

Λειτουργικά Συστήματα εργασία 1

<u>Μέλη ομάδας</u>	<u>ΑΜ</u>
Δημήτριος Τσιομπίκας	3180223
Όλγα Μαργέλη	3180103
Ελένη Φουρτούνη	3180196

Δομή Κώδικα

Έχουμε υλοποιήσει 2 αρχεία , το p3180223-p3180103-p3180196-pizza1.h και το p3180223-p3180103-p3180196-pizza1.c . Στο header file έχουμε ορίσει στον προεπεξεργαστή τις μεταβλητές Ncook,Noven,Torderlow,Torderhigh,Norderlow,Norderhigh,Tprep και Tbakе. Επίσης έχουμε δηλώσει την συνάρτηση order που θα χρησιμοποιηθεί στο C File.

Στο C File, έχουμε δηλώσει τις εξής global variables : ένα mutex με όνομα lock για την οθόνη ,ένα mutex το cookLock για τους μάγειρες, ένα mutex το ovenLock για τους φούρνους, ένα condition variable με όνομα cond ,2 ακέραιες μεταβλητές την cookC και την ovenC που χρησιμοποιούνται για τα mutexes των μαγείρων και φούρνων αντίστοιχα, 2 ακέραιες μεταβλητές την cooks και την ovens που χρησιμοποιούνται για να μπορούμε να προσθαφαιρέσουμε το πλήθος παρασκευαστών και φούρνων στη συνάρτηση , τον αριθμό των πελατών Ncust, έναν πίνακα ακεραίων (pizzaCount) μίας θέσης που αργότερα θα γίνει πίνακας Ncust θέσεων , τον χρειαζόμαστε και στη

main και στη συνάρτηση, 3 ακέραιες μεταβλητές την max που θα χρησιμοποιηθεί για το μέγιστο χρόνο αναμονής και τις avg,sum που θα χρησιμοποιηθούν για το μέσο χρόνο αναμονής.

Συνάρτηση Order

Η συνάρτηση order χρησιμοποιείται για την εξυπηρέτηση των παραγγελιών και για την εύρεση του μέσου και του μέγιστου χρόνου αναμονής. Παίρνει ένα όρισμα, το οποίο είναι το id της παραγγελίας ώστε να ξέρουμε ποια εξυπηρετείται. Έχουμε ορίσει 2 μεταβλητές τύπου struct timespec την start και την end για τη μέτρηση του χρόνου.

Μόλις εκτυπωθεί το μήνυμα ότι η παραγγελία έχει ληφθεί, αρχίζει να μετράει ο χρόνος με την clock_gettime. Εδώ επίσης κλειδώνουμε το mutex cookLock για να υπάρχει η εξής σχέση : 1 παραγγελία άνα 1 μάγειρα (1 νήμα άνα ένα παρασκευαστή) έτσι ώστε όταν και οι 6 μάγειρες εξυπηρετούν από μία παραγγελία οι υπόλοιπες παραγγελίες να μπλοκάρονται μέχρι να τελειώσει ένας από τους μάγειρες με την παραγγελία του (εδώ χρησιμοποιείται και η cond_wait).

Αφού μία παραγγελία πάρει έναν μάγειρα το mutex ξεκλειδώνεται, εκτυπώνεται μήνυμα για το πλήθος των πιτσών που θέλει η παραγγελία και αρχίζει η ετοιμασία τους (η ετοιμασία των εξυπηρετούμενων παραγγελιών γίνεται ταυτόχρονα). Αμέσως μετά το mutex ovenLock κλειδώνει ώστε να υπάρχει η σχέση : 1 παραγγελία άνα ένα φούρνο.

Μόλις και οι 5 φούρνοι ψήνουν από μία παραγγελία, ο τελευταίος μάγειρας θα μπλοκαριστεί (με χρήση της cond_wait) ώσπου να τελειώσει ένα ψήσιμο από κάποιον άλλο μάγειρα. Αφού αρχίσει και το

ψήσιμο , η παραγγελία ψήνεται για 10 δευτερόλεπτα (Αν οι παραγγελίες που θα ψηθούν είναι παραπάνω από μία το ψήσιμο γίνεται ταυτόχρονα).

Τέλος κλειδώνει το mutex lock για την εκτύπωση των μηνυμάτων και τον υπολογισμό του χρόνου αναμονής. Εδώ χρησιμοποιείται η end με την clock_gettime και τελειώνει η μέτρηση του χρόνου. Εκτυπώνεται το μήνυμα ότι η παραγγελία εξυπηρετήθηκε επιτυχώς στα αντίστοιχα λεπτά , προσθέτονται πάλι ένας φούρνος και ένας μάγειρας , χρησιμοποιείται η signal για να είναι έτοιμη η cond variable για την επόμενη παραγγελία και ξεκλειδώνεται το mutex lock.

Main

Στην main παίρνουμε τα 2 ορίσματα από τον χρήστη , τον αριθμό των πελατών (Ncust) και τον σπόρο των τυχαίων αριθμών. Χρησιμοποιούμε το srand(time(NULL)) για το seeding , κάνουμε initialize τα mutex και το condition variable , δημιουργούμε έναν πίνακα ακεραίων Ncust θέσεων και βάζουμε Ncust θέσεις στον πίνακα pizzaCount.

Ξεκινάμε μία επανάληψη από 0 έως Ncust στην οποία γεμίζουμε το pizzaCount με τυχαίους αριθμούς στο διάστημα [Norderlow,Norderhigh] , γεμίζουμε το id , δημιουργούμε τα threads και κάθε φορά δημιουργούμε έναν τυχαίο αριθμό στην ακέραιο μεταβλητή orderTime στο διάστημα [Torderlow,Torderhigh] , ώστε να κάνει sleep για τόσα «λεπτά» μέχρι να δωθεί η επόμενη παραγγελία.

Τέλος , φτιάχνουμε άλλο ένα for loop από 0 έως Ncust που κάνει join τα threads ώστε να εκτελείται η συνάρτηση, καταστρέφουμε το condition variable και τα mutex και εκτυπώνουμε τον μέγιστο και τον μέσο χρόνο αναμονής.