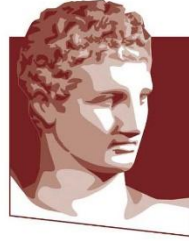


**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS

**ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

*Εργασία στο μάθημα: «Λειτουργικά Συστήματα - 4<sup>ο</sup> εξάμηνο»*

# «Προσομοίωση Πιτσαρίας με Πολυνηματισμό σε Γλώσσα C»

**ΦΟΙΤΗΤΕΣ**

ΠΡΟΕΣΤΑΚΗΣ ΦΟΙΒΟΣ ΤΙΜΟΘΕΟΣ - t8210126

ΝΕΡΑΝΤΖΑΚΗΣ ΙΑΣΟΝΑΣ - t8210103

ΡΑΖΕΛΟΣ ΝΙΚΟΛΑΟΣ - t8210128

**ΕΠΙΒΛΕΠΩΝ: ΓΕΩΡΓΙΟΣ ΞΥΛΩΜΕΝΟΣ**

Αθήνα, Μάιος 2024

## Περιεχόμενα

1. Εισαγωγή.....	3
2. Αρχείο Σταθερών.....	3
2.1 Σταθερές .....	3
2.2 Μεταβλητές .....	4
2.3 Νήματα .....	4
3. Δομές Δεδομένων .....	5
4. Κύρια Δομή του Κώδικα .....	5
5. Περιγραφή Λειτουργιών .....	5
5.1 Customer Thread .....	5
5.2 Telephone Thread.....	6
5.3 Cook Thread .....	7
5.4 Oven Thread .....	7
5.5 Deliverer Thread.....	8
6. Μέθοδος Main.....	8
7. Στατιστικά και Αποτελέσματα .....	9
8. Παράδειγμα εκτέλεσης του προγράμματος.....	9

## 1. Εισαγωγή

Η παρούσα προσομοίωση έχει ως στόχο την αναπαράσταση της λειτουργίας μιας πιτσαρίας χρησιμοποιώντας πολυνηματισμό (multithreading) και συγκεκριμένα pthreads σε γλώσσα C. Ο κώδικας της παρούσας εργασίας βρίσκεται και στο Github στον παρακάτω σύνδεσμο: <https://github.com/foivospro/OS-Assignment>.

## 2. Αρχείο Σταθερών

Το αρχείο .h περιλαμβάνει τις σταθερές, τις μεταβλητές και ορίζει τα νήματα για τους πελάτες, τους τηλεφωνητές, τους παρασκευαστές, τους φούρνους και τους διανομείς που χρησιμοποιούνται στην προσομοίωση. Συγκεκριμένα:

### 2.1 Σταθερές

**NUM\_TELEPHONES:** Αριθμός τηλεφώνων.

**NUM\_COOKS:** Αριθμός μαγείρων.

**NUM\_OVENS:** Αριθμός φούρνων.

**NUM\_DELIVERERS:** Αριθμός διανομέων.

**ORDER\_MIN\_TIME, ORDER\_MAX\_TIME:** Ελάχιστος και μέγιστος χρόνος μεταξύ παραγγελιών.

**PIZZA\_MIN\_QUANTITY, PIZZA\_MAX\_QUANTITY:** Ελάχιστη και μέγιστη ποσότητα πίτσες ανά παραγγελία.

**P\_MARGARITA, P\_PEPPERONI, P\_SPECIAL:** Πιθανότητες παραγγελίας κάθε τύπου πίτσας.

**PAYMENT\_MIN\_TIME, PAYMENT\_MAX\_TIME:** Ελάχιστος και μέγιστος χρόνος πληρωμής.

**P\_FAILURE:** Πιθανότητα αποτυχίας πληρωμής.

**C\_MARGARITA, C\_PEPPERONI, C\_SPECIAL:** Τιμές των πίτσες ανά τύπο.

**PREP\_TIME, BAKE\_TIME, PACK\_TIME:** Χρόνοι προετοιμασίας, ψησίματος και πακεταρίσματος.

**DELIVERY\_MIN\_TIME, DELIVERY\_MAX\_TIME:** Ελάχιστος και μέγιστος χρόνος διανομής.

## 2.2 Μεταβλητές

**unsigned int seed:** Σπόρος για την γεννήτρια τυχαίων αριθμών.

**int random\_number, rc:** Μεταβλητές για αποθήκευση τυχαίων αριθμών και κωδικών επιστροφής.

**int current\_thread = 1:** Μεταβλητή για την παρακολούθηση του τρέχοντος νήματος.

**int phone\_calls = 0, total\_revenue = 0:** Μεταβλητές για την παρακολούθηση των τηλεφωνικών κλήσεων και των συνολικών εσόδων.

