

## 2<sup>η</sup> Προγραμματιστική Εργασία

Δημήτριος Βαρδακαστάνης - 3160011,  
Στέφανος Μιχάλας - 3160098

---

### Περιγραφή

Στην εργασία μας τροποποιήσαμε το πρόγραμμα της πρώτης που προσομοιώνει ένα σύστημα παραγγελιών και παρασκευής πίτσας με χρήση του πακέτου νημάτων POSIX threads (pthreads). Χρησιμοποιώντας νήματα, mutexes και conditions προσθέσαμε διανομείς που παραδίδουν τις παραγγελίες στους πελάτες, με τέτοιο τρόπο ώστε να ελευθερώνεται ο παρασκευαστής μόλις ετοιμάσει τις πίτσες, να ελευθερώνεται ο φούρνος μόνο αν κάποιος διανομέας πάρει τις πίτσες από το φούρνο για να τις παραδώσει και αντίστοιχα αυτός να ελευθερώνεται όταν παραδώσει και επιστρέψει στο μαγαζί.

### Δομή Κώδικα

Αρχικά, σε ένα αρχείο header δηλώνουμε τις σταθερές που μας ζητούνται από την εκφώνηση, τις μεταβλητές που θα χρειαστούμε για το κύριο πρόγραμμα και τα νήματα. Επίσης δηλώνουμε και την συνάρτηση `print_order_final` που θα χρησιμοποιούμε κάθε φορά που το νήμα τελειώνει με μία παραγγελία.

Στο αρχείο κώδικα C, υπάρχει το κυρίως πρόγραμμα. Όταν καλείται το εκτελέσιμο, το πρόγραμμα περιμένει να του περαστούν ο αριθμός των πελατών προς εξυπηρέτηση και ο σπόρος για τη γεννήτρια τυχαίων αριθμών. Γίνονται οι κατάλληλοι έλεγχοι για να βεβαιωθούμε ότι έχουν περαστεί ακριβώς όσοι παράμετροι θέλουμε και με σωστά δεδομένα.

Έπειτα αρχικοποιούνται οι μεταβλητές που θα χρησιμοποιούν τα νήματα για να μεταβάλλουν τις τιμές των διαθέσιμων φούρνων/παρασκευαστών/διανομέων κάθε στιγμή και για να υπολογίζουν χρόνους παραγγελίας και χρόνους που κρυώνουν οι πίτσες μέχρι να παραδοθούν.

---

---

Ένας πίνακας `order_id` αρχικοποιείται που κρατάει τα `id` της κάθε παραγγελίας και με τη χρήση της `malloc` φτιάχνουμε ένα πίνακα για τα `threads`. Αυτοί οι πίνακες και η συνάρτηση `gabagool` είναι οι παράμετροι που περνάμε στην `pthread_create` ώστε να φτιάξουμε νήμα για κάθε πελάτη. Με την `sleep` και την `rand_r` προσομοιώνουμε την αναμονή ενός τυχαίου διαστήματος `wait` μέχρι να έρθει η επόμενη παραγγελία.

Τέλος, καλούμε την `pthread_join` για κάθε νήμα, ώστε να βεβαιωθούμε ότι το `main` πρόγραμμα δεν θα τελειώσει πριν τελειώσουν όλα τα νήματα τις εργασίες τους, και τυπώνουμε το μέσο και το μέγιστο χρόνο ολοκλήρωσης παραγγελιών, καθώς και το μέσο και μέγιστο χρόνο κρυώματος των παραγγελιών (όπου μέσος είναι το άθροισμα όλων των χρόνων προς το σύνολο των πελατών και ο μέγιστος βρίσκεται από την `gabagool`, όπου κάθε φορά ελέγχεται αν κάποιος καινούριος χρόνος ολοκλήρωσης ξεπερνά τον τρέχον μέγιστο, αντίστοιχα για χρόνους ολοκλήρωσης παραγγελίας και κρυώματος).

