

Το πρόγραμμα ξεκινάει ελέγχοντας αν υπάρχουν ακριβώς 3 arguments. Στη συνέχεια, δημιουργούνται όσα threads όσα το δεύτερο argument σε ένα for loop και σε ένα δεύτερο γίνονται join. Ολόκληρη την παραγγελία την χειρίζεται μία συνάρτηση, η handle_clients. Αρχικά, εξακριβώνουμε ότι η πρώτη παραγγελία φτάνει την στιγμή 0 και οι υπόλοιπες περιμένουν. Μετά, οι τηλεφωνητές, οι φούρνοι, οι διανομείς και τα λοιπά είναι μεταβλητές που μειώνονται κάθε φορά που ένα thread χρειάζεται. Για να γίνει αυτό βάζουμε ένα while loop που ελέγχει αν υπάρχουν διαθέσιμοι πόροι. Σε περίπτωση που δεν υπάρχουν οι πόροι τότε καλείται η pthread_cond_wait και περιμένει μέχρι να αυξηθεί η αντίστοιχη μεταβλητή. Αν υπάρχουν οι πόροι, τότε το thread δεν σταματάει καθόλου στο while loop και απλά συνεχίζει μειώνοντας την αντίστοιχη μεταβλητή. Αυτό γίνεται κάθε φορά που πρέπει να περιμένει κάποιο thread για να πάρει πόρους. Για να απελευθερωθούν οι πόροι και τα thread που περιμένουν να συνεχίσουν, όταν κάποιο thread τελειώσει την λειτουργία που ήθελε, αυξάνει την αντίστοιχη μεταβλητή-πόρο και καλεί την συνάρτηση pthread_cond_signal η οποία κάνει την συνάρτηση pthread_cond_wait να ξαναελέγξει την συνθήκη του while loop και έτσι συνεχίζει το thread παίρνοντας πόρους. Επίσης κάθε φορά που ένα thread θέλει να αυξομειώσει global μεταβλητές κλειδώνουμε προσωρινά τα δεδομένα με την συνάρτηση pthread_mutex_lock ώστε να μπορεί μόνο ένα νήμα την φορά να έχει πρόσβαση σε αυτά. Όταν τελειώνει τους υπολογισμούς καλούμε την συνάρτηση pthread_mutex_unlock ξεκλειδώνοντας τα δεδομένα. Τέλος, αφού τελειώσουν τα threads εμφανίζουμε στην οθόνη στατιστικά όπως total earnings, successful_orders, avg_wait_time και λοιπά.

Επιπλέον, δεν χρησιμοποιήσαμε την `rand_r` ούτε την `clock_gettime` οπότε υλοποιήσαμε τις `rand_num` για το `rng` και για να βρούμε την ημερομηνία την `get_time` ενώ την χρονομέτρηση την κάναμε ως εξής: Πριν μια χρονική στιγμή θέταμε μία μεταβλητή πχ `time_before = time(NULL)` και μετά την χρονική στιγμή κάναμε το ίδιο `time_after = time(NULL)` και μετά κάναμε μια αφαίρεση.