

ΕΡΓΑΣΙΑ ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ!!

ΓΕΩΡΓΙΑ ΨΥΧΑ ΑΜ:3190225



Καλήσπερα! Ακολουθεί η επεξήγηση του κώδικα μου παρόλο των ήδη υπάρχοντων σχολίων ελπίζω να βοηθήσει στην κατανόηση.

ΔΟΜΗ ΚΩΔΙΚΑ...

```
int main(int argc, char *argv[])
```

- Αρχικά δημιουργήσαμε μια main που εκεί δημιουργούμε τα νηματα και τα καταστρέφουμε
- Αρχικά ελέγχουμε αν το σύνολο των παραμέτρων εισόδου αν είναι σωστό και στην συνέχεια τα αρχικοποιούμε στον αριθμό των πελατών N_Cust και στην γεννήτρια τυχαίων αριθμών seed
- Δεσμεύουμε δυναμικά την μνήμη με malloc δημιουργούμε τα threads (ένα για κάθε πελάτη) με pthread_create(&threads[i], NULL, &order, &customer_id[i]) και κανουμε τα joins προκειμένου να περιμενει η main να τελειωσουν τα νηματα για να τερματισει
- Εμφανίζω τα ζητούμενα μηνύματα χρησιμοποιώντας lock και unlock για τα mutex της οθονης.
- Κατάστρεφω τα conditions και mutexes

```
void *order(void *x) ....
```

- seed= seed+oid το χρησιμοποιώ προκειμένου κάθε πελάτης να έχει το δικό του seed
- δημιουργώ τα struct timespec για να μπορέσω να πάρω τους χρόνους με την συνάρτηση clock_gettime

- **ΤΗΛΕΦΩΝΗΤΗΣ** κλειδώνω το mutex των τηλεφωνητών
pthread_mutex_lock(&mutex_tel) ελέγχω αν υπάρχει διαθέσιμος
τηλεφωνητής αν ναι τον δεσμεύω [N_tel--;] αλλιώς περιμένω
μέχρι να βρεθεί rc = pthread_cond_wait(&cond_tel, &mutex_t).
Μετα παίρνω τον χρόνο που βρήκε το τηλέφωνο ο πελάτης
clock_gettime(CLOCK_REALTIME, &customer_wait); Για να βρω σε
ποσό χρόνο ήταν σε αναμονή κατά μέσο όρο όλοι οι πελάτες.
- int pizzas = int pizzas = rand_r(&seed) % (N_orderhigh +1 -
N_orderlow)+ N_orderlow; Βρίσκω με την γεννήτρια τυχαίων
αριθμών τον αριθμό των πιτσών που θα αγοράσει ο πελάτης και
τις χρεώνω σε χρόνο τυχαίο πάλι pay_time
- Κάνω unlock το mutex του τηλεφωνητή προκειμένου να δουλέψω
με το pay_time και να κάνω sleep(pay_time) όσο δουλεύω με τον
πόρο για να μπει το πρόγραμμα σε αναμονή .
- Βρίσκω το max_wait_time που ζητάτε προκειμένου να το
εμφανίσω στο τέλος .
- Ελέγχω την πιθανότητα 5% [rand_r(&seed) %100 <5] να
απορριφθεί η κάρτα του πελάτη. Αν μπει στο loop κλειδώνω τα
mutexes tel ,screen προκειμένου να απελευθερώσω τον
τηλεφωνητή tel++; και να εμφανίσω το ζητούμενο μυνήμα από
εκκώνηση . Υστέρα τα ξεκλειδώνω και κάνω signal
pthread_cond_signal(&cond_tel) προκειμένου να απελευθερώσω
τον πόρο και να τον χρησιμοποιήσει κάποιο νήμα που περιμένει
τηλεφωνητή τελείως και σταματάω την order και τερματίζω το
νήμα pthread_exit(NULL). Αντιθετα αν η καρτα γινει αποδεκτη
ξεκιναι η διαδικασια της πιτσας .

Εξήγησα για τον τηλεφωνητή αναλυτικά την διαδικασία τώρα για
τα υπόλοιπα στάδια ακολουθώ μια πιο γενική εξήγηση που
αναφέρεται κυρίως σε εντολές.