**int margherita\_sold = 0, pepperoni\_sold = 0, special\_sold = 0:** Μεταβλητές για την παρακολούθηση των πωλήσεων ανά τύπο πίτσας.

**int successful\_orders = 0, failed\_orders = 0:** Μεταβλητές για την παρακολούθηση των επιτυχημένων και αποτυχημένων παραγγελιών.

**int cooks\_occupied = 0, ovens\_occupied = 0, deliverers\_occupied = 0:** Μεταβλητές για την παρακολούθηση των κατειλημμένων πόρων.

**double total\_service\_time = 0, total\_cooling\_time = 0:** Μεταβλητές για την παρακολούθηση του συνολικού χρόνου εξυπηρέτησης και κρυώματος.

**int max\_service\_time = 0, max\_cooling\_time = 0:** Μεταβλητές για την παρακολούθηση του μέγιστου χρόνου εξυπηρέτησης και κρυώματος.

## 2.3 Νήματα

Για τον συγχρονισμό των threads χρησιμοποιούνται mutexes και condition variables. Συγκεκριμένα:

**order\_threads\_mutex και order\_threads\_cond:** Χρησιμοποιούνται για τον συγχρονισμό των παραγγελιών.

**calling\_mutex και call\_available:** Χρησιμοποιούνται για τον περιορισμό των ταυτόχρονων κλήσεων.

**cook\_mutex και cook\_available:** Χρησιμοποιούνται για τον περιορισμό των ταυτόχρονων μαγειρίων.

**oven\_mutex και oven\_available:** Χρησιμοποιούνται για τον περιορισμό των ταυτόχρονων φούρνων.

**deliverer\_mutex και deliverer\_available:** Χρησιμοποιούνται για τον περιορισμό των ταυτόχρονων διανομέων.

**print\_mutex:** Χρησιμοποιείται για την προστασία των εκτυπώσεων στην κονσόλα.

### 3. Δομές Δεδομένων

**OrderStatus:** Τύπος δεδομένων enum για την κατάσταση της παραγγελίας (SUCCESSFUL ή FAILED).

**OrderData:** Τύπος δεδομένων struct που περιλαμβάνει τα στοιχεία της παραγγελίας (αριθμός πελάτη, αριθμός παραγγελίας, ποσότητα πίτσας, χρόνοι άφιξης, ολοκλήρωσης, παράδοσης κ.λ.π).

### 4. Κύρια Δομή του Κώδικα

Αρχικά, το πρόγραμμα κάνει περιλαμβάνει (include) τις απαραίτητες βιβλιοθήκες και αρχεία (π.χ την <pthread.h> για την υποστήριξη πολυνηματισμού).

Στην συνέχεια, ο κώδικας αποτελείται από διάφορες συναρτήσεις που αντιστοιχούν στις διάφορες λειτουργίες της πιτσαρίας:

**customer\_thread:** Χειρίζεται την παραγγελία από τον πελάτη.

**telephone\_thread:** Αντιπροσωπεύει την τηλεφωνική εξυπηρέτηση.

**cook\_thread:** Αντιπροσωπεύει τη διαδικασία μαγειρέματος (παρασκευής) της πίτσας.

**oven\_thread:** Χειρίζεται την τοποθέτηση της πίτσας στο φούρνο.

**deliverer\_thread:** Χειρίζεται τη διανομή της παραγγελίας.

Κάθε πελάτης δημιουργεί ένα thread, το οποίο περνάει από όλα τα στάδια της παραγγελίας (παραγγελία, μαγείρεμα, ψήσιμο, διανομή).

### 5. Περιγραφή Λειτουργιών

#### 5.1 Customer Thread

Το νήμα του πελάτη λειτουργεί ως εξής:

- Λαμβάνει ως όρισμα τα δεδομένα παραγγελίας (OrderData).

