

1^η Εργασία Λειτουργικά Συστήματα, 2019-2020

Παναγιώτης Παναγιώτου, 3180139

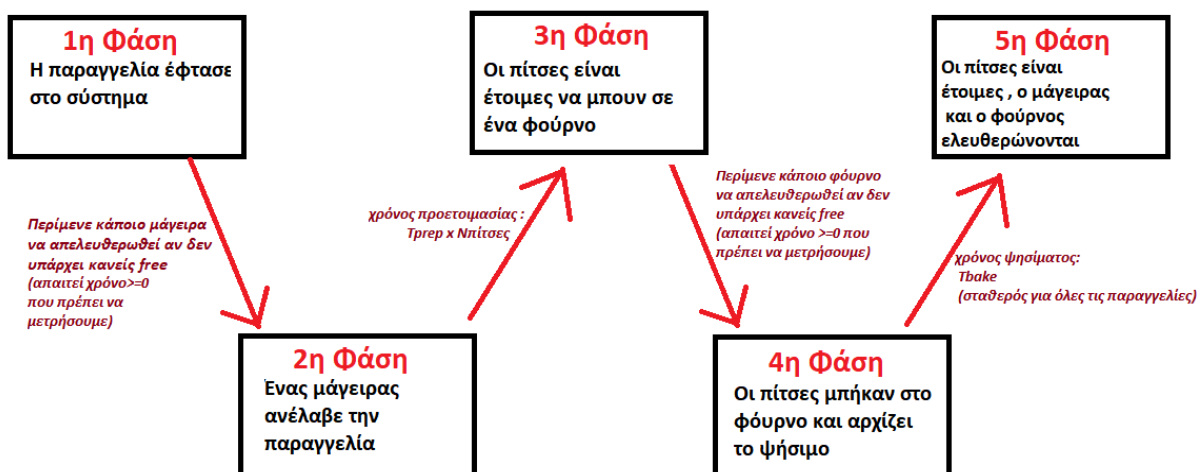
Ειρήνη Χρυσικοπούλου, 3180208

1) Γενική Ιδέα :

Για την εργασία αυτή χρειάστηκε να προσομοιώσουμε ένα σύστημα παραγγελιών μίας πιτσαρίας χρησιμοποιώντας **pthread** νήματα (**POSIX**). Κάθε νήμα μας αναπαριστούσε μία παραγγελία. Χρησιμοποιήσαμε την τεχνική του αμοιβαίου αποκλεισμού (**Mutual Exclusion**) για το συγχρονισμό των νημάτων και τη σωστή διαχείριση μνήμης, κάνοντας τα κατάλληλα **lock/unlock** σε **mutexes**, για την κατάλληλη ενημέρωση των τιμών των μεταβλητών.

Η γενική ιδέα ήταν να χωρίσουμε την κάθε παραγγελία σε 5 φάσεις/στάδια, όπως φαίνεται στο παρακάτω σχήμα. Με αυτόν τον τρόπο καταφέραμε επιτυχώς να διακρίνουμε τους χρόνους όπου έπρεπε τα νήματα μας να περιμένουν με χρήση **pthread_cond_t** μεταβλητών και της εντολής **sleep()**. Επίσης, χρήσιμη ήταν η χρήση των **struct timespec**, από τα οποία μπορέσαμε να αντλήσουμε το συνολικό χρόνο που πήρε μια παραγγελία.

Τα στάδια μιας παραγγελίας



2) Δομή Κώδικα:

Στο "**3180139-3180208-pizza1.h**" δηλώσαμε τις βιβλιοθήκες που θα χρησιμοποιήσουμε, τις σταθερές μεταβλητές και τις συναρτήσεις μας, δηλαδή ότι αφορά τον προεπεξεργαστή.

Στο "**3180139-3180208-pizza1.c**" αρχικά δηλώσαμε τα ονόματα και τους τύπους των μεταβλητών που χρησιμοποιούμε σε όλη την πορεία του προγράμματος, τα **mutexes** κάποιων από των μεταβλητών αυτών και φυσικά το **mutex** της οθόνης για να γίνεται σωστά η εκτύπωση αποτελεσμάτων.

Ξεκινώντας τη **main**, ελέγχεται ότι το πλήθος ορισμάτων που έδωσε ο χρήστης ισούται με δύο (αριθμός πελατών, σπόρος) και ότι ο αριθμός πελατών (**Ncust**) είναι ακέραιος αριθμός. Αν δεν ικανοποιούνται αυτές οι συνθήκες το πρόγραμμα τερματίζει και εμφανίζει μήνυμα λάθους.

Αν όλα πάνε καλά, αρχικοποιούμε τις απαραίτητες μεταβλητές, έναν πίνακα με τα νήματα που θα δημιουργήσουμε αργότερα και έναν πίνακα με το (μοναδικό) **id** κάθε πελάτη.

