

# 1η Άσκηση νημάτων Λειτουργικά

Τζένη Μπολένα 3170117  
Κωνσταντίνος Νικολούτσος 3170122



`#include<stdio.h>`

## 1 Επεξήγηση προγράμματος

Η συγκεκριμένη άσκηση μας βοήθησε να κατανοήσουμε πως μπορούμε να μεταχειριστούμε και να συγχρονίσουμε νήματα (POSIX) χρησιμοποιώντας την βιβλιοθήκη `lpthread`.

Το πρόγραμμα μας εκανε συχνή χρήση `mutexes` (Mutual exclusion) το οποίο βοηθάει τους προγραμματιστές(δηλαδή εμάς) να συγχρονίζουμε τα νήματα. Πιο συγκεκριμένα τα `mutexes` χρησιμοποιούνται κατά κόρων σε κρίσιμες περιοχές η οποίες συνήθως έχουν να κάνουν με `shared memory`.

Εκτός αυτού έχει χρησιμοποιηθεί η εντολή `cond_wait` σε συνδιασμό με `broadcast` ή `signa`. Αυτό ουσιαστιά μας δίνει την δυνατότητα να κοιμήσουμε το νήμα του πελάτη ώστε να περιμένει έως ότου ικανοποιηθεί η συνθηκα του να έχουμε τουλάχιστον έναν διαθέσιμο τηλεφωνητή ο οποίος προκειται να εξυπηρετήσει τον πελάτη ώστε να βρεί θέσεις στο θέατρο.

## 2 Περιορισμοί στο πρόγραμμά μας

Το πρόγραμμα που υλοποιήσαμε ακολουθεί πιστά τις οδηγίες την εκφώνησης και επομένως δεν έχει έξτρα περιορισμούς.

Αν θέλετε μπορείτε να διαβάσετε τα comments στον κώδικα μας τα οποία είναι self-explanatory.

## 3 Δομή πηγαίου κώδικα

- Ξεκινάμε πέρνοντας απο τον χρήστη τον αριθμό πελατών και τον σπόρο. Κάνουμε τους ελέγχους οτι αυτά δώθηκαν σωστά, αλλιώς τερματίζει το πρόγραμμα.
- Κάνουμε κατάλληλες αρχικοποιήσεις σε μεταβλητές.
- Δημιουργούμε όλα τα νήματα(είναι όσα και οι πελάτες) με τη χρήση της εντολής **pthread\_create** και πλέον ξεκινάει η διαδικασία booking εισητηρίων.
- Για κάθε νήμα(πελάτη):
  1. Κράτα χρόνους για την αναμονή για το πότε ξεκίνησε, πότε συνδέθηκε με τηλεφωνητή και πότε τελείωσε.
  2. Κάνε lock τους τηλεφωνητές και αν υπρχει διαθέσιμος του τον δίνεις και τους μειώνεις κατα 1, αλλιώς με την εντολή **pthread\_cond\_wait** τον βάζεις να περιμένει μέχρι να βρεθεί διαθέσιμος τηλεφωνητής.Κάνε unlock τους τηλεφωνητές.
  3. Βρες πόσες θέσεις θέλει ο πελάτης(εδώ χρησιμοποιείται και ο σπόρος) και για να γίνει καλή προσομοίωση βάλε το thread σε sleep για random time.
  4. Κλείδωσε τις θέσεις, αν υπάρχουν διαθέσιμες δώστα στον πελάτη βάζοντας σε χ θέσεις το id του, αλλιώς επέστρεψε κάποιο result. Ξεκλείδωσε θέσεις.
  5. Αν το result μας λέει ότι μπορέσαμε και δώσαμε στον πελάτη όσες θέσεις ήθελε, τότε αν υπάρξει επιτυχή συνναλαγή κάνε lock στο balance και αύξησε το κέρδος της επιχείρησης και κάνε unlock. Αν η συνναλλαγή απέτυχε τότε ελευθέρωσε τις θέσεις που κρατήσαμε πριν για τον συγκεκριμένο πελάτη(lock, unlock στο mutex για τις θέσεις).

6. Έχουμε φτάσει σε σημείο όπου η κλίση του πελάτη ολοκληρώθηκε οπότε ελευθερώνουμε τον τηλεφωνητή.
  7. Στη διάρκεια του νήματος έχουμε ένα flag που μας λέει μέχρι που έφτασε ο πελάτης στη συνναλλαγή, οπότε ανλγως την τιμή του τυπώνουμε κατάλληλο μήνυμα για την πορεία της συνναλλαγής(lock/unlock το mutex για τύπωμα στην οθόνη).
  8. Υπολογίζουμε χρόνο αναμονής για να μιλήσει στον τηλεφωνητή και χρόνο εξυπηρέτησης και τα προσθέτουμε στους αντίστοιχους συνολικούς χρόνους(πάντα κάνουμε lock/unlock τα αντίστοιχα mutex ).
  9. Και τέλος κάνουμε έξοδο απο το νήμα αφού ο πελάτης εξυπηρετήθηκε.
- Στη main κάνουμε **pthread\_join** ώστε να περιμένουμε να τελειώσουν όλα τα νήματα(πελάτες).
  - Και τέλος τυπώνουμε το πλάνο του θεάτρου και ό,τι άλλο ζητείται απο την εκφώνηση.