



ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ (2024)

ERNEST BETA (3220239)

ΠΑΝΑΓΙΩΤΗΣ ΜΠΙΧΑΛΤΖΗΣ (3220133)

Η εργασία αφορά ένα σύστημα παραγγελιών και διανομής πίτσας με χρήση του πακέτου νημάτων (pthreads). Συγκεκριμένα, έχουμε δυο αρχεία p3220133-p3220239-pizza.h και p3220133-p3220239-pizza.c. Στο p3220133-p3220239-pizza.h έχουμε ορίσει όλες τις σταθερές μεταβλητές που μας δίνεται από την εκφώνηση, αλλά και mutex and function declarations οι οποίοι θα χρειαστούν για την σωστοί χρήση των πόρων. Στο αρχείο p3220133-p3220239-pizza.c βρίσκεται η main συνάρτηση, το **σημείο εκκίνησης** για την εκτέλεση του κύριου προγράμματος. Η συνάρτηση αυτή ξεκινάει ελέγχοντας τον αριθμό των ορισμάτων που δίνονται κατά την εκκίνηση του προγράμματος από τη γραμμή εντολών. Ο αριθμός αυτός επιβάλλεται να είναι 3 καθώς θα έχει: το όνομα του εκτελέσιμου αρχείου (argv[0]), τον αριθμό των πελατών (argv[1]) και τον αριθμό Seed (argv[2]). Ύστερα, αποθηκεύει τις τιμές αυτές στις κατάλληλες μεταβλητές που ορίσαμε καθώς και αρχικοποιεί την γεννήτρια τυχαίων αριθμών με ένα συγκεκριμένο seed.

Όταν γίνει αυτό, τρέχει την συνάρτηση **init()** και έτσι αρχικοποιεί τον χρονισμό της, καθώς και όλα τα απαραίτητα mutex και condition variables για **τον συγχρονισμό και την προστασία των κοινών πόρων**. Ακολουθεί η δημιουργία του πίνακα νημάτων των πελατών και έναν πίνακα των ID τους, και στη συνέχεια εκκινεί τα νήματα για κάθε πελάτη, συνδέοντας τους με το ID του καθενός ξεχωριστά για να ταυτοποιούνται όταν εξυπηρετούνται. Ταυτόχρονα, εκτελείται και η **orderHandler()**, η οποία είναι υπεύθυνη για **την εξυπηρέτηση των πελατών**.

Για κάθε ενέργεια που συμβαίνει παρέχουμε αρχεία καταγραφής σε πραγματικό χρόνο ώστε κάθε ενέργεια να ελέγχεται λεπτομερώς για την ορθότητά της. Παρέχουμε ποια ενέργεια συνέβη και τον ακριβή χρόνο που συνέβη σε σχέση με την ώρα του προγράμματος.

i) Αρχικά, κλειδώνεται το global_time_mutex για να εξασφαλιστεί **η αποκλειστική πρόσβαση στην παγκόσμια χρονική μεταβλητή** κάθε φορά που ανακτάται ο πραγματικός χρόνος του συστήματος όπως εδώ. Αν ο πελάτης που εξυπηρετείται δεν είναι ο πρώτος του simulation τότε ωφείλεται να θεωρηθεί μία **καθυστέρηση** (random delay από 1-5) πριν πάρει ο άλλος πελάτης τηλέφωνο την πιτσαρία, του οποίου ο τρόπος υπολογίζεται σύμφωνα με τις οδηγίες. Όταν πραγματοποιηθεί η κλήση εκτυπώνεται κατάλληλο μήνυμα και αμέσως ελέγχεται αν υπάρχει διαθέσιμος τηλεφωνητής. Αν δεν υπάρχει τότε ο πελάτης μένει σε αναμονή μέχρι να βρεθεί ένας διαθέσιμος. Όλα αυτά γίνονται με προϋπόθεση ότι έχουμε κλειδώσει το call_center_mutex. Ομοίως, εκτυπώνεται το κατάλληλο μήνυμα μόλις συνδεθεί με τηλεφωνητή συνοδευόμενο με την στιγμή σύνδεσης (υπολογιζόμενο με την καθυστέρηση).

