

2η εργασία Λειτουργικά Συστήματα - Προσομοίωση λειτουργίας πιτσαρίας με ντελιβερι

Η εργασία αποτελείται από 2 αρχεία, το αρχείο κεφαλίδας και εκείνο του κώδικα σε C. Στο header αρχείο έχουν γίνει οι δηλώσεις σταθερών και μεθόδων που χρησιμοποιούνται στο .c αρχείο. Το 2ο αρχείο είναι και εκείνο που παράγει λειτουργικό κώδικα και αποτελείται από 2 μεθόδους, την main και την order.

Στη **main** γίνεται αρχικοποίηση των global mutexes που έχουν δηλωθεί στην αρχή του file, η δημιουργία των threads, το join τους καθώς και η καταστροφή τους αμέσως μετά το πέρας της λειτουργίας τους. Για να διατηρηθούν τα id των threads έχουν χρησιμοποιηθεί 2 πίνακες μεγέθους ίσου με τον αριθμό των πελατών, τα οποία και περνούν σαν ορίσματα στην pthread_create, ενώ η ανάγκη του join να αναφερθεί σε συγκεκριμένα threads προκειμένου να κάνει την δουλειά του έχει καλυφθεί από τον πίνακα των threads: threads_t (table). Τέλος ολοκληρώνεται η main με την εκτύπωση των στατιστικών που προκύπτουν με κατάλληλες πράξεις.

Ο σκοπός της **order** είναι να πραγματοποιήσει όλη την υπόλοιπη διαδικασία που αναφέρεται στην εκφώνηση της εργασίας, δηλαδή την προσομοίωση της λειτουργίας της πιτσαρίας από τη στιγμή που φτάνει μια παραγγελία μέχρι την ολοκλήρωση της (για όλες τις παραγγελίες) - φυσικά κάθε thread που δημιουργείται αποτελεί μια καινούρια κλήση της order - . Αρχικά λαμβανεται ένα στιγμιότυπο χρόνου για τον υπολογισμό του χρόνου έναρξης της παραγγελίας. Το κάθε thread ξεκινά τη διαδικασία λειτουργίας του αλλάζοντας την τιμή του seed που θα χρησιμοποιήσει στο κορμό του, το οποίο seed έχει γίνει τοπικό για να αποφευχθεί η επανάληψη της ίδιας σειράς ψευδοτυχαίων αριθμών. Επειτα ξεκινά μια επαναληπτική διαδικασία αναμονής μέχρις ότου να βρεθεί ένας διαθέσιμος μάγειρας - το wait γίνεται σε βρόχο επανάληψης για να αποφευχθούν πιθανότητες spurious wake-up που μπορεί να αντιμετωπίζαμε αν είχε γίνει μέσα σε if -. Στη συνέχεια και αφού έχει βρεθεί διαθέσιμος μάγειρας, το thread τον δεσμεύει. Χρησιμοποιείται η γεννήτρια τυχαίων αριθμών με σπόρο τοπικό για να μας δώσει το πλήθος από πίτσες που επιθυμεί ο πελάτης και γίνεται ένα sleep αντίστοιχης διάρκειας μέχρι να ετοιμαστούν. Εμφωλευμένα αρχίζει και η υποδιαδικασία που προσομοιώνει την αναμονή εύρεσης διαθέσιμου φούρνου, η οποία πραγματοποιείται με τον ίδιο τρόπο όπως με την αναμονή εύρεσης διαθέσιμου μάγειρα. Ο φούρνος δεσμεύεται, ο ψηστής απελευθερώνεται και καλείται η sleep για να προσομοιώσει την διαδικασία του ψησίματος. Τότε κρατείται ένα στιγμιότυπο χρόνου το οποίο χρησιμεύει στον υπολογισμό του χρόνου κρυψώματος της παραγγελίας. Στη συνέχεια ο φούρνος χωρίς να αποδεσμευτεί περιμένει να γίνει διαθέσιμος ένας διανομέας για να αφαιρέσει τις πίτσες από αυτόν. Ο διανομέας γίνεται διαθέσιμος και δεσμεύεται αμέσως με την αποδεσμευση του φούρνου να ακολουθεί. Γίνεται διπλό sleep για να προσομοιωθεί το χρονικό διάστημα αφίξης και επιστροφής του διανομέα και ο διανομέας αποδεσμεύεται. Τέλος λαμβανεται ένα τελευταίο στιγμιότυπο χρόνου που σημαίνει την ολοκλήρωση της τρεχουσας παραγγελίας και χρησιμοποιώντας καταλλήλους υπολογισμούς, ανανεώνονται οι συνολικοί χρόνοι (service, cooling).

Η διαδικασία χρήσης της μεθόδου **rand_r(&seed)** περιέχει μια ιδιαιτερότητα που χρήζει επεξήγησης. Ο σπόρος αν ήταν global και ίσος με την τιμή του δωθέντος σπόρου θα δημιουργούσε επαναλαμβανόμενες σειρές ψευδοτυχαίων αριθμών για τα νήματα, μιας και ένας δωθέν σπόρος βγάζει τα ίδια αποτελέσματα κάθε φορά που καλείται. Για το λόγο αυτό στην εργασία έχει υλοποιηθεί μια ιδέα αλλαγής της τιμής του σπόρου για κάθε thread. Οι μεταβλητές temp και temp_o (order) χρησιμοποιούνται ως τοπικές μεταβλητές και αποτελούν την τιμή του σπόρου για κάθε κλήση της rand_r στη main καθώς και για κάθε thread.

Επίσης πρέπει να σημειωθεί πως οποιαδήποτε αλλαγή γίνεται σε global μεταβλητή, καλείται lock-unlock για να μην υπάρξει μπλέξιμο μεταξύ των threads.

Στη νέα έκδοση της εργασίας έχουν αλλάξει ορισμένοι λανθασμένοι τυποί υπολογισμού ορίων που χρησιμοποιούσαν τη rand και έχουν προστεθεί mutexes για τους διανομείς και τους χρονούς κρυψώματος. Πρακτικά η δομή και η λειτουργία της 1ης και της 2ης εργασίας είναι ίδιες.

Επιπλέον αναφορές για κάθε κομμάτι κώδικα θα βρείτε στα σχόλια του .c αρχείου.