Κάθε νήμα μέσω της συνάρτησης `gabagool` εκτελεί τα ακόλουθα:

- Κρατάει πάντα τον χρόνο που δίνεται η παραγγελία, που ξεκινάει να προετοιμάζεται, που μπαίνει στο φούρνο, που τελειώνει το ψήσιμο και που παραδίδεται στον πελάτη.
- Με `pthread_mutex_lock` το νήμα “κλειδώνει” ώστε να βρει διαθέσιμο παρασκευαστή. Αν δεν βρει, με την `pthread_cond_wait` περιμένει μέχρι κάποιος να “ελευθερωθεί” (δηλαδή ο αριθμός των διαθέσιμων παρασκευαστών να γίνει θετικός), όπου και τον μειώνει κατά 1, με `pthread_mutex_unlock` “ξεκλειδώνει” το `mutex` για τους παρασκευαστές και συνεχίζει.
- 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 προετοιμαστούν οι πίτσες για το φούρνο. Ο παρασκευαστής ελευθερώνεται για να ετοιμάσει άλλη παραγγελία.
- Με `pthread_mutex_lock/unlock` και `pthread_cond_wait` ακολουθεί μια παρόμοια διαδικασία με τους παρασκευαστές, τώρα για τους φούρνους (κλείδωμα-ξεκλείδωμα, αναμονή για διαθέσιμο και μείωση κατά 1).
- Το νήμα περιμένει με τη `sleep` για το χρόνο ψησίματος της πίτσας.

- 
- Τώρα που οι πίτσες είναι έτοιμες, ο φούρνος περιμένει να βρεθεί και να δεσμευθεί διανομέας, ώστε ο πρώτος να ελευθερωθεί (με `pthread_mutex_lock/unlock` και `pthread_cond_wait` ακολουθεί μια παρόμοια διαδικασία για τους διανομείς).
  - Με κλείδωμα αυξάνουμε την μεταβλητή των διαθέσιμων φούρνων κατά 1 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 της παράδοσης: με `rand_r` βρίσκουμε το χρόνο παράδοσης και κάνουμε `sleep` δύο φορές για αυτό το χρόνο, μία για την παράδοση και μία για την επιστροφή του διανομέα στο μαγαζί. Ο διανομέας ελευθερώνεται και δίνουμε `signal` για άλλες πιθανές πίτσες που περιμένουν να τις βγάλουν από τον φούρνο.
  - Έπειτα, κλειδώνουμε το `mutex` για τη μεταβλητή των συνολικών χρόνων εξυπηρέτησεων και κρυώματος, κλειδώνουμε το `lock_screen` για το τύπωμα του τελικού μηνύματος, υπολογίζουμε χρόνους παραγγελίας-κρυώματος και ανανεώνουμε τον συνολικό χρόνο παραγγελιών-κρυώματος, τυπώνουμε το μήνυμα με την `print_order_final` και ξεκλειδώνουμε το `lock_screen`. Τέλος, κλειδώνουμε και ξεκλειδώνουμε για την ανανέωση των μέγιστων χρόνων και ξεκλειδώνουμε οριστικά για το συνολικό χρόνο εξυπηρέτησεων/κρυώματος.
  - Το νήμα έχει τελειώσει τις εργασίες του και κάνει `pthread_exit` με το αντίστοιχο `order_id` του.

## Πρόσθετα Χαρακτηριστικά

Ακολουθήσαμε την εκφώνηση ακριβώς όπως μας δόθηκε για την δημιουργία του προγράμματος, απλά έχουμε προσθέσει μηνύματα που βοηθούν στην προσομοίωση της διαδικασίας (όταν έρχεται καινούρια παραγγελία, όταν βρίσκεται παρασκευαστής κτλ). Επίσης, ο σπόρος που δίνεται στο πρόγραμμα είναι καθολική μεταβλητή, αλλά κάθε νήμα τον αλλοιώνει με ένα `xor` αυτού και του `order_id` του και τον αποθηκεύει σε τοπική μεταβλητή `seedp`, ώστε η ακολουθία των ψευδοτυχαίων που έχει κάθε νήμα να μην είναι ίδια με των άλλων.