



Operating Systems

Assignment 2022



Χρήστος Πατρινόπουλος 3200150

Αλβιόνα Μάντσο 3200098

Στο *global* επίπεδο υπάρχουν:

- **Μεταβλητές** που αφορούν τα νήματα/πελάτες και ενημερώνονται από αυτά. Αυτές είναι:
 - ο λογαριασμός εσόδων (iban)
 - ο αθροίσματα έκβασης συναλλαγών (tickets_failed, credit_failed_zA, credit_failed_zB, txn_completed)
 - ο συνολικοί χρόνοι αναμονής → (tel_standby, cash_standby) και εξυπηρέτησης → (total_serving_time)
- **Πίνακας θέσεων** (seats) μονοδιάστατος (τόσες θέσεις όσες και η χωρητικότητα του θεάτρου), όπου κάθε μία έχει τιμή 0 αν είναι διαθέσιμη ή διαφορετικά το id του πελάτη που τη δέσμευσε.
- **Πίνακας των διαθέσιμων θέσεων ανά σειρά** (avail_seats_pRow) μονοδιάστατος (τόσες θέσεις όσες και το άθροισμα των σειρών των δύο ζωνών).
- **Μεταβλητές αριθμού διαθέσιμων τηλεφωνητών** (avail_lines) και διαθέσιμων ταμιών (avail_cash).
- **Μεταβλητές συνθήκης για τους τηλεφωνητές** (line_cond) και τους ταμίες (cash_cond).
- **Mutexes** για τηλεφωνητές (line_mutex), ταμίες (cash_mutex), για χειριστές ζώνης A και B (zoneA_mutex, zoneB_mutex), για εκτέλεση συναλλαγής (txn_mutex) και για κλείδωμα οθόνης εξόδου (print_mutex).

Στη *main* συνάρτηση:

ελέγχονται κατάλληλα οι δύο είσοδοι του χρήστη για αριθμό πελατών (N_cust) και για το seed της συνάρτησης rand (init_seed). Αρχικοποιούνται οι μονοδιάστατοι πίνακες που αφορούν το πλάνο θέσεων, τα mutexes και οι μεταβλητές συνθήκης. Γίνεται η δημιουργία πίνακα νημάτων/πελατών (customers) και πίνακα που συγκρατεί τα id των νημάτων (cust_id) με καταχώρηση δυναμικής μνήμης, αφού ο αριθμός πελατών (N_cust) δίνεται από το χρήστη.

Σε loop δημιουργούνται τα νήματα, μετά από τυχαίο χρόνο (res_time).

Σε loop γίνεται η αναμονή τερματισμού των νημάτων μέσω της join.

Εκτυπώνεται το πλάνο θέσεων, τα συνολικά έσοδα (box office), τα ζητούμενα στατιστικά έκβασης των συναλλαγών και μέσοι χρόνοι αναμονής και εξυπηρέτησης πελατών.

Επίσης εκτυπώνουμε έναν **πίνακα-πλάνο του θεάτρου** που δείχνει ποιες θέσεις και σε ποιες σειρές είναι κατειλημμένες, καθώς και το από ποιον πελάτη. Η αρίθμηση ξεκινάει από το 0.

Τέλος απελευθερώνεται η δυναμικά δεσμευμένη μνήμη, τα mutexes και οι μεταβλητές συνθήκης.

threads

Κάθε **νήμα** εκκινεί την εκτέλεσή του από τη συνάρτηση **reservation()** με όρισμα το id που του δόθηκε. Υπολογίζεται το ατομικό seed κάθε νήματος/πελάτη (this_seed) ως άθροισμα του αρχικού (init_seed) και του id του (cust_id).

Υπολογίζει πιθανοτικά τη ζώνη (zone) που θα επιχειρήσει να κλείσει εισιτήρια (enum zones)* και το πλήθος των εισιτηρίων (tickets).

Στη συνέχεια αναμένει τηλεφωνητή και μόλις βρεθεί διαθέσιμος τον δεσμεύει μειώνοντας τη μεταβλητή (avail_lines). (χρήση line_mutex και line_cond)

Καλείται η συνάρτηση **phone_operator()** που αναλαμβάνει τη λειτουργία του τηλεφωνητή και ενημερώνει την τιμή των μεταβλητών flag* και first_seat, όπου flag = true όταν δεσμεύονται θέσεις και first_seat ο αριθμός της 1^η θέσης της κράτησης με βάση τον μονοδιάστατο πίνακα (seats). flag = false όταν δεν υπάρχουν επαρκείς διαθέσιμες θέσεις.

Το νήμα αποδεσμεύει τον τηλεφωνητή (avail_lines), ενημερώνει το συνολικό χρόνο αναμονής τηλεφωνητή και αν flag = false αυξάνει τις αποτυχημένες συναλλαγές λόγω μη εύρεσης εισιτηρίων και ενημερώνει το συνολικό χρόνο εξυπηρέτησης, αφού το νήμα πρόκειται να τερματίσει. (χρήση line_mutex και line_cond)

Στην περίπτωση flag = false εκτυπώνεται κατάλληλο μήνυμα και το νήμα/πελάτης τερματίζει.

