ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

Υποχρεωτικό Μάθημα 4ου εξαμήνου

Εαρινό Εξάμηνο 2022-2023

Προγραμματιστική Εργασία

ΜΕΛΗ ΟΜΑΔΑΣ:

ΟΝ/ΜΟ: ΓΕΩΡΓΟΠΟΥΛΟΥ ΚΩΝΣΤΑΝΤΙΝΑ, ΑΜ: 3180029

ΟΝ/ΜΟ: ΣΤΑΥΡΟΥΛΑΚΗ ΝΙΚΟΛΙΤΣΑ-ΜΑΡΙΑ, ΑΜ: 3150266

ΟΝ/ΜΟ: ΣΤΑΥΡΟΥΛΑΚΗ ΜΑΡΙΑ, ΑΜ: 3160168

Επεξήγηση Προγράμματος

Στην συγκεκριμένη εργασία, κατασκευάσαμε ένα πρόγραμμα που προσομοιώνει ένα σύστημα παραγγελιών, παρασκευής και διανομής πίτσας με την χρήση του πακέτου νημάτων POSIX, μέσω της βιβλιοθήκης pthread. Το πρόγραμμά μας κάνει συχνή χρήση νημάτων, mutexes (Mutual excusion) και conditions, για τον βέλτιστο συγχρονισμό τους, δίνοντάς μας την δυνατότητα να ελευθερώσουμε και να κοιμίσουμε τις παραγγελιές ανάλογα με το αν ικανοποιείται κάποια συνθήκη ή όχι αντίστοιχα. Η οποία είναι να υπάρχει τουλάχιστον ένας παρασκευαστής ο οποίος θα ετοιμάσει τις πίτσες για να ψηθούν. Με την ίδια διαδικασία να υπάρχουν τουλάχιστον τόσοι φούρνοι όσες είναι και οι πίτσες. Αντίστοιχα, αφού ολοκληρωθούν τα παραπάνω, να υπάρχει τουλάχιστον ένας υπάλληλος πακεταρίσματος ο οποίος θα βγάλει τις πίτσες από τον φούρνο και θα τις πακετάρει. Και τέλος, να είναι διαθέσιμος τουλάχιστον ένας διανομέας, ο οποίος χρειάζεται τον ίδιο χρόνο που θα κάνει για να παραδώσει την παραγγελία και για να γυρίσει, ώστε να ελευθερωθεί κι αυτός με την σειρά του τελευταίος. Κλείνοντας το πρόγραμμα, υπολογίζονται και εμφανίζονται ορισμένες πληροφορίες για την κάθε παραγγελία και ορισμένα στατιστικά από όλες τις παραγγελίες μαζί.

Δομή Πηγαίου Κώδικα

Αρχικά, για να ξεκινήσει το πρόγραμμα των παραγγελιών, απαιτείται είσοδος δύο παραμέτρων από το χρήστη, και σε περίπτωση δεν τηρηθεί ο αριθμός αυτός εκτυπώνεται αντίστοιχο μήνυμα και τερματίζει. Το πρώτο όρισμα αφορά τον αριθμό των πελατών, ο οποίος θα καθορίσει και τον αριθμό των νημάτων που θα δημιουργηθούν, ώστε κάθε

πελάτης να εξυπηρετείται από ένα νήμα. Το δεύτερο όρισμα αφορά το seed, που χρησιμεύει στο πρόγραμμα για την επιλογή τυχαίων δεδομένων μέσω της μεθόδου random(). Για την είσοδο γίνεται έλεγχος για το αν είναι έγκυρη ώστε να συνεχίσει το πρόγραμμα, σε άλλη περίπτωση τυπώνει μήνυμα και τερματίζει.

Χρησιμοποιούνται πίνακες threads, για την αποθήκευση των αντικειμένων τύπου thread, και threaded, για το αναγνωριστικό id τους. Μέσω της συνάρτησης malloc() δεσμεύεται δυναμικά μνήμη για αυτούς τους πίνακες. Οι συναρτήσεις pthread_mutex_init() και pthread_cond_init() χρησιμοποιούνται για να αρχικοποιήσουν τα αντικείμενα mutex και condition αντίστοιχα. Τα mutex χρησιμοποιούνται για τον έλεγχο της πρόσβασης σε κοινόχρηστους πόρους από διάφορα νήματα, ενώ τα condition χρησιμοποιούνται για τον συγχρονισμό και τον έλεγχο της εκτέλεσης των νημάτων.

Στο βρόγχο του for, ο οποίος επαναλαμβάνεται για κάθε πελάτη ώστε να δημιουργεί ένα νήμα για τον καθένα, αρχικά γεμίζει τον πίνακα threaded και ύστερα, μέσω της pthread_create, δημιουργεί ένα νήμα και συνδέει τη συνάρτηση Order με αυτό. Η δημιουργία του νήματος αποθηκεύεται στη μεταβλητή rc, ώστε να γίνεται έλεγχος σφαλμάτων κατά τη δημιουργία του νήματος. Η συνάρτηση sleep καθυστερεί τον επόμενο πελάτη για χρονικό διάστημα ανάμεσά σε Torderlow και Torderhigh. Στη μεταβλητή status αποθηκεύεται η κατάσταση του κάθε νήματος και μέσω της pthread_join αναστέλλεται η εκτέλεση του νήματος που καλεί τη συνάρτηση μέχρι το νήμα να ολοκληρωθεί, δηλαδή η pthread_join να επιστρέψει 0. Αν δεν επιστραφεί 0 σημαίνει ότι έγινε σφάλμα και τερματίζει.

Στην Order, η οποία έχει κληθεί από τη pthread_create για έναν πελάτη τη φορά, αρχικά δηλώνει τις απαιτούμενες μεταβλητές που αφορούν τους χρόνