

## ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

### Υποχρεωτικό Μάθημα 4<sup>ου</sup> εξαμήνου

Εαρινό Εξάμηνο 2021-2022

### Προγραμματιστική Εργασία

**Σκοπός:** Στην εργασία σας θα υλοποιήσετε ένα σύστημα αγοράς θέσεων σε θεατρικές παραστάσεις, με χρήση του πακέτου νημάτων POSIX threads (pthreads). Στο σύστημα αυτό, κάθε πελάτης αρχικά κλείνει τις θέσεις που επιθυμεί, στη συνέχεια τις πληρώνει με πιστωτική κάρτα και, τέλος, τα χρήματα μεταφέρονται στο λογαριασμό της εταιρείας. Στα συστήματα αυτά έχουμε έναν μεγάλο αριθμό πελατών οι οποίοι εξυπηρετούνται από έναν περιορισμένο αριθμό σημείων εξυπηρέτησης, συνεπώς το πρόγραμμά σας πρέπει να υλοποιεί αμοιβαίο αποκλεισμό (με mutexes) και συγχρονισμό (με μεταβλητές συνθήκης). Ο κώδικάς σας θα πρέπει να λειτουργεί σωστά στην εικονική μηχανή που διατίθεται στα CSLAB και στο διαδίκτυο (βλ. οδηγίες στο eclass). Η εργασία είναι ομαδική και είναι σχεδιασμένη για ομάδες δύο (2) ή τριών (3) ατόμων. Ο βαθμός της εργασίας θα είναι το 30% του τελικού βαθμού (δηλαδή, 3 μονάδες στις 10).

**Αντικείμενο:** Το σύστημα αγοράς θέσεων διαθέτει έναν **λογαριασμό στην τράπεζα**, ένα τηλεφωνικό κέντρο με  **$N_{tel}$  τηλεφωνητές** οι οποίοι εξυπηρετούν τους πελάτες (βρίσκουν θέσεις) και  **$N_{cash}$  ταμίες** οι οποίοι εκτελούν τις πληρωμές (χρεώνουν τις πιστωτικές κάρτες). Οι θέσεις του θεάτρου έχουν ορθογώνια διάταξη, με **κάθε σειρά** να έχει  **$N_{seat}$  θέσεις** και οι σειρές να χωρίζονται σε δύο ζώνες: οι **μπροστινές  $N_{zoneA}$  σειρές** κοστίζουν  **$C_{zoneA}$  ευρώ** ανά θέση και οι πίσω  **$N_{zoneB}$  σειρές** κοστίζουν  **$C_{zoneB}$  ευρώ** ανά θέση. Ο **πρώτος πελάτης** τηλεφωνεί τη **χρονική στιγμή 0**, και **κάθε επόμενος** πελάτης τηλεφωνεί μετά από ένα **τυχαίο ακέραιο πλήθος δευτερολέπτων** στο διάστημα  $[t_{reslow}, t_{reshigh}]$ . Όταν **όλοι οι τηλεφωνητές είναι απασχολημένοι**, ο πελάτης **περιμένει** τον επόμενο διαθέσιμο τηλεφωνητή. Όταν ένας πελάτης **συνδεθεί** με κάποιον **τηλεφωνητή**, επιλέγει **τυχαία μία ζώνη** με **πιθανότητα  $P_{zoneA}$**  για τις μπροστινές σειρές ( $1 - P_{zoneA}$ ) για τις πίσω σειρές, κι έναν **τυχαίο ακέραιο αριθμό εισιτηρίων** στο διάστημα  $[N_{seatlow}, N_{seathigh}]$ . Ο **τηλεφωνητής χρειάζεται** ένα **τυχαίο ακέραιο πλήθος δευτερολέπτων** στο **διάστημα  $[t_{seatlow}, t_{seathigh}]$**  για να **εξετάσει** αν υπάρχουν **αρκετές συνεχόμενες θέσεις** σε κάποια σειρά στη **συγκεκριμένη ζώνη**. Αν **δεν υπάρχουν** συνεχόμενες θέσεις, ο **πελάτης λαμβάνει** ένα μήνυμα **λάθους** και **ολοκληρώνει** την κλήση. Αν **υπάρχουν** συνεχόμενες θέσεις, **δεσμεύονται** στο **πλάνο του θεάτρου**, **υπολογίζεται το συνολικό τους κόστος**, και προχωράμε στην **πληρωμή με πιστωτική κάρτα**. Όταν **όλοι οι ταμίες είναι απασχολημένοι**, ο πελάτης **περιμένει** τον επόμενο διαθέσιμο ταμία. Όταν ο πελάτης **συνδεθεί** με τον ταμία, ο **ταμίας χρειάζεται** ένα **τυχαίο ακέραιο πλήθος δευτερολέπτων** στο διάστημα  $[t_{cashlow}, t_{cashhigh}]$  για να **δοκιμάσει να κάνει την πληρωμή**. Με **πιθανότητα  $P_{cardsuccess}$**  η πληρωμή γίνεται αποδεκτή, ο πελάτης **χρεώνεται** το κατάλληλο **κόστος**, τα **χρήματα μεταφέρονται** στο **λογαριασμό της εταιρείας** και ο πελάτης **ενημερώνεται** για το **συνολικό κόστος** και τις **θέσεις** του στην επιλεγμένη ζώνη. Αν η πληρωμή **αποτύχει**, οι θέσεις **επιστρέφονται** στο πλάνο του θεάτρου από τον ταμία κι ο **πελάτης λαμβάνει** ένα **μήνυμα λάθους** και ολοκληρώνει την κλήση. Όταν επεξεργαστεί  **$N_{cust}$  πελάτες**, το σύστημα **τυπώνει τα αποτελέσματά του**.

