

Όνοματεπώνυμο:Καρράς Κωνσταντίνος Α.Μ.:3180076

Όνοματεπώνυμο:Πέππα Χριστίνα Α.Μ.:3180154

Στην αρχή του κώδικα κάνουμε import τις απαραίτητες βιβλιοθήκες και το header αρχείο, αρχικοποιούμε 2 global μεταβλητές που θα χρειαστούμε για να κρατήσουμε τον αριθμό από τους μάγειρες και τους φούρνους, 2 μεταβλητές για τον μέσο και τον μέγιστο χρόνο ένα δείκτη σε double που θα δημιουργήσουμε δυναμικό πίνακα, μία μεταβλητή για τον σπόρο, 4 mutexes(για φούρνους, μάγειρες, σπόρο, οθόνη) και 2 condition variables(για ενεργοποίηση παραγγελίας που περιμένει φούρνο ή μάγειρα).

Στη main συνάρτηση ελέγχουμε αν τα ορίσματα που έχει δώσει ο χρήστης είναι 3 και αν ο αριθμός των πελατών είναι θετικός ακέραιος. Εκχωρούμε τις τιμές εφόσον είναι σωστές, δημιουργούμε τον πίνακα των threads και κάνουμε initialize τα mutexes και τα condition variables. Ορίζουμε το μέγεθος του δυναμικού πίνακα βάσει του αριθμού των πελατών και δημιουργούμε τα νήματα με παράμετρο την συνάρτηση order και τον αριθμό της παραγγελίας(id[i] = i + 1, ώστε να ξεκινάει ο αριθμός από 1). Προηγουμένως βάζουμε το ίδιο for να ΜΗΝ δημιουργεί ακαριαία όλες τις παραγγελίες μαζί αλλά, πέρα από την 1η που δεν έχει καμία καθυστέρηση, όλες οι υπόλοιπες περιμένουν για ένα τυχαίο διάστημα από [1,5] (χρησιμοποιώντας την (τοπική) μεταβλητή seed1 = seed για να μην κλειδώνουμε το mutex)(το seed είναι global και περιέχει την τιμή που δίνει ο χρήστης). Ύστερα περιμένουμε όλα τα νήματα να τελειώσουν και τρέχουμε ένα for loop όσος και ο αριθμός των πελατών για να βγάλουμε τα συγκεντρωτικά αποτελέσματα για τον μέσο και μέγιστο χρόνο ολοκλήρωσης των παραγγελιών. Τέλος, κλειδώνουμε την οθόνη και εμφανίζουμε τα αντίστοιχα μηνύματα, απελευθερώνουμε τον χώρο που έχουμε δεσμεύσει δυναμικά και καταστρέφουμε τα mutexes και τα condition variables.

Στην συνάρτηση order ορίζουμε το start και το finish προκειμένου να γνωρίζουμε την στιγμή που άρχισε και τελείωσε η εκάστοτε παραγγελία και κλειδώνουμε έναν από τους μάγειρες. Όσο δεν υπάρχει κανένας μάγειρας εμφανίζουμε μήνυμα ότι η παραγγελία είναι μπλοκαρισμένη και το βάζουμε να περιμένει. Όταν βγει από το while loop τότε σημαίνει ότι βρήκε διαθέσιμο μάγειρα, μειώνουμε λοιπόν τους μάγειρες και ξεκλειδώνουμε το mutex. Χρησιμοποιούμε μία νέα τοπική μεταβλητή το seed2 στο οποίο εκχωρούμε το seed + id(για μεγαλύτερη τυχαιότητα) ώστε να πάρουμε ένα τυχαίο αριθμό, και το βάζουμε να "πέσει για ύπνο" για ένα τυχαίο χρονικό διάστημα από 1-5(βάσει του τυχαίου αριθμού). Μετά κάνουμε για τους φούρνους ότι ακριβώς και για τους μάγειρες. Στη συνέχεια, το βάζουμε "για ύπνο" για χρόνο Ttake και μέχρι τότε δεν αποδεσμεύουμε κάποιο μάγειρα, αφού ο μάγειρας αποδεσμεύεται εφόσον ψηθούν οι πίτσες που έχει αναλάβει. Εφόσον περάσει από το sleep τότε κλειδώνουμε τους μάγειρες, τους αυξάνουμε κατά έναν, κάνουμε signal σε περίπτωση που κάποια παραγγελία περιμένει παρασκευαστή και ύστερα τους ξεκλειδώνουμε. Ακριβώς αντίστοιχη είναι στη συνέχεια και η υλοποίηση για τους φούρνους. Ύστερα, κλειδώνουμε την οθόνη και παίρνουμε τον χρόνο ώστε να βγάλουμε την συνολική διάρκεια της παραγγελίας και να την εκχωρήσουμε στο αντίστοιχο κελί του timer(id - 1 εφόσον στη main περνούσαμε σαν όρισμα το id + 1), για να βγάλουμε στη main τον μέσο και μέγιστο χρόνος ολοκλήρωσης. Τέλος, εμφανίζουμε το μήνυμα:"Η παραγγελία με αριθμό <oid> ετοιμάστηκε σε <X> λεπτά", αποδεσμεύουμε την οθόνη και βγαίνουμε από το thread χωρίς να επιστρέφουμε κάτι.