

README

*Στα παραδοτέα υπάρχουν **δύο** αρχεία κώδικα. Το πρώτο είναι το **p3160230-p3160265-pizzeria.h** το οποίο περιλαμβάνει:

- 1) Τις δηλώσεις των **σταθερών** που περιγράφονται στην εκφώνηση της εργασίας όπως :
N_COOK, N_PACKER, N_ORDERLOW, P_FAIL, T_PREP κλπ.
- 2) Τις δηλώσεις των **mutexes** και των **condition variables**.
- 3) Τις δηλώσεις των **συναρτήσεων**
- 4) Δηλώσεις **global** μεταβλητών που χρησιμοποιούνται για τα **στατιστικά** του προγράμματος.

*Το δεύτερο αρχείο είναι το **p3160230-p3160265-pizzeria.c** στο οποίο υπάρχει, η συνάρτηση **main** απο την οποία ξεκινάει η εκτέλεση του προγράμματος. Δημιουργεί νήματα σε αριθμό ίσο με τον αριθμό των πελατών ,περιμένει μεχρι να τελειώσει η εκτέλεση τους και τέλος, τα καταστρέφει ώστε να μην παρακρατούν πόρους από το λειτουργικό σύστημα.

*Η εργασία υλοποιήθηκε σε υπολογιστή με λειτουργικό **Ubuntu 20.04.1 LTS** .Ο κώδικας μεταγλωττίζεται και εκτελείται χωρίς λάθη. Χρησιμοποιήθηκαν όλες οι υποδείξεις της εκφώνησης. Έχει γίνει ο καταλληλος σχολιασμος του κώδικα και στα δυο αρχεία.

*Στην εργασία αποφασίσαμε να χρησιμοποιήσουμε συνολικά **6 mutexes** και **4 condition variables**. Για καθε πόρο στην γραμμή παραγωγής της πίτσας(**cook,oven,packer,deliverer**) χρησιμοποιούμε ενα **mutex** και ενα **condition variable** (πχ για τον cook έχουμε cookMutex και cookCondVar). Επιπλεον ενα mutex , το **printMutex** χρησιμοποιείται για τις εκτυπώσεις στην οθόνη(Π.χ. αν μια παραγγελια ειναι επιτυχής ή όχι). Ενα άλλο mutex το **statsMutex** χρησιμοποιειται οταν αλλάζουμε τιμή στα στατιστικά της εργασίας(global μεταβλητές). Π.χ: Όταν υπολογίζουμε το **totalIncome**.

*Η αρχικοποίηση και η καταστροφή των mutexes και condition variables γίνεται αντίστοιχα από τις συναρτήσεις: **init_mutexes_conditions()** και **destroy_mutexes_conditions()**. Ενώ η εκτύπωση των στατιστικών μέσω της συνάρτησης **print_stats()**.

*Η συνάρτηση **pizzeria()** είναι η συνάρτηση την οποία τρέχει το κάθε νήμα και προσομοιώνει την λειτουργία μιας πιτσερίας. Τυπώνονται τα καταλληλα μηνύματα στην οθονη και γίνεται **pthread_exit()** οταν μια παραγγελία είναι αποτυχημένη ή αφού έχει παραδωθεί στον πελάτη.

*Για την μεταγλώτιση και την εκτελεση του προγραμματος τρεχουμε το script, **test-res.sh** το οποίο θα μεταγλωττίζει και θα τρέξει το προγραμμα με παραμέτρους: **100** πελάτες και αρχικό seed **1000**.

*Όταν τελειώσει η προσομοίωση της λειτουργίας της πιτσερίας **εκτυπώνονται** όλα τα στατιστικά που ζητούνται στη εκφώνηση.

*Έχουμε προσθέσει **δυο** επιπλέον global μεταβλητές, τις **totalServingTime** και **totalCoolTime** οι οποίες κρατάνε την τιμή του αθροισματος όλων των χρόνων εξυπηρέτησης των πελατών και το άθροισμα των χρόνων κρυώματος των παραγγελιών αντίστοιχα. Οι παραπάνω μεταβλητές χρησιμοποιούνται στο να βρεθούν οι μεσοι χρόνοι.