**Είσοδος και δεδομένα:** Το αρχικό ποσό στο λογαριασμό της εταιρείας είναι 0 ευρώ. Οι ακόλουθες σταθερές θα ορίζονται σε ένα αρχείο δηλώσεων:

- $N_{tel}=3$  τηλεφωνητές,  $N_{cash}=2$  ταμίες
- $N_{seat}=10$  θέσεις ανά σειρά
- $N_{zoneA}=10$  σειρές,  $N_{zoneB}=20$  σειρές
- $P_{zoneA}=30\%$

- $C_{zoneA}=30$  ευρώ,  $C_{zoneB}=20$  ευρώ
- $N_{seatlow}=1$  θέση,  $N_{seathigh}=5$  θέσεις
- $t_{reslow}=1$  sec,  $t_{reshigh}=5$  sec
- $t_{seatlow}=5$  sec,  $t_{seathigh}=13$  sec
- $t_{cashlow}=4$  sec,  $t_{cashhigh}=8$  sec
- $P_{cardsucces}=90\%$

Το πρόγραμμά σας θα δέχεται δύο (ακριβώς) παραμέτρους από τη γραμμή εντολών: το πλήθος των πελατών προς εξυπηρέτηση,  $N_{cust}$ , και τον τυχαίο σπόρο για τη γεννήτρια των τυχαίων αριθμών.

**Έξοδος εργασίας:** Για **κάθε πελάτη** θα **τυπώνεται** στην οθόνη **ένα από τα παρακάτω μηνύματα**, ανάλογα με το πώς κατέληξε η κράτησή του, το οποίο θα ξεκινά με τον αριθμό του πελάτη:

- Η **κράτηση ολοκληρώθηκε επιτυχώς**. Οι θέσεις σας είναι στη ζώνη <a>, σειρά <b>, αριθμός <c>, d, ...> και το κόστος της συναλλαγής είναι <X> ευρώ.
- Η **κράτηση απέτυχε** γιατί **δεν υπάρχουν κατάλληλες θέσεις**.
- Η **κράτηση απέτυχε** γιατί η **συναλλαγή με την πιστωτική κάρτα δεν έγινε αποδεκτή**.