- Στην συνεχεια για την υπόλοιπη εργασία ακολουθώ την ίδια ιδέα
περίπου, δηλαδή: Αναλογα με το σταδιο που βρισκομαι
κλειδωνω το καταλληλο mutex με thread_mutex_lock ελεγχω αν
υπαρχει διαθεσιμος πόρος αν όχι περιμενω pthread_cond_wait
αλλιώς δεσμεύω τον πόρο μειώνοντας τον αναλόγως (--) [στην
περιπτωση των φούρνων θελω οσους φούρνους είναι και οι

πίτσες μου αρα N_oven-pizzas] . Μετα κάνω unlock τα mutex μου προκειμένου να κάνω το sleep που δείχνει την αναμονή στην ουσία του προγράμματος όσο κάνω μια δουλειά με τον πόρο πχ μέχρι να ψηθούν οι πίτσες sleep(T_bake). Όταν τελειωσω με τους πόρους που κάνω χρήση κανω pthread_mutex_unlock για να τους ξανα βαλω στους διαθέσιμους πόρους δηλ (N_cooks++) (N_oven=N_oven+pizzas) και επίσης pthread_cond_signal για να ξεμπλοκαρει ένα από τα νήματα που ίσως περιμένουν για διαθέσιμο πόρο.

- Οπου χρειάζεται παραπάνω χρησιμοποιώ την συνάρτηση clock_gettime (CLOCK_REALTIME, &χ) προκειμένου να πάρω την ακριβή στιγμή χρόνου που θα χρειαστώ σύμφωνα με την εκφώνηση . προσοχη! Αλλαζω το χ.tv_sec προκειμένου να παρώ τα λεπτά και όχι τα δευτερόλεπτα.
- Οπου θελω να εμφανισω ζητούμενο μυνημα από την εκφωνηση της ακσης κλειδωνω το mutex της οθονης rc = pthread_mutex_lock(&mutex_screen); Το εμφανιζω και μετα το ξεκλειδωνω rc=pthread_mutex_unlock(&mutex_screen);
- Τελος απελευθερωνω το νημα pthread_exit(NULL);

ΠΑΡΑΤΗΡΗΣΕΙΣ

* ΠΡΟΣΟΧΗ στην εργασία μου έχω χρησιμοποιήσει την εντολη

```
pthread_mutex_t mutex_tel = PTHREAD_MUTEX_INITIALIZER;
```

```
pthread_cond_t cond_pack = PTHREAD_COND_INITIALIZER;
```

Για τα mutexes και τα conditions ΩΣΤΟΣΟ αυτή εντολη δεν τρεχει στην

παλια εκδοση του νm μονο την καινουργια για την παλια όπως διαπιστωσαμε με τον κβυλωμένο στις ωρες γράφειου.Με την παλια θα επρεπε να ειχα κανει

ως global

```
pthread_mutex_t lock;
```

```
pthread_cond_t cond;
```

Και μεσα στην main

```
: pthread_mutex_init(&lock, NULL);
```

```
pthread_cond_init(&cond, NULL);
```

*Προσθετα στοιχεία δεν υπάρχουν ακολούθησα πλήρως την εκφώνηση τα εργαστήρια και τις υποδείξεις

* το προγραμμα μου εμφανιζει max average χρονο αναμονης 0.00(μεχρι ο πελατης να ερθει σε επικοινωνία με ένα τηλεφωνητη)όταν εχω αριθμο N_TEL=3 ενώ όταν N_TEL=1 μου εμφανιζει max=2 και avg = 0.66333

* Σε εμενα ο κωδικας τρεχει κανονικα και εμφανιζει όλα τα ζητουμενα μυνηματα **χωρις καποιο error** και 2 warnings.

* επιπλεον αρχικα ημουν συνεραγτης με τον Δημήτρη Καββαδα αμ 3190064 ο οποιος τελικα αποχώρησε από την ομαδα αλλα ειχε τον κωδικα μου μεχρι ένα σημειο . το αναφέρω σε περίπτωση εντοπισμού του συστήματος για αντιφραφη .

Καλο καλοκαιριuuuu !!!



Με σχόλια [GP1]: Οι αντιστοιχες εντολες για την παλια εκδοση του vm