**int main(int argc, char \*argv[]):**

Κάνουμε έναν αρχικό έλεγχο για να σιγουρευτούμε ότι έχουμε τον σωστό αριθμό ορισμάτων. Χρησιμοποιούμε τις μεταβλητές Ncust και seed για να αποθηκεύσουμε τα ορίσματα (αριθμός πελατών και σπόρος) που δίνει ο χρήστης, ενώ έχουμε και μια μεταβλητή rc η οποία αξιοποιείται στα πλαίσια του ελέγχου των system calls. Έπειτα δεσμεύουμε χώρο για 2 πίνακες id και threads τα θα περιέχουν τα threads και τα ids τους και κάνουμε initialize όλα τα mutexes και condition variables που χρησιμοποιούμε. Μετά έχουμε ένα for loop στο οποίο δημιουργούνται τα threads ( και ξεκινάει το κάθε ένα να εκτελεί την συνάρτηση order). Αντίστοιχα υπάρχει ένα for loop για το join των threads με την main αφού τερματίσουν, ενώ ακολουθεί η καταστροφή των mutexes και condition variables, εκτύπωση των στατιστικών στοιχείων και η απελευθέρωση του χώρου που δεσμεύσαμε για τους πίνακες id και threads.

**void \* order(void \*x):**

Αρχικοποιούμε την μεταβλητή id με το id του thread, καθώς ορίζουμε και κάποιες άλλες μεταβλητές που θα χρησιμοποιηθούν για τα sleeps (wait\_time) και τα στατιστικά, καθώς και την rc που λειτουργεί όπως και πριν . Χρησιμοποιούμε ένα if else statement(το οποίο κάνουμε και lock λόγω της κοινόχρηστης μεταβλητής s) για να υπολογίσουμε το αρχικό wait\_time πριν πάρουμε τηλέφωνο. Μετά μπαίνουμε σε αναμονή για έναν τηλεφωνητή, οπότε χρησιμοποιούμε mutex lock και condition variables.

Αφού λάβουμε τον τηλεφωνητή επιλέγουμε έναν τυχαίο αριθμό στο διάστημα [Norderlow, Norderhigh] για τον αριθμό πιτσών. Έπειτα προσθέτουμε στα variables margaritas, pepperoni και special +1 ανά πίτσα ανάλογα με το είδος της, πράγμα που προκύπτει με βάση τις πιθανότητες Pm, Pp, Ps (και κυρίως τις 2 πρώτες αφού την 3η την βγάζουμε και με μια απλή αφαίρεση με το 100 ) και περιμένουμε ένα συγκεκριμένο χρόνο wait\_time (υπολογίστηκε προσθέτοντας τον ελάχιστο χρόνο που χρειάζεται, με ένα τυχαιοποιημένο αριθμό , βασισμένος στο υπόλοιπο της ακέραιας διαίρεσης ενος τυχαίο αριθμού και του διαστήματος [Tpaymentlow,Tpaymenthigh], με την βοήθεια του seed). Ύστερα, επιλέγουμε τυχαία, πάλι με την βοήθεια του seed, ένα αριθμό και αν αυτός είναι μεγαλύτερος από την Pfail τότε προσθέτουμε, μέσα σε mutex, τα έσοδα από τις πίτσες στην καθολική μεταβλητή revenue, καθώς και τις καθολικές μεταβλητές που μετράνε πόσες πίτσες πουλήθηκαν από κάθε κατηγορία (σε άλλο mutex όμως), διαφορετικά αυξάνουμε κατά 1 το πλήθος των αποτυχημένων παραγγελιών, αποδεσμεύεται ο τηλεφωνητής, ξυπνάμε κάποιο άλλο νήμα, και το νήμα τερματίζει. Τέλος, αποδεσμεύεται ο τηλεφωνητής με τον οποίο επικοινωνούσε ο συγκεκριμένος πελάτης και ειδοποιούμε ένα άλλο νήμα το οποίο περίμενε για κάποιο διαθέσιμο τηλεφωνητή ως τώρα.