Η σειρά των γραμμών θα είναι τυχαία, αλλά **οι γραμμές δεν πρέπει να μπλέκονται μεταξύ τους**. Στο τέλος της εκτέλεσης, το σύστημα θα τυπώνει τα ακόλουθα:

- Το **πλάνο των θέσεων**, π.χ. Ζώνη A / Σειρά 1 / Θέση 1 / Πελάτης 3, Ζώνη 2 / Σειρά 5 / Θέση 2 / Πελάτης 4, κ.λπ.
- Τα **συνολικά έσοδα** από τις πωλήσεις.
- Το **ποσοστό των συναλλαγών** που **ολοκληρώνεται** με **κάθε έναν** από τους παραπάνω τρεις τρόπους.
- Το **μέσο χρόνο αναμονής** των **πελατών** (από τη στιγμή που εμφανίζεται ο πελάτης μέχρι να μιλήσει με τον τηλεφωνητή και, αν η συναλλαγή προχωρήσει σε πληρωμή, από τη στιγμή που τελειώνει ο τηλεφωνητής μέχρι τη στιγμή που αναλαμβάνει ο ταμίας).
- Το **μέσο χρόνο εξυπηρέτησης** των πελατών (από τη στιγμή που εμφανίζεται ο πελάτης, μέχρι να ολοκληρωθεί ή να αποτύχει η κράτηση).

**Δομή κώδικα:** Το **αρχικό νήμα** του προγράμματός σας θα **δημιουργεί ένα νήμα** ανά **πελάτη** (συνολικά  $N_{cust}$  νήματα) στα οποία θα **περνάτε έναν αριθμό νήματος** (από 1 έως  $N_{cust}$ ) ώστε να τα **διακρίνετε** – αυτός είναι ο **αριθμός πελάτη**. **Κάθε νήμα** στη συνέχεια θα εκτελεί τα παραπάνω βήματα μέχρι να **ολοκληρωθεί η παραγγελία** του και θα **τυπώνει την κατάλληλη έξοδο**. Τέλος, το **αρχικό νήμα** θα **τυπώνει την τελική έξοδο**. Θα χρειαστείτε τουλάχιστον τα ακόλουθα:

- Μία **ακέραιη μεταβλητή** και ένα **mutex** για να **μετράτε το πλήθος των διαθέσιμων τηλεφωνητών** και μία **μεταβλητή συνθήκης** για να **συγχρονίσετε τους τηλεφωνητές με τους πελάτες**, έτσι ώστε όταν **δεν υπάρχουν διαθέσιμοι** τηλεφωνητές να **μπλοκάρονται οι πελάτες**. Με ανάλογο τρόπο θα πρέπει να χειριστείτε τους ταμίες.
- Έναν **πίνακα με το πλάνο των θέσεων** και τουλάχιστον ένα **mutex** για να τον **κλειδώνετε**. Προσοχή στο ότι οι **θέσεις** κάθε κράτησης πρέπει να είναι **συνεχόμενες και στην ίδια σειρά**. **Σκεφτείτε** πώς μπορείτε να **επιταχύνετε τη διαδικασία**.
- **Μεταβλητές** και τα **σχετικά mutex** για τον **τραπεζικό λογαριασμό**, τα **στατιστικά** και για το **κλείδωμα της οθόνης** όταν **τυπώνετε την έξοδο**.

Προσοχή στο να τερματίζετε σωστά τα νήματα, να τα περιμένετε όπου χρειάζεται και (κυρίως!) να απελευθερώνετε σωστά όση μνήμη δεσμεύετε.

**Υποδείξεις:**

- Κατά τη μεταγλώττιση πρέπει να δίνετε την επιλογή `-pthread` για να χρησιμοποιηθεί η βιβλιοθήκη POSIX threads.
- Για να προσομοιώσετε το χρόνο που διαρκεί η αναζήτηση θέσεων θα χρησιμοποιήσετε την `unsigned int sleep(unsigned int seconds)`.

