

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
Λειτουργικά Συστήματα
Εαρινό Εξάμηνο 2019-2020

Ελένη Σαζώνη 3160153
Αριστείδης Χρονόπουλος 3160194
Αγγελική Φέκα 3140290



1η Προγραμματιστική Εργασία

1 Δομή πηγαίου κώδικα

- Αρχικά παίρνουμε ως όρισμα από τον χρήστη τον αριθμό των πελατών και τον σπόρο (seed) για την `rand_r` με την οποία παράγουμε τυχαία τον χρόνο που δεχόμαστε κάθε επόμενη παραγγελία και αριθμό από πίτσες που περιέχει κάθε παραγγελία.
- Ανάλογα με τον αριθμό των πελατών δημιουργούμε και τα αντίστοιχα νήματα με την χρήση της `pthread_create`.
- Με την `pthread_create` καλείται η `execute` για κάθε νήμα που δημιουργούμε, η οποία υλοποιεί τα εξής:
 1. Δέχεται κάθε παραγγελία μετά από ένα τυχαίο χρόνο `accept_delay_time` και κρατά την χρονική στιγμή που δέχεται την παραγγελία στην μεταβλητή `order_start` με την χρήση της `clock_gettime`.
 2. Κρατάμε τον χρόνο όταν μια παραγγελία παραδίδεται σε κάποιον παρασκευαστή στην μεταβλητή `prep_start`.
 3. Υπολογίζει τυχαία με την χρήση της `rand_r` των αριθμό των πιτσών που θα έχει η κάθε παραγγελία και περιμένει `T_prep` για κάθε μια από αυτές.

4. Ύστερα όλες οι πίτσες ψήνονται όλες μαζί όταν βρεθεί διαθέσιμος φούρνος και περιμένουν για **T_bake**.
 5. Τέλος, υπολογίζεται ο συνολικός χρόνος παρασκευής μιας παραγγελίας **total** και εμφανίζει σχετικό μήνυμα.
- Όταν τα νήματα ολοκληρώνονται και έλεγχος επιστρέφει στην main κάνουμε ένα iteration στον πίνακα orders στον οποίο είναι αποθηκευμένοι οι χρόνοι ολοκλήρωσης κάθε παραγγελίας και εμφανίζουμε τον μέσο και μέγιστο χρόνο παρασκευής.

2 Περιγραφή Υλοποίησης Μεθόδων

- Για τη δημιουργία των νημάτων κάναμε χρήση της
`int pthread_create(pthread_t *thread, const pthread_attr_t *attr,
void *(*start_routine) (void *), void *arg);`

Το νέο thread που δημιουργείται ξεκινά εκτελώντας την μέθοδο που καλείται ως `start_routine()`.

- Προκειμένου να πετύχουμε τον συγχρονισμό των νημάτων χρησιμοποιήσαμε mutexes. Ουσιαστικά ένα mutex είναι ένα lock, το οποίο θέτουμε πριν την χρήση των κοινών δεδομένων των threads και το ελευθερώνουμε εφόσον τελειώσουμε με την χρήση αυτή. Συγκεκριμένα:
`int pthread_mutex_lock(pthread_mutex_t *mutex);` : κλειδώνει το συγκεκριμένο mutex που του δίνεται. Αν είναι ξεκλείδωτο, το κλειδώνει και ανήκει στο thread που το καλεί. Αν είναι κλειδωμένο από κάποιο άλλο thread αναστέλλει την λειτουργία του thread μέχρι να ελευθερωθεί.
`int pthread_mutex_unlock(pthread_mutex_t *mutex);` : Αντίστοιχα, ξεκλειδώνει το mutex που δίνεται σαν παράμετρο. Το mutex θεωρείται ότι είναι κλειδωμένο από το thread που το καλεί.
- Με την χρήση της `int pthread_join(pthread_t thread, void **retval);` περιμένει το συγκεκριμένο thread που δέχεται σαν όρισμα να τελειώσει την εκτέλεση του. Αν το `retval` δεν είναι NULL τότε η `pthread_join` αντιγράφει το exti status του thread

στην θέση που υποδεικνύει η *retval. Στην περίπτωση μας κάθε thread επιστρέφει τον χρόνο ολοκλήρωσης του.

3 Αποτελέσματα

```
(base) helena@rogue-one:~/Desktop/gia_paradosh/out$ ls
p3160153-p3140290-p3160194.c p3160153-p3140290-p3160194.h test-res1.sh
(base) helena@rogue-one:~/Desktop/gia_paradosh/out$ bash ./test-res1.sh
p3160153-p3140290-p3160194.c: In function 'execute':
p3160153-p3140290-p3160194.c:47:5: warning: implicit declaration of function 'sleep' [-Wimplicit-function-declaration]
    sleep(accept_delay_time);
    ^
Welcome to AUEB's Pizza Store simulation.
You have chosen 10 for the number of customers and seed: 100

Order with id:5 has completed in 11.000000 seconds
Order with id:4 has completed in 11.000000 seconds
Order with id:2 has completed in 11.000000 seconds
Order with id:1 has completed in 14.000000 seconds
Order with id:8 has completed in 13.000000 seconds
Order with id:3 has completed in 19.000000 seconds
Order with id:9 has completed in 22.000000 seconds
Order with id:7 has completed in 23.000000 seconds
Order with id:10 has completed in 21.000000 seconds
Order with id:6 has completed in 23.000000 seconds
Maximum time for an order to be prepared: 23.000000
Average time for an order to be prepared: 14.600000
(base) helena@rogue-one:~/Desktop/gia_paradosh/out$
```