

Εργασία Λειτουργικά Συστήματα

Μέλη ομάδας & Α.Μ

Ηλίας Μπεζαντάκος (3160241)

Αριστοτέλης Μικρώνης (3190122)

Κομνάς Καφάσης (3190081)

Δομή κώδικα:

Πρώτιστα, δηλώνουμε μέσα σε ένα αρχείο .h (header) όλες τις σταθερές οι οποίες ζητούνται μέσω της εκφώνησης . Έπειτα, δηλώνουμε τις μεταβλητές τις οποίες θα χρειαστούμε για το κύριο πρόγραμμα μετά όπως και τα νήματα. Εν τέλει, ορίζουμε και τις συναρτήσεις οι οποίες θα εμφανίζουν τα κατάλληλα μηνύματα εξόδου για τα ζητούμενα της εργασίας. Μάλιστα, στο κυρίως πρόγραμμα περιλαμβάνουμε αυτό το αρχείο μέσα στον κώδικα (#include "p3160241-p3190122-pizza.h")

Όταν καλείται το αρχείο για την εκτέλεση, στο πρόγραμμα πρέπει να περαστούν ο αριθμός των πελατών οι οποίοι πρέπει να εξυπηρετηθούν, καθώς και ο σπόρος για τον generator τυχαίων αριθμών . Ύστερα, ελέγχεται αν έχουν εισαχθεί οι παράμετροι και μαζί και τα δεδομένα. Στη συνέχεια, γίνονται initialize (αρχικοποίηση) οι μεταβλητές οι οποίες θα κάνουν χρήση των νημάτων , προκειμένου να αλλάζουν κάθε στιγμή κατάλληλα τις τιμές των πόρων για την υλοποίηση της παραγγελίας, καθώς και τους ζητούμενους χρόνους.

Ένας πίνακας order_id αρχικοποιείται προκειμένου να κρατάει τα id της κάθε παραγγελίας , ενώ με τη χρήση της malloc κατασκευάζουμε πίνακα για τα threads. Αυτοί οι πίνακες λοιπόν, καθώς και η συνάρτηση order , είναι οι παράμετροι που περνάμε στην pthread_create ώστε να φτιάξουμε νήμα για κάθε πελάτη. Επιπρόσθετα, με τη χρησιμοποίηση των sleep και rand_r, υλοποιούμε την αναμονή ενός τυχαίου διαστήματος wait έως ότου έρθει επόμενη παραγγελία.

Έπειτα,καλούμε την `pthread_join` για κάθε νήμα για να είμαστε βέβαιοι ότι το κυρίως(main) πρόγραμμα δεν θα τερματίσει προτού τελειώσουν όλα τα νήματα τις εργασίες τους και τυπώνουμε τα συνολικά έσοδα από τις πωλήσεις, το πλήθος επιτυχημένων και αποτυχημένων παραγγελιών, το μέσο και το μέγιστο χρόνο αναμονής, εξυπηρέτησης και κρυώματος των (επιτυχημένων) παραγγελιών (όπου μέσος χρόνος είναι το άθροισμα όλων των χρόνων προς τους συνολικούς πελάτες και ο μέγιστος χρόνος μέσω της `order`, όπου κάθε φορά ελέγχεται αν κάποιος καινούριος χρόνος ολοκλήρωσης υπερβαίνει τον τρέχοντα μέγιστο, για κάθε χρόνο από τους παραπάνω που περιγράφηκαν).

Κάθε νήμα λοιπόν στην `order` πραγματοποιεί :

- Αποθηκεύει πάντα τον χρόνο όπου εμφανίζεται ο πελάτης,το χρόνο που σηκώνει το τηλέφωνο ο τηλεφωνητής , το χρόνου που ξεκινάει να προετοιμάζεται η παραγγελία, το χρόνο που μπαίνει στο φούρνο, το χρόνο που βγαίνει από το φούρνο και ξεκινάει το πακετάρισμα , το χρόνο που τελειώνει το πακετάρισμα και εν τέλει το χρόνο στον οποίο παραδίδεται στον πελάτη.
- Με `pthread_mutex_lock` το νήμα “κλειδώνει” ώστε να βρει διαθέσιμο τηλεφωνητή. Αν δεν βρει, με την `pthread_cond_wait` περιμένει μέχρι κάποιος να “ελευθερωθεί” (δηλαδή ο αριθμός των διαθέσιμων τηλεφωνητών να γίνει θετικός),όπου και μειώνει τον αριθμό διαθέσιμων τηλεφωνητών κατά 1, με `pthread_mutex_unlock` “ξεκλειδώνει” το `mutex` για τους τηλεφωνητές , ενώ έπειτα συνεχίζει.
- Στη συνέχεια , κάνει `mutex lock` και `unlock` τα `mutexes` του `total wait time` και `max wait time` , προκειμένου να ανανεώσει αυτούς τους χρόνους.
- Υπολογίζεται με την `rand_r` ένας τυχαίος αριθμός από πίτσες για την τρέχουσα παραγγελία και με την `sleep` προσομοιώνεται ο χρόνος που πρέπει να περάσει για να γίνει μέσω του τηλεφωνητή η πληρωμή της παραγγελίας. Αν η πληρωμή αποτύχει , με `lock` και `unlock` στο `mutex` των `failed orders` ανανεώνουμε αυτή την τιμή, και με `mutex lock` κα `unlock` στην οθόνη, τυπώνεται το μήνυμα αποτυχίας. Ο τηλεφωνητής απελευθερώνεται και με `pthread_exit` τερματίζουμε το νήμα. Αν η πληρωμή είναι επιτυχής, ανανεώνουμε ομοίως το `successful orders` και πάλι ομοίως τυπώνουμε αντίστοιχα το μήνυμα επιτυχίας. Έπειτα με `mutex lock` και `unlock` στο `total income` το ανανεωνούμε. Ο τηλεφωνητής τότε απελευθερώνεται.