Δημιουργούμε νήματα όσα οι πελάτες που θα εξυπηρετηθούν με την **pthread_create**, βάζοντας την ***Order** ως την ρουτίνα που θα εκτελέσει το νήμα και το **id** του πελάτη σαν παράμετρο της ***Order**.

Αφού κάθε νήμα αναπαριστά μια παραγγελία, αμέσως πριν τη δημιουργία του θα καλέσουμε τη **sleep()**, με όρισμα το χρόνος άφιξης της στο κατάστημα, που θα πάρουμε από την **rand_r** με διαφορετικό σπόρο κάθε φορά (για να είμαστε **thread safe**).

Για κάθε νήμα στην ***Order**:

1^η ΦΑΣΗ:

- Καλούμε την **rand_r** (με διαφορετικό σπόρο κάθε φορά για να είμαστε **thread safe** και να έχουμε διαφορετικά αποτελέσματα σε κάθε τρέξιμο του προγράμματος) για να παράξει τον αριθμό των πιτσών της παραγγελίας (μέσα στα όρια που πρέπει. Δηλαδή 1 έως 5 το πολύ)
- Καλούμε την **clock_gettime** για να κρατήσουμε την ώρα άφιξης της παραγγελίας στο κατάστημα.

2^η ΦΑΣΗ:

- Ελέγχουμε αν υπάρχουν διαθέσιμοι παρασκευαστές. Αν δεν υπάρχουν περιμένουμε μέχρι κάποιος παρασκευστής να γίνει διαθέσιμος.
- Όταν βρεθεί διαθέσιμος παρασκευαστής ενημερώνουμε την σχετική μεταβλητή (**A_cooks**) μειώνοντας τη, καθώς ένας από αυτούς στην επόμενη φάση θα ασχοληθεί με την παραγγελία.
- Χρησιμοποιούμε **mutex** για να ενημερώσουμε την μεταβλητή και για να εξασφαλίσουμε ότι μόνο ένα νήμα την φορά μπορεί να αλλάξει την τιμή της μεταβλητής.

3^η ΦΑΣΗ:

- Ο παρασκευαστής προετοιμάζει τις πίτσες της παραγγελίας που του ανατέθηκε για να μπουν σε ένα φούρνο, ο χρόνος που παίρνει αυτό είναι $T_{prep} \times N_{πίτσες}$.

4^η ΦΑΣΗ:

- Ελέγχουμε αν υπάρχουν διαθέσιμοι φούρνοι. Αν δεν υπάρχουν περιμένουμε μέχρι κάποιος φούρνος να γίνει διαθέσιμος.
- Όταν βρεθεί διαθέσιμος φούρνος ενημερώνουμε την σχετική μεταβλητή (**A_ovens**) καθώς στην επόμενη φάση θα ψήσουμε τις πίτσες της παραγγελίας σε έναν από αυτούς.
- Χρησιμοποιούμε **mutex** για να ενημερώσουμε την μεταβλητή για να διαφαλίσουμε ότι μόνο ένα νήμα την φορά μπορεί να αλλάξει την τιμή της μεταβλητής.

5^η ΦΑΣΗ:

- Ψήνονται όλες οι πίτσες της παραγγελίας στον ίδιο φούρνο. (Κρατάει σταθερό χρόνο T_{bake}).
- Μετά το ψήσιμο η παραγγελία έχει ολοκληρωθεί οπότε αποδεσμεύουμε τον φούρνο και τον παρασκευαστή (Παλί χρησιμοποιούμε **mutex**).
- Καλούμε την **clock_gettime** για να κρατήσουμε την ώρα ολοκλήρωσης της παραγγελίας.
- Υπολογίζουμε τον συνολικό χρόνο που χρειάστηκε η παραγγελία .
- Εκτυπώνουμε τα αποτελέσματα σχετικά με την παραγγελία στην οθόνη. Χρησιμοποιούμε **mutex** και για την εκτύπωση των δεδομένων προκειμένου μόνο ένα νήμα κάθε φορά να μπορεί να τυπώσει, για να μη γίνει μπλέξιμο γραμμών.
- Ενημερώνουμε τις μεταβλητές μέσου και συνολικού χρόνου.
- Τέλος καλούμε την **pthread_exit** για να κάνουμε έξοδο από το νήμα.

Σε κάθε περίπτωση που γίνει κάποιο λάθος σχετικό με τα νήματα, έχουμε κάνει μια συνάρτηση , τη **check_rc(int rc)**, να διαβάξει το response code και να κάνει **exit** αν χρειαστεί.

Τα νήματα έπειτα κάνουν **join** στη **main** και εκτυπώνονται τα τελικά αποτελέσματα.