

Έχουμε δημιουργήσει 3 αρχεία. Ένα σκριπτάκι που μεταγλωττίζει και τρέχει το πρόγραμμα (.sh) , ένα αρχείο δηλώσεων των σταθερών που χρησιμοποιούνται (.h) και τέλος το κυρίως πρόγραμμα (.c). Το header file περιέχει τις δηλώσεις των σταθερών του προγράμματος , με τις τιμές που ορίζουν οι οδηγίες της εργασίας. Το κυρίως πρόγραμμα είναι δομημένο ως εξής:

Έχουμε ορίσει κάποιες global μεταβλητές για να αποθηκεύσουμε πληροφορίες στις οποίες θα έχουν πρόσβαση όλα τα νήματα. Όταν ξεκινάει το main νήμα του προγράμματος γίνεται ένας έλεγχος για τον αριθμό και τις τιμές των παραμέτρων. Οι τιμές των παραμέτρων του προγράμματος αναθέτονται σε μεταβλητές της main. Στη συνέχεια δημιουργούμε έναν πίνακα από νήματα. Για το κάθε νήμα δημιουργούμε ένα id. Βάζουμε το πρόγραμμα να "κοιμηθεί" για τυχαίο χρόνο με εύρος που ορίζεται από τις σταθερές μας , για να προσομοιώσουμε τον τυχαίο χρόνο μέχρι να τηλεφωνήσει ο επόμενος πελάτης. Στη συνέχεια δημιουργούνται τα νήματα και πάνε στη συνάρτηση pizzeria όπου ξεκινάει η ρουτίνα του "μαγαζιού". Το κάθε νήμα διαφοροποιεί τον σπόρο έτσι ώστε να μην έχουν όλα τα νήματα τον ίδιο σπόρο και συνεπώς την ίδια ακολουθία. Κάθε νήμα περνάει από όλα τα στάδια της ρουτίνας (εκτός και αν αποτύχει η πληρωμή όπου το νήμα επιστρέφει στη main). Έχουμε δημιουργήσει και έχουμε κάνει initialize mutexes για "το μπλοκάρισμα" σημείων του κώδικα , έτσι ώστε να έχει πρόσβαση μόνο ένα νήμα τη φορά. Με while loops και mutex conditions σταματάμε τα νήματα όταν οι συνθήκες της ρουτίνας δεν επιτρέπουν το επόμενο νήμα να συνεχίσει την ρουτίνα του. Όλα τα mutex και τα conditions καταστρέφονται στο τέλος της main αφού το κάθε νήμα έχει ολοκληρώσει τη ρουτίνα του. Μόλις τα νήματα επιστρέφουν από την ρουτίνα τα κάνουμε join ώστε να μην τερματίσουν το πρόγραμμα αλλά να περιμένουν να τελειώσουν και τα υπόλοιπα. Από τη ρουτίνα δεν επιστρέφουν καμία τιμή , ότι ενημέρωση έχουμε από τα νήματα για στατιστικά στοιχεία γίνεται μέσω global μεταβλητών που στο τέλος μας δίνουν τα στοιχεία που θέλουμε να τυπώσουμε. Η μέτρηση του χρόνου γίνεται με τη βοήθεια της συνάρτησης clock_gettime() με τη οποία ορίζουμε χρονικές στιγμές μέσα στη ρουτίνα και με τις κατάλληλες πράξεις παίρνουμε τα χρονικά διαστήματα που μας ενδιαφέρουν.