- Κλειδώνει το `order_threads_mutex` για αποκλειστική πρόσβαση στους κοινόχρηστους πόρους.
- Εξασφαλίζει ότι οι πελάτες εξυπηρετούνται με τη σειρά (από 1 έως `Ncust`) με την συνθήκη `'wait'`. Αυτή η συνθήκη είναι απαραίτητη γιατί διαφορετικά τα `cid` δεν θα αντιστοιχούσαν στη σειρά που έρχονται οι πελάτες. (π.χ. θα μπορούσε ο πρώτος πελάτης να έχει `cid = 3`).
- Καταγράφει τον χρόνο άφιξης και προσθέτει ένα τυχαίο (στα πλαίσια που ορίζει η εκφώνηση) χρονικό διάστημα καθυστέρησης για κάθε επόμενο πελάτη.
- Αυξάνει το `current_thread` και κάνει broadcast στα νήματα πελατών που περιμένουν, ώστε να βγεί το επόμενο νήμα στην σειρά (π.χ. μετά το νήμα με `cid = 1`, θα βγεί το νήμα με `cid = 2`).
- Ελέγχει αν υπάρχουν διαθέσιμοι τηλεφωνητές και αυξάνει τον μετρητή `phone_calls`.
- Ξεκλειδώνει το `order_threads_mutex` και περνάει στο νήμα του τηλεφωνητή (`telephone_thread`).
- Τερματίζει το τρέχον νήμα μέσω της `pthread_exit`.

## 5.2 Telephone Thread

Το νήμα των τηλεφωνητών λειτουργεί ως εξής:

- Λαμβάνει ως όρισμα τα δεδομένα παραγγελίας (`OrderData`).
- Κλειδώνει το `phone_mutex`.
- Υπολογίζει τυχαία αλλά με βάση την εκφώνηση αν η πληρωμή είναι επιτυχημένη ή όχι.
- Αν η πληρωμή είναι επιτυχημένη:
  - Τυπώνει τον αριθμό παραγγελίας.
  - Ενημερώνει το `customer_status` σε `SUCCESSFUL`.
  - Αυξάνει τις επιτυχημένες παραγγελίες και ενημερώνει τα στατιστικά πωλήσεων.
  - Μειώνει το `phone_calls` και κάνει broadcast για την διαθεσιμότητα τηλεφωνητή.
  - Καλεί το νήμα του παρασκευαστή (`cook_thread`).
- Αν η πληρωμή αποτύχει:
  - Ενημερώνει το `customer_status` σε `FAILED`.
  - Αυξάνει τις αποτυχημένες παραγγελίες.

- Μειώνει το `phone_calls` και κάνει broadcast για διαθεσιμότητα τηλεφωνητή.
- Ξεκλειδώνει το `phone_mutex`.
- Τερματίζει το τρέχον νήμα μέσω της `pthread_exit`

## 5.3 Cook Thread

Το νήμα των παρασκευαστών λειτουργεί ως εξής:

- Λαμβάνει ως όρισμα τα δεδομένα παραγγελίας (`OrderData`).
- Κλειδώνει το `cook_mutex`.
- Ελέγχει αν υπάρχει διαθέσιμος παρασκευαστής και αυξάνει τον μετρητή `cooks_occupied`.
- Ξεκλειδώνει το `cook_mutex`.
- Ετοιμάζει τις πίτσες, δηλαδή γίνεται `sleep` στο νήμα κατά `PREP_TIME`.
- Καλεί το νήμα του φούρνου (`oven_thread`).
- Κλειδώνει το `cook_mutex` για να μειώσει τον μετρητή `cooks_occupied`.
- Κάνει signal για διαθεσιμότητα παρασκευαστή και ξεκλειδώνει το `cook_mutex`.
- Τερματίζει το τρέχον νήμα μέσω της `pthread_exit`.

## 5.4 Oven Thread

Το νήμα των φούρνων λειτουργεί ως εξής:

- Λαμβάνει ως όρισμα τα δεδομένα παραγγελίας (`OrderData`).
- Κλειδώνει το `oven_mutex`.
- Ελέγχει αν υπάρχουν διαθέσιμοι φούρνοι για να εκτελεστεί η παραγγελία (ώστε να ψηθούν οι πίτσες ταυτόχρονα).
- Αυξάνει τον μετρητή `ovens_occupied` κατά τον αριθμό των πιτσών.
- Ξεκλειδώνει το `oven_mutex`.
- Καλεί το νήμα του διανομέα (`deliverer_thread`).