- Με `pthread_mutex_lock/unlock` και `pthread_cond_wait` ακολουθεί μια παρόμοια διαδικασία με τους τηλεφωνητές, τώρα για τους ψήστες (κλείδωμα-ξεκλείδωμα, αναμονή για διαθέσιμο και μείωση κατά 1).
- Το νήμα περιμένει μέσω της `sleep` για το χρόνο προετοιμασίας της πίτσας.
- Με την ίδια διαδικασία (κλείδωμα-ξεκλείδωμα, αναμονή για διαθέσιμο και μείωση κατά 1), ο χρήστης περιμένει έως ότου γίνουν αρκετοί φούρνοι διαθέσιμοι, ώστε όλες οι πίτσες να ψηθούν παράλληλα, γιατί κάθε φούρνος χωράει μία μόνο πίτσα. Όταν οι κατάλληλοι φούρνοι είναι διαθέσιμοι, κάθε πίτσα μπαίνει σε έναν φούρνο και ο ψήστης απελευθερώνεται για να αναλάβει την επόμενη παραγγελία.
- Το νήμα περιμένει μέσω της `sleep` για το χρόνο ψησίματος των πιτσών της παραγγελίας.
- Τώρα που οι πίτσες είναι έτοιμες, ο φούρνος περιμένει να βρεθεί και να δεσμευθεί ο (μοναδικός) υπάλληλος πακεταρίσματος , ώστε ο πρώτος να ελευθερωθεί (με `pthread_mutex_lock/unlock` και `pthread_cond_wait` ακολουθεί μια παρόμοια διαδικασία για τον υπάλληλο πακεταρίσματος).
- Με κλείδωμα αυξάνουμε την μεταβλητή των διαθέσιμων φούρνων ανάλογα και καλούμε την `pthread_cond_signal` για να “ξεμπλοκάρει” ένα από τα νήματα που είναι πιθανό να περιμένουν για διαθέσιμο φούρνο και τέλος ξεκλειδώνουμε. Ακολουθεί η προσομοίωση του πακεταρίσματος. Έπειτα, υπολογίζεται ο χρόνος ετοιμασίας της πίτσας (το διάστημα από την εμφάνιση του πελάτη μέχρι και το πακετάρισμα της παραγγελίας) και με `lock` και `unlock` στο `mutex` της οθόνης, τυπώνεται το μήνυμα της ετοιμασίας. Ο υπάλληλος ελευθερώνεται με το γνωστό τρόπο και αναμένουμε μέχρι να βρεθεί και να δεσμευθεί διανομέας πάλι με το γνωστό τρόπο.
- Ακολουθεί η προσομοίωση της διανομής: με `rand_r` βρίσκουμε το συνολικό χρόνο διανομής και κάνουμε `sleep` .Υπολογίζουμε το συνολικό χρόνο παράδοσης της παραγγελίας και με `mutex lock` και `unlock` στην οθόνη τυπώνουμε το μήνυμα της παράδοσης. Κάνουμε πάλι `sleep` για το χρόνο διανομής ο οποίος απαιτείται ώστε να επιστρέψει ο διανομέας πίσω στην πιτσαρία. Εν συνεχεία, ο διανομέας απελευθερώνεται .
- Έπειτα, κάνοντας `lock` και `unlock` στα αντίστοιχα `mutexes` ανανεώνουμε το συνολικό και μέγιστο χρόνο εξυπηρέτησης και το συνολικό και μέγιστο χρόνο κρυώματος.
- Το νήμα έχει τελειώσει τις εργασίες του και κάνει `pthread_exit` με το αντίστοιχο `order_id` του.

ΠΡΟΣΘΕΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Ακολουθήσαμε την εκφώνηση ακριβώς όπως μας δόθηκε για την δημιουργία του προγράμματος, απλά έχουμε προσθέσει μηνύματα που βοηθούν στην προσομοίωση της διαδικασίας. Σχετικά με τον σπόρο, έχουμε την μεταβλητή `seed` που είναι καθολική μεταβλητή και προσθέτωντάς την με το `order id` της κάθε παραγγελίας παίρνουμε την μεταβλητή `seedr` που είναι μοναδική για κάθε παραγγελία για να μην είναι ίδιοι οι ψευδοτυχαίοι αριθμοί με αυτούς των άλλων παραγγελιών