Εργασία-Λειτουργικά Συστήματα

Στην συγκεκριμένη άσκηση, υλοποιούμε ένα σύστημα αγοράς θέσεων σε θεατρικές παραστάσεις στο οποίο πραγματοποιούνται παράλληλες κρατήσεις θέσεων, και γι' αυτό γίνεται χρήση νημάτων (threads).

Η γενική ιδέα για την υλοποίηση του κώδικα είναι η εξής:

- Δημιουργούμε ένα struct για τους customers, το οποίο περιέχει τις απαραίτητες ιδιότητες για κάθε customer.
- Δημιουργούμε και έναν πίνακα από pointers σε customers (customers_array), ο οποίος θα περιέχει τους customers των οποίων η συναλλαγή ολοκληρώθηκε επιτυχώς (αξιοποιείται για τα τελικά prints).
- Έναν δισδιάστατο πίνακα int seats [NUM_ZONE_A + NUM_ZONE_B][NUM_SEAT], ο οποίος θα περιέχει τους αριθμούς 0 ή 1, ανάλογα με το αν η θέση είναι διαθέσιμη για να προχωρήσει ο πελάτης στην κράτησή της ή όχι αντίστοιχα.
- Βοηθητικές μεταβλητές για την διεκπεραίωση των ζητουμένων.

Ο κώδικας μας αποτελείται επίσης από τις εξής συναρτήσεις:

1. void elegxos(int rc): βοηθητική συνάρτηση η οποία εμφανίζει μήνυμα λάθους σε περίπτωση αποτυχίας κάποιας λειτουργίας των νημάτων (δημιουργία,mutex_lock,mutex_unlock,cond).

2. double random_number(int x, int y, int id): βοηθητική συνάρτηση για την παραγωγή ενός τυχαίου αριθμού στο εύρος (x,y).

3. void *call(void *x): η συγκεκριμένη συνάρτηση αντιπροσωπεύει τον τηλεφωνητή. Αναλυτικά, η λειτουργία της είναι η εξής:

   Αρχικά, δημιουργούμε ένα instance του struct customers για κάθε thread (πελάτη). Έπειτα, περιμένουμε μέχρι να υπάρξει κάποιος διαθέσιμος τηλεφωνητής. Από τη στιγμή που βρεθεί, παράγονται αυτόματα τυχαία δεδομένα σχετικά με τις ζητούμενες θέσεις, τη ζώνη θέσεων και τον χρόνο αναζήτησης.

   Έπειτα, ψάχνουμε στον πίνακα seats για διαθέσιμες θέσεις, ανάλογα με την τιμή της αντίστοιχης θέσης του πίνακα.
   Αν υπάρχουν διαθέσιμες θέσεις αλλάζουμε τις τιμές του πίνακα seats από 0 σε 1 και η διαδικασία συνεχίζεται με την κλήση της συνάρτησης cashier.
   Σε αντίθετη περίπτωση, τυπώνεται το αντίστοιχο μήνυμα και γίνεται έξοδος από το νήμα.

4. int cashier(void * customer1, int rc, struct timespec * start_eksypiretisi, struct timespec * stop_eksypiretisi):
   Για όποιον πελάτη προχωρήσει σε πληρωμή της κράτησής του, περιμένουμε μέχρι να γίνει διαθέσιμος κάποιος ταμίας.
   Ανάλογα με την τυχαία παραγόμενη τιμή της πιθανότητας, γίνεται δεκτή ή απορρίπτεται η πληρωμή της κράτησης. Αν γίνει το τελευταίο, γίνεται αλλαγή των αντίστοιχων θέσεων από 1 σε 0, ώστε να φαίνονται πάλι διαθέσιμες.

5. int main(int argc, int* argv[]): Στην αρχή δημιουργούμε έναν πίνακα id, που περιέχει τα ids των νημάτων (πελάτες) ,αρχικοποιούμε τον πίνακα seats με 0 (όλες οι θέσεις διαθέσιμες), έπειτα δημιουργούμε τα νήματα, περνώντας τις κατάλληλες παραμέτρους και περιμένουμε μέχρι να τελειώσουν όλα τα νήματα μέσω του pthread_join. Τέλος, κάνουμε τις κατάλληλες εκτυπώσεις.

Όσον αφορά στις λειτουργίες του πολυνηματισμού, χρησιμοποιούμε τα lock-unlock για την αποφυγή παράλληλων συγκρούσεων στις εξής περιπτώσεις:

- theater: για την προσπέλαση του πίνακα seats και την αλλαγή των τιμών του.
- prints: για να μην μπλέκονται οι γραμμές μεταξύ τους
- sums: για να μην μπλέκεται ο υπολογισμός των αθροισμάτων για τις επιτυχείς συναλλαγές, ανεπιτυχείς πληρωμές και για τη μη εύρεση κατάλληλων/διαθέσιμων θέσεων.
- tels: για να δεσμευτεί και να αποδεσμευτεί ένας από τους τηλεφωνητές
- cashs: για να δεσμευτεί και να αποδεσμευτεί ένας από τους ταμίες

Τέλος, τα cond_wait περιμένουν από τα cond_signal το σήμα όταν γίνει διαθέσιμος κάποιος τηλεφωνητής ή ταμίας.

ΙΔΙΑΙΤΕΡΟΤΗΤΕΣ

Για την εκτέλεσή του προγράμματος, η εντολή στο virtual box είναι:

Η μετατροπή σε εκτελέσιμο με την εντολή:

chmod +x test-res.sh

και έπειτα η εκτέλεσή του, με:

./test-res.sh

Πριν αρχίσει η εκτέλεση χρειάζεται λίγος χρόνος για να εμφανιστεί το πρώτο print.

Γεωργία Πέτσα-ΑΜ: 3200155

Δημοσθένης Πλαβός- ΑΜ: 3200156