- Για να παράγετε μία σειρά ψευδοτυχαίων αριθμών, θα χρησιμοποιήσετε την `int rand_r(unsigned int *seedp)`. Χρησιμοποιήστε τον τελεστή `%` για να περιορίσετε το εύρος τιμών των τυχαίων αριθμών.
- Για τον υπολογισμό του χρόνου αναμονής και εξυπηρέτησης χρησιμοποιείτε την `int clock_gettime(clockid_t clk_id, const struct timespec *tp)` στην αρχή και τέλος κάθε πράξης, με πρώτη παράμετρο τη σταθερά `CLOCK_REALTIME`.
- Χρησιμοποιήστε έναν βρόχο με `while` για να ελέγχετε τη συνθήκη αναμονής και να κάνετε `pthread_cond_wait` για όσες φορές χρειάζεται.
- Προσοχή στο ότι οι χρόνοι αναμονής μπορεί να περιλαμβάνουν δύο στάδια αναμονής.
- Προσοχή στο ότι το πλάνο του θεάτρου εκτός από τους τηλεφωνητές μπορεί να το ενημερώνουν και οι ταμίες (όταν αποτύχει η πληρωμή, οι θέσεις επιστρέφονται).

**Παραδοτέα:** Ο κώδικάς σας πρέπει να αποτελείται από ένα αρχείο με δηλώσεις (συμπεριλαμβανομένων των σταθερών) και ένα αρχείο κώδικα C για το πρόγραμμα. Τα αρχεία αυτά πρέπει να έχουν ονόματα της μορφής `p3x-p3y-p3z-res.h` για τις δηλώσεις, `p3x-p3y-p3z-res.c` για τον κώδικα C, όπου και `p3x-p3y-p3z` είναι οι αριθμοί μητρώου σας. Εκτός από τον κώδικα, θα πρέπει να γράψετε μία αναφορά η οποία να περιγράφει τη δομή του κώδικά σας και να αναφέρει τυχόν περιορισμούς ή πρόσθετα χαρακτηριστικά που έχετε υλοποιήσει. Η αναφορά πρέπει να είναι ένα αρχείο σε μορφή PDF με όνομα της μορφής `p3x-p3y-p3z-res.pdf`. Τέλος, θα πρέπει να συμπεριλάβετε ένα αρχείο με όνομα `test-res.sh` το οποίο θα μεταγλωττίζει και θα εκτελεί το πρόγραμμά σας με παραμέτρους 100 πελάτες και αρχικό σπόρο 1000. Αυτά τα τέσσερα αρχεία (και τίποτα άλλο) θα πρέπει να συμπιεστούν σε ένα αρχείο σε μορφή 7zip με όνομα της μορφής `p3x-p3y-p3z-res.7z` και να υποβληθούν από ένα μόνο μέλος της ομάδας μέσω της υποβολής εργασιών του eclass.

**Προθεσμία υποβολής:** Τα συμπιεσμένα αρχεία με τις εργασίες σας θα πρέπει να παραδοθούν μέσω του eclass μέχρι την Δευτέρα ~~30/5/2022~~ 6/6/2022 και ώρα 23:59.

**Βαθμολόγηση και εξέταση:** Αρχεία που δεν θα έχουν την *ακριβή* ονοματολογία που ζητείται παραπάνω *θα μηδενιστούν*. Εργασίες με ομοιότητες που υποδεικνύουν αντιγραφή *θα μηδενιστούν όλες* (θα γίνει έλεγχος με ειδικό πρόγραμμα). Δειγματοληπτικά, θα κληθούν ορισμένες ομάδες για προφορική εξέταση στα CSLAB. Μέλη ομάδων, ή και ολόκληρες ομάδες, που δεν θα προσέλθουν στην προφορική εξέταση, *θα μηδενιστούν*.