Αλλιώς, υπολογίζεται το κόστος κράτησης των εισιτηρίων με βάση τη ζώνη και η εκτέλεση συνεχίζεται με τον ταμιά.

Σε αυτό στο σημείο το νήμα-πελάτης αναμένει ταμιά και μόλις βρεθεί διαθέσιμος τον δεσμεύει μειώνοντας τη μεταβλητή (avail_cash). (χρήση cash_mutex και cash_cond)

Καλείται η συνάρτηση **cashier()** που αναλαμβάνει τη λειτουργία ταμιά.

Το νήμα αποδεσμεύει τον ταμιά (avail_cash), ενημερώνει το συνολικό χρόνο αναμονής ταμιά και το συνολικό χρόνο εξυπηρέτησης. (χρήση cash_mutex και cash_cond)

Το νήμα/πελάτης **τερματίζει**.

Outline

```
void* reservation( void* id)
```

```
    calculate inner seed;
```

```
    determine zone and tickets;
```

```
    wait for an available phone operator;
```

```
    call phone_operator( ... );
```

```
    free the phone operator;
```

```
    if available seats were found, calculate cost and move on to cashier, else terminate;
```

```
    call cashier( ... );
```

```
    free the cashier;
```

```
    terminate;
```

Η συνάρτηση **phone_operator()** καλεί τη `sleep` για ένα τυχαίο χρόνο (`seat_time`) και στη συνέχεια ανάλογα με τη ζώνη που έχει επιλεγεί (`zone`), κλειδώνει το αντίστοιχο τμήμα του πίνακα θέσεων (`seats`) και του πίνακα διαθέσιμων θέσεων σειράς (`avail_seats_pRow`). Αυτό επιτυγχάνεται με το `lock` στο κατάλληλο `mutex` (`zoneA_mutex` ή `zoneB_mutex`). Αναζητά στον πίνακα τις πρώτες συνεχόμενες ελεύθερες θέσεις (όσες προσδιορίζει το `tickets`) με τη βοήθεια της συνάρτησης `seats_searching()` και ενημερώνει το `flag` (εύρεση θέσεων), το `first_seat` (1^η θέση κράτησης) και κατάλληλα τους δύο πίνακες, δεσμεύοντας τις θέσεις.

Η συνάρτηση **cashier()** καλεί τη `sleep` για ένα τυχαίο χρόνο (`cash_time`) και υπολογίζει πιθανοτικά την επιτυχία ή μη της κράτησης (`succ_txn`)*. Αν `succ_txn = true` τότε ενημερώνει τη μεταβλητή λογαριασμού εσόδων (`iban`) και τον αριθμό επιτυχημένων συναλλαγών (`txn_completed`), με χρήση του `txn_mutex`. Εκτυπώνει μήνυμα στον πελάτη με πληροφορίες για τις θέσεις και το κόστος κράτησης και τερματίζει.

Αν `succ_txn = false` τότε η συναλλαγή ακυρώνεται και επιστρέφει τη διαθεσιμότητα των θέσεων στο πλάνο. Ανάλογα με τη ζώνη κάνει `lock` στο αντίστοιχο `mutex` (`zoneA_mutex` ή `zoneB_mutex`) και με τη βοήθεια των μεταβλητών `first_seat` και `tickets` επαναφέρει στις αντίστοιχες θέσεις του `seats` το 0 (άρα διαθέσιμες) και στον `avail_seats_pRow` τις διαθέσιμες θέσεις της σειράς στην οποία είχε γίνει κράτηση. Ενημερώνει τον αριθμό αποτυχημένων συναλλαγών λόγω κάρτας, εκτυπώνει μήνυμα αποτυχίας ολοκλήρωσης συναλλαγής στον πελάτη και τερματίζει.

* Σημειώσεις:

Οι κλήσεις των `pthread` συναρτήσεων για `mutex_init()`, `cond_init()`, `lock()`, `unlock()`, `wait()`, `signal()`, `mutex_destroy()`, `cond_destroy()` καλούνται από αντίστοιχου ονόματος συναρτήσεις που επιπλέον εκτελούν έλεγχο του κωδικού επιστροφής της συνάρτησης και σε περίπτωση σφάλματος τερματίζει το νήμα ή το πρόγραμμα.

Στο αρχείο δηλώσεων (`header file`) έχει υλοποιηθεί **bool** τύπος μέσω **enum** που χρησιμοποιείται στις μεταβλητές `flag` και `succ_txn`.

Επίσης έχει υλοποιηθεί το **enum zones** με δυνατές τιμές τα `zoneA` και `zoneB` που χρησιμοποιείται στη μεταβλητή `zone`, για καθορισμό της ζώνης επιλογής εισιτηρίων.