Για την πραγματοποίηση της παραγγελίας, παίζει πρωτεύων ρόλο η **Pfail** ως πιθανότητα αποτυχίας πληρωμής. Αν η παραγγελία του πελάτη «πέσει» στο ποσοστό των 5% αποτυχίας τότε εμφανίζεται το μήνυμα αποτυχίας πληρωμής και η στιγμή που έγινε. Παρόλαυτα στον μετρητή παραγγελιών προτίθεται για την ανάδειξη στατιστικών στο τέλος του simulation. Στην περίπτωση που δεν προέκυψε πρόβλημα στην χρέωση, ο πελάτης διαλέγει σύμφωνα με τα δοθέντα ποσοστά τον αριθμό των πιτσών, καθώς και το είδος τους. Με την ολοκλήρωση αυτών των διαδικασιών εκτυπώνουμε ξανά, το αποτέλεσμα της παραγγελίας (δηλαδή το είδος της κάθε πίτσας) και την τρέχοντα χρονικής στιγμή. Τέλος, προσθέτουμε πίσω στον μετρητή των διαθέσιμων τηλεφωνητών τον τηλεφωνητή που είχαμε αφαιρέσει όταν συνδέθηκε ο πελάτης με το τηλέφωνο του μαγαζιού.

Για το μαγείρεμα των πιτσών, όπως έγινε και με τους τηλεφωνητές, χρειαζόμαστε διαθέσιμο μάγειρα, δηλαδή condition variable (cook_cond). Έτσι, μπορεί να υπάρξει μια **καθυστέρηση** πριν την προετοιμασία των πιτσών αν δεν βρεθεί άμεσα μάγειρας. Ανεξαρτήτως όμως αν υπήρξε καθυστέρηση ή όχι, όταν η παραγγελία φτάσει σε έναν μάγειρα και αυτός αρχίσει να τις προετοιμάζει εκτυπώνεται η χρονική εκείνη στιγμή. Αργότερα εκτυπώνεται και η στιγμή κατά την οποία λήγει η προετοιμασία των πιτσών πριν μπουν στον φούρνο, η οποία θα υπολογιστεί με γνώμονα ότι η κάθε πίτσα **θέλει 10 λεπτα για να προετοιμαστεί**. Μόλις προετοιμαστούν, αναζητείται ο αριθμός των διαθέσιμων φούρνων να καλύψει το ψήσιμο όλων των πιτσών της τρέχουσας παραγγελίας μαζί. Αν δεν βρεθεί, προκύπτει μία **καθυστέρηση** (oven_cond). Έπειτα, για άλλη μια φορά εκτυπώνεται η χρονική στιγμή που μπήκαν και που βγήκαν από τον φούρνο (λαμβάνοντας υπόψη ότι η κάθε πίτσα θέλει 10 λεπτά για να ψηθεί).

Οι πίτσες για να φτάσουν στον πελάτη χρειάζονται διανομείς για να τις μεταφέρουν, οι οποίοι είναι περιορισμένοι σε αριθμό. Με παρόμοιο τρόπο η πιτσαρία, περιμένει να προκύψει ένας διανομέας, **διαθέσιμος για την μεταφορά των πιτσών** (delivery_cond). Όταν βρεθεί αυτός ο διανομέας, πακετάρει τις πίτσες (σε 1 λεπτό) και ξεκινάει μία διαδρομή που θα διαρκέσει ένα τυχαίο ακέραιο χρονικό διάστημα στην περιοχή που μας δίνεται στην εκφώνηση. Εκτυπώνουμε, στην συνέχεια την στιγμή που ο διανομέας πακέτωσε την πίτσα, την παρέδωσε και την στιγμή που γύρισε πίσω (**δηλαδή τον χρόνο στον οποίο έφυγε από την πιτσαρία έφτασε στον πελάτη και γύρισε ξανά πίσω στην πιτσαρία**).

Τέλος, υπολογίζει τα:

- Total Calls (order_count)
- Successful Orders (successful_orders)
- Failed Orders
- Total Margaritas + Sub Totals In Euros
- Total Pepperoni + Sub Totals In Euros
- Total Special + Sub Totals In Euros
- Total Revenue
- maxCoolingTime (τον μέγιστο χρόνο από τότε που βγήκαν οι πίτσες μιας παραγγελίας από τον φούρνο έως την παράδοση τους στον πελάτη)
- max_calling_waiting_time
- maxServiceTime

- max_delivery_ride_time
- averageCoolingTime
- average_calling_waiting_time
- averageServiceTime
- average_delivery_ride_time

To **output** θα είναι το εξής (100 customers, seed 10):

```
#####
|                                     Revenue Report                                     |
#####
| Total Calls           | 100 |
| Successful Orders     | 96  |
| Failed Orders        | 4   | | |
|---|---|---|---|
| Item                  | Quantity Sold | Price | Subtotal |
|-----|-----|-----|-----|
| Margaritas           | 105           | 10€   | 1050.00€ |
| Pepperonis          | 67            | 11€   | 737.00€  |
| Specials             | 124           | 12€   | 1488.00€ |
|=====|=====|=====|=====|
| Total Revenue        | 3275.00€     |
|
|-----|-----|
| Metric              | Value |
|-----|-----|
| Max CallWait Time   | 171.13 minutes (ID: 42) |
| Max Service Time    | 398.36 minutes (ID: 99) |
| Max Cooling Time    | 19.01 minutes (ID: 81) |
| Max Delivery Ride   | 30.04 minutes (ID: 59) |
| Avg Call Waiting Time | 67.48 minutes |
| Avg Waiting Time    | 211.32 minutes |
| Avg Cooling Time    | 13.14 minutes |
| Avg Delivery Ride   | 20.12 minutes |
|-----|-----|
csuser@csuser-virtualbox:~/Desktop/Project01$
```

Με την ολοκλήρωση του παραπάνω υπολογισμού, διασφαλίζεται μέσω του **pthread join** ότι όλα τα νήματα πελατών έχουν τελειώσει τις εργασίες τους προτού προχωρήσει το κύριο πρόγραμμα, διατηρώντας έτσι την ακεραιότητα της προσομοίωσης και την ορθή διαχείριση των πόρων.

Μετά από τις διαδικασίες που πραγματοποιήθηκαν στο **orderHandler()**, ακολουθεί η συνάρτηση **printRevenue()** για να εκτυπώσει όλα τα τελικά **στατιστικά** που κρατήσαμε από τις παραγγελίες, σε συνδυασμό με την **erase()** που καθαρίζει τους πόρους που χρησιμοποιήθηκαν κατά τη διάρκεια του simulation και να διασφαλίσει ότι όλα τα mutexes και οι συνθήκες έχουν απελευθερωθεί σωστά.

```
csuser@csuser-virtualbox:~/Desktop/Project01$ ./a.out 4 10
Caller is calling [Time: 0.00 minutes]
Client was connected with call receiver [Time: 0.00 minutes]
Order 1 was charged SUCCESSFULLY [Margaritas: 0, Pepperonis: 0, Specials: 1][Time: 2.00 minutes]
Order 1 was taken by a cook [Time: 2.00 minutes]
Order 1 was prepared [Time: 3.00 minutes]
The Order 1 was placed in the Oven [Time: 3.00 minutes]
Caller is calling [Time: 4.00 minutes]
Caller is calling [Time: 7.00 minutes]
Client was connected with call receiver [Time: 7.00 minutes]
Caller is calling [Time: 9.00 minutes]
Client was connected with call receiver [Time: 9.00 minutes]
Order 2 was charged SUCCESSFULLY [Margaritas: 1, Pepperonis: 0, Specials: 3][Time: 10.00 minutes]
Order 2 was taken by a cook [Time: 10.00 minutes]
Client was connected with call receiver [Time: 10.00 minutes]
Order 3 was charged SUCCESSFULLY [Margaritas: 0, Pepperonis: 1, Specials: 3][Time: 12.00 minutes]
Order 4 was charged SUCCESSFULLY [Margaritas: 2, Pepperonis: 0, Specials: 0][Time: 12.00 minutes]
The Order 1 is baked [Time: 13.01 minutes]
Order 3 was taken by a cook [Time: 13.01 minutes]
Order 1 Packed [Time: 14.01 minutes]
Order 2 was prepared [Time: 14.01 minutes]
The Order 2 was placed in the Oven [Time: 14.01 minutes]
Order 3 was prepared [Time: 17.01 minutes]
The Order 3 was placed in the Oven [Time: 17.01 minutes]
The Order 2 is baked [Time: 24.01 minutes]
Order 4 was taken by a cook [Time: 24.01 minutes]
Order 1 delivered [Time: 26.01 minutes]
Order 4 was prepared [Time: 26.01 minutes]
The Order 4 was placed in the Oven [Time: 26.01 minutes]
The Order 3 is baked [Time: 27.01 minutes]
Order 2 Packed [Time: 28.01 minutes]
Order 3 Packed [Time: 31.01 minutes]
The Order 4 is baked [Time: 36.01 minutes]
Order 2 delivered [Time: 37.03 minutes]
Order 4 Packed [Time: 38.03 minutes]
Delivery boy returned from Order 1 [Time: 38.03 minutes]
Order 3 delivered [Time: 45.01 minutes]
Delivery boy returned from Order 2 [Time: 46.03 minutes]
Order 4 delivered [Time: 52.14 minutes]
Delivery boy returned from Order 3 [Time: 59.01 minutes]
Delivery boy returned from Order 4 [Time: 66.14 minutes]
```