Έχει τελειώσει η πληρωμή, οπότε το νήμα, με condition variable, περιμένει να το ξυπνήσει κάποιο άλλο νήμα για να ξεκινήσει η διαδικασία ψησίματος. Αν βρεί διαθέσιμο μάγειρα, μειώνουμε κατά 1 την καθολική N\_cooks, δείχνοντας πως δεσμεύεται ο μάγειρας. Περιμένουμε να ετοιμαστούν όλες οι πίτσες και ελέγχουμε, πάλι με condition variables, για διαθέσιμους φούρνους. Αφού βρεθούν τόσοι διαθέσιμοι φούρνοι όσες οι πίτσες που έχουμε παραγγείλει, αποδεσμεύουμε τον μάγειρα (προσθέτουμε κατά 1) και καταλαμβάνουμε τους φούρνους ( αφαιρούμε, από το N\_ovens, τόσους φούρνους όσους χρειαζόμαστε). Περιμένουμε Tbake χρόνο, δηλαδή τον χρόνο που χρειάζεται να ψηθεί μία πίτσα, γιατί όλες ψήνονται ταυτόχρονα.

Έχουν ψηθεί οι πίτσες σε αυτό το σημείο, οπότε περιμένουμε να εμφανιστεί ένας ντελιβεράς (με κάποιο σήμα) για να παραλάβει τις πίτσες (όσο περιμένουμε, οι πίτσες παραμένουν στον φούρνο). Μόλις λάβουμε σήμα, δεσμεύουμε τον ντελιβερά και αμέσως μετά αποδεσμεύουμε τους φούρνους. Κάτι που αξίζει να σημειωθεί εδώ, είναι πως αντί να ειδοποιήσουμε μόνο ένα νήμα που περιμένει για φούρνους, τα ειδοποιούμε όλα (με broadcast), καθώς δεν θέλουμε να κάνουμε πολλά signal wake up σε περίπτωση που το νήμα που ξυπνήσαμε θέλει περισσότερους φούρνους από τους διαθέσιμους. Περιμένουμε τώρα να πακεταριστούν όλες οι πίτσες και όταν γίνει αυτό τελειώνει η διαδικασία ολοκλήρωσης της παραγγελίας. Τέλος, υπολογίζουμε με μία αφαίρεση, πόσο χρόνο διήρκησε η διαδικασία (**order\_end.tv\_sec - order\_start.tv\_sec**).

Μετά το πακετάρισμα της παραγγελίας, ακολουθεί η διανομή της. Επομένως, περιμένουμε χρόνο wait\_time, για να φτάσει η παραγγελία στον προορισμό της (υπολογίστηκε προσθέτοντας τον ελάχιστο χρόνο που χρειάζεται, με ένα τυχαιοποιημένο αριθμό , βασισμένος στο υπόλοιπο της ακέραιας διαίρεσης ενος τυχαίο αριθμού και του διαστήματος [Tdellow,Tdelhigh], με την βοήθεια του seed). Τελειώνει ο χρόνος αυτός, δηλαδή η παραγγελία έφτασε. Άρα με τις **overall\_end.tv\_sec**, **overall\_start.tv\_sec** και **cold\_end.tv\_sec**, **cold\_start.tv\_se**, υπολογίζουμε τις **o\_time** (χρόνος εξυπηρέτησης) και **c\_time** (χρόνος κρυώματος πιτσών) μεταβλητές, που με την σειρά τους χρειάζονται για να αυξήσουμε τις καθολικές μεταβλητές των συνολικών χρόνων **εξυπηρέτησης** (**order\_waiting\_total**), και **κρυώματος** **πιτσών** (**cold\_waiting\_total**). Μάλιστα, συγκρίνουμε τα **o\_time** και **max\_order**, σε περίπτωση που έχουμε νέα μεγάλη (σε χρόνο) παραγγελία. Στην περίπτωση που έχουμε,αλλάζουμε την καθολική μεταβλητή **max\_order** με τον νέο μεγαλύτερο χρόνο. Ακόμη, υπολογίζουμε τον χρόνο delivery, ώστε να τον εμφανίσουμε.

Ως τελευταίο βήμα, πρέπει να έρθει ο ντελιβεράς πίσω στο μαγαζί. Για αυτό περιμένουμε τον ίδιο χρόνο που περιμέναμε για να φτάσει η παραγγελία. Μόλις φτάσει, απελευθερώνουμε τον ντελιβερά και ειδοποιούμε ένα νήμα. Αφού έχουν γίνει τα παραπάνω, χωρίς κάποιο σφάλμα, τερματίζεται το νήμα.