γραμμές 273-284:** Η μέθοδος `bookSeats()` δέχεται ως όρισμα δύο μεταβλητές την `numOfSeats` και την `custID`. Διαγράφει όλες τις αλλαγές που έχει κάνει η `bookSeats()` για το ίδιο `custID`.

Γραμμές 286-295:

Η `check_rc()` δέχεται ως όρισμα μια μεταβλητή `rc` τύπου `int` και σε περίπτωση που η μεταβλητή `rc` ισούται με 1 εμφανίζει ένα μήνυμα σφάλματος και στην συνέχεια τερματίζει την εκτέλεση του προγράμματος με την `exit()`. Την χρησιμοποιούμε για να ελέγξουμε εάν η μέθοδοι `pthread_mutex_init()`, `pthread_cond_init()`, `pthread_create()`, `pthread_join()`, `pthread_mutex_lock()`, `pthread_mutex_unlock()`, `pthread_cond_wait()`, `pthread_cond_broadcast()`, `pthread_mutex_destroy()` εκτελέστηκαν με επιτυχία.

Γραμμές 297-333:

- **Γραμμές 297-308:** Η `checkAvailableSeats()` ελέγχει εάν το πλήθος των θέσεων που ζήτησε να κλείσει ο πελάτης είναι μικρότερο ή ίσο από τις θέσεις που απομένουν ελεύθερες στο θέατρο. Επιστρέφει `true` ή `false` εάν ο έλεγχος αληθεύει ή όχι αντίστοιχα.
- **Γραμμές 310-321:** Η συνάρτηση `checkRemainingSeats()` ελέγχει εάν υπάρχουν διαθέσιμες θέσεις στο θέατρο προς κράτηση. Επιστρέφει `true` ή `false` εάν ο έλεγχος αληθεύει ή όχι αντίστοιχα.
- **Γραμμές 323-334:** Η συνάρτηση `POS()` ελέγχει εάν η κάρτα έγινε αποδεκτή ή όχι. Επιστρέφει `true` ή `false` εάν ο έλεγχος αληθεύει ή όχι αντίστοιχα.

Γραμμές 335-366:

- **Γραμμές 335-346:** Η `printDuration()` εμφανίζει στην κονσόλα την διάρκεια εκτέλεσης όλων των παιδιών νημάτων της `main`.
- **Γραμμές 348-364:** Η `printSeatsPlan()` εμφανίζει στην κονσόλα το πλάνο των θέσεων του θεάτρου με τον εξής τρόπο: σε κάθε γραμμή εμφανίζονται 4 θέσεις χωριζόμενες από κάθετες παύλες `"|"`.
- **Γραμμές 366-384:** Η `printInfo()` εμφανίζει στην κονσόλα την Έναρξη και την Λήξη της χρονομέτρησης της εκτέλεσης όλων των νημάτων-παιδιών της `main`, καλεί την `printDuration()`, την `printSeatsPlan()` και στην συνέχεια τον μέσο χρόνο αναμονής, εξυπηρέτησης, το πλήθος των πελατών που εξυπηρετήθηκαν, το πλήθος των θέσεων που δεσμεύθηκαν, το πλήθος των θέσεων που έμειναν κενές, το πλήθος των συναλλαγών και το πλήθος των κερδών.