## 5.5 Deliverer Thread

Το νήμα των διανομέων λειτουργεί ως εξής:

- Λαμβάνει ως όρισμα τα δεδομένα παραγγελίας (OrderData).
- Κλειδώνει το `deliverer_mutex`.
- Ελέγχει αν υπάρχει διαθέσιμος διανομέας και αυξάνει τον μετρητή `deliverers_occupied`.
- Μειώνει τον μετρητή `ovens_occupied` κατά τον αριθμό των πιτσών.
- Κάνει broadcast για διαθεσιμότητα φούρνου.
- Ξεκλειδώνει το `deliverer_mutex`.
- Ετοιμάζει την παραγγελία και υπολογίζει τον χρόνο παράδοσης.
- Τυπώνει τον συνολικό χρόνο εξυπηρέτησης.
- Υπολογίζει τον χρόνο διανομής και τον συνολικό χρόνο από την παραγγελία μέχρι την παράδοση.
- Τυπώνει τον συνολικό χρόνο παράδοσης.
- Κοιμίζει το νήμα για τον χρόνο επιστροφής του διανομέα.
- Μειώνει τον μετρητή `deliverers_occupied` και κάνει signal για διαθεσιμότητα διανομέα.
- Ξεκλειδώνει το `deliverer_mutex`.

## 6. Μέθοδος Main

Η μέθοδος `main` λειτουργεί ως εξής:

- Ελέγχεται ο αριθμός των ορισμάτων.
- Αν τα ορίσματα είναι δύο:
  - Τα μετατρέπει σε ακεραίους και τα αποθηκεύει ως `Ncust` και `seed`.
  - Ορίζει έναν πίνακα για τον χειρισμό των νημάτων και τα `OrderData` για κάθε πελάτη.
  - Στην πρώτη `for`, για κάθε πελάτη, δεσμεύει μνήμη, ορίζει τον αριθμό πελάτη `'cid'` και δημιουργεί ένα νήμα.
  - Στην δεύτερη `for`, περιμένει να τερματίσει κάθε νήμα πελάτη χρησιμοποιώντας την `'pthread_join'`.
  - Μόλις ένα νήμα τερματίσει, αποδεσμεύει την αντίστοιχη μνήμη.
  - Τέλος, τυπώνονται όλα τα στατιστικά, διαγράφει όλα τα `'mutex'`, τα `'conditions'` και τα `'threads'` που έχουν χρησιμοποιηθεί με τα κατάλληλα μηνύματα σφάλματος αν κάτι πάει στραβά.
- Αν τα ορίσματα δεν είναι δύο:



- Βγάζει μήνυμα σφάλματος και το πρόγραμμα τερματίζει.

## 7. Στατιστικά και Αποτελέσματα

Μετά την ολοκλήρωση όλων των παραγγελιών, η συνάρτηση **print\_statistics** υπολογίζει και εκτυπώνει τα εξής στατιστικά:

- **Συνολικά έσοδα από τις πωλήσεις:** Ο συνολικός τζίρος από τις πίτσες που πουλήθηκαν.
- **Πίτσες που πουλήθηκαν από κάθε τύπο:** Ο αριθμός των πιτσών μαργαρίτα, πεπερόνι και σπέσιαλ που πουλήθηκαν.
- **Πλήθος επιτυχημένων και αποτυχημένων παραγγελιών:** Το πλήθος των επιτυχημένων και αποτυχημένων παραγγελιών.
- **Μέσος και μέγιστος χρόνος εξυπηρέτησης:** Ο μέσος και ο μέγιστος χρόνος που χρειάστηκε από την άφιξη της παραγγελίας μέχρι την παράδοσή της.
- **Μέσος και μέγιστος χρόνος κρυώματος:** Ο μέσος και ο μέγιστος χρόνος που μεσολάβησε από την ολοκλήρωση της πίτσας μέχρι την παράδοσή της.

