Αναφορά 1ης Εργασίας στο μάθημα Λειτουργικών Συστημάτων

Μέλη: Στελλάτου Χαρά (p3170225), Μεϊδάνης Στέφανος (p3170107)

Το πρόγραμμα που χρειάζεται να υλοποιήσουμε αφορά σε μια πιτσαρία από την οποία οι πελάτες παραλαμβάνουν την παραγγελία από το κατάστημα. Στο παρόν πρόγραμμα αποφασίσαμε να υλοποιήσουμε δυο συναρτήσεις. Η πρώτη και πιο σημαντική είναι η order και η δεύτερη είναι η main.

Αρχικά δηλώνουμε τα απαραίτητα mutexes και conditions. Τα mutexes είναι τέσσερα και χρησιμοποιούνται για τους μάγειρες (cookLock), τους φούρνους (ovenLock), τη δέσμευση της οθόνης (screenLock) και για την καταγραφή του χρόνου(timeLock). Τα conditions από την άλλη είναι δύο και αφορούν στους μάγειρες(cookCond) και στους φούρνους (ovenCond), τα χρησιμοποιούμε για να ξέρουμε πότε υπάρχουν και πότε δεν υπάρχουν διαθέσιμοι μάγειρες και φούρνοι.

Στη συνέχεια, παραθέτουμε τις ολικές μεταβλητές που χρειάζονται για να παίρνουμε το χρόνο(sumTime, maxTime), καθώς και εκείνες που μετράν τους διαθέσιμους πόρους του καταστήματος(cooks,ovens) για να γνωρίζουμε τον διαθέσιμο αριθμό τους κάθε φορά.

Αναλύοντας τη main , αφού ελέγξουμε ότι έχουν δοθεί οι σωστές παράμετροι από τον χρήστη(argc=3, argv(1)>0) , δηλώνουμε και αρχικοποιούμε τη μεταβλητή seedp για να αξιοποιηθεί στη rand_r. Έπειτα δεσμεύουμε δυναμική μνήμη για να αποθηκευτούν τα νέα threads και αρχικοποιούμε τα mutexes και τα conditions.Δημιουργούμε τα threads σε τυχαίες χρονικές στιγμές ενώ παράλληλα δίνουμε τιμές στο odArray οι οποίες περιλαμβάνουν το id (παραγγελία) ως δεκάδες και τον αριθμό των πιτσών που ζητούνται σαν το τελευταίο ψηφίο (πχ. Id =10, αριθμός πιτσών= 5 => od= 105).

Προσθέτουμε και τη Join, η οποία αναγκάζει το πρόγραμμα να περιμένει να ολοκληρωθούν τα threads ένα προς ένα πριν τελειώσει το πρόγραμμα.

Τέλος, η main εκτυπώνει το μέγιστο χρόνο προετοιμασίας και το μέσο όρο προετοιμασίας των παραγγελιών, καταστρέφουμε τα mutexes, τα conditions, και ελευθερώνουμε την δεσμευμένη τη μνήμη μας.

Όσον αφορά στην order,

Αφού αρχικοποιήσουμε την μεταβλητή od (περιέχει τις πληροφορίες της παραγγελίας από τον odArray) τυπώνουμε μήνυμα αποδοχής της παραγγελίας. Προκειμένου να μην υπάρξει κάποια παρεμβολή, κλειδώνουμε τον mutex της οθόνης πριν εμφανίσουμε την καινούρια παραγγελία, αφού εμφανιστεί τον ξεκλειδώνουμε. Η διαδικασία αυτή πραγματοποιείται κάθε φορά που χρειάζεται να εκτυπωθεί κάποιο μήνυμα. Έπειτα για κάθε παραγγελία ξεκινάμε το χρόνο να μετράει. Προχωρούμε σε έναν έλεγχο για να εξετάσουμε αν υπάρχουν διαθέσιμοι μάγειρες. Αν υπάρχουν μάγειρες, προχωρούμε στο πρόγραμμα μας και μειώνονται οι διαθέσιμοι μάγειρες κατά ένα. Έπειτα περιμένουμε από την παραγγελία να ετοιμαστεί. Στη συνέχεια, χρειάζεται να ελέγξουμε αν υπάρχουν και διαθέσιμοι φούρνοι οπότε κάνουμε μια παρόμοια διαδικασία με αυτήν για τους μάγειρες, δηλαδή αν υπάρχουν διαθέσιμοι φούρνοι, μειώνουμε πάλι τον αριθμό των διαθέσιμων φούρνων,

πραγματοποιείται το ψήσιμο των πιτσών της παραγγελίας και αφού τελειώσει το ψήσιμο , απελευθερώνονται και ο φούρνος και ο μάγειρας, καθώς και τα αντίστοιχα mutexes και conditions.

Τέλος, σταματάμε το ρολόι και υπολογίζουμε το χρόνο που χρειάστηκε για την παραγγελία, εμφανίζουμε την παραγγελία και το χρόνο που χρειάστηκε αυτή για να ολοκληρωθεί και μετά ενημερώνουμε τις μεταβλητές sumTime, maxTime και το νήμα τερματίζει.