Οι χρόνοι καταγράφονται χρησιμοποιώντας τη βιβλιοθήκη `<time.h>` και τη δομή `OrderData` που περιλαμβάνει τα στοιχεία για τους χρόνους. Συγκεκριμένα:

Η `clock_gettime(CLOCK_REALTIME, &time)` χρησιμοποιείται για να καταγράψει τον ακριβή χρόνο σε διάφορα σημεία της διαδικασίας (άφιξη παραγγελίας, ολοκλήρωση προετοιμασίας, πακετάρισμα, παράδοση). Οι διαφορές μεταξύ αυτών των χρονικών σημείων υπολογίζονται σε δευτερόλεπτα για να βρεθούν οι χρόνοι εξυπηρέτησης και κρυώματος.

## 8. Παράδειγμα εκτέλεσης του προγράμματος

Παρακάτω φαίνονται τα αποτελέσματα από την εκτέλεση του προγράμματος με 10 πελάτες και σπόρο τυχαίων αριθμών 10. Συγκεκριμένα, εκτελέστηκε η εντολή **./test-res.sh 10 10** και το αποτέλεσμα είναι το εξής:

- Η παραγγελία με αριθμό 1 καταχωρήθηκε.
- Η παραγγελία με αριθμό 2 καταχωρήθηκε.
- Η παραγγελία με αριθμό 3 καταχωρήθηκε.
- Η παραγγελία με αριθμό 4 καταχωρήθηκε.
- Η παραγγελία με αριθμό 5 καταχωρήθηκε.
- Η παραγγελία με αριθμό 1 ετοιμάστηκε σε 15 λεπτά.

- Η παραγγελία με αριθμό 6 καταχωρήθηκε.
- Η παραγγελία με αριθμό 2 ετοιμάστηκε σε 18 λεπτά.
- Η παραγγελία με αριθμό 7 καταχωρήθηκε.
- Η παραγγελία με αριθμό 2 παραδόθηκε σε 23 λεπτά.
- Η παραγγελία με αριθμό 3 ετοιμάστηκε σε 21 λεπτά.
- Η παραγγελία με αριθμό 8 απέτυχε.
- Η παραγγελία με αριθμό 1 παραδόθηκε σε 24 λεπτά.
- Η παραγγελία με αριθμό 4 ετοιμάστηκε σε 19 λεπτά.
- Η παραγγελία με αριθμό 9 καταχωρήθηκε.
- Η παραγγελία με αριθμό 10 καταχωρήθηκε.
- Η παραγγελία με αριθμό 6 ετοιμάστηκε σε 22 λεπτά.
- Η παραγγελία με αριθμό 4 παραδόθηκε σε 28 λεπτά.
- Η παραγγελία με αριθμό 5 ετοιμάστηκε σε 26 λεπτά.
- Η παραγγελία με αριθμό 3 παραδόθηκε σε 34 λεπτά.
- Η παραγγελία με αριθμό 5 παραδόθηκε σε 33 λεπτά.
- Η παραγγελία με αριθμό 6 παραδόθηκε σε 32 λεπτά.
- Η παραγγελία με αριθμό 9 ετοιμάστηκε σε 21 λεπτά.
- Η παραγγελία με αριθμό 7 ετοιμάστηκε σε 28 λεπτά.
- Η παραγγελία με αριθμό 10 ετοιμάστηκε σε 21 λεπτά.
- Η παραγγελία με αριθμό 9 παραδόθηκε σε 26 λεπτά.
- Η παραγγελία με αριθμό 7 παραδόθηκε σε 38 λεπτά.
- Η παραγγελία με αριθμό 10 παραδόθηκε σε 27 λεπτά.

**Τα συνολικά έσοδα από τις πωλήσεις είναι: 335€**

Πίτσες που πουλήθηκαν από κάθε τύπο

- Μαργαρίτα: 8
  - Πεπερόνι: 9
  - Σπέσιαλ: 13
- 
- Πλήθος επιτυχημένων παραγγελιών: 9
  - Πλήθος αποτυχημένων παραγγελιών: 1
- 
- Μέσος χρόνος εξυπηρέτησης: 29.44 λεπτά
  - Μέγιστος χρόνος εξυπηρέτησης: 38 λεπτά
- 
- Μέσος χρόνος κρυώματος: 9.22 λεπτά
  - Μέγιστος χρόνος κρυώματος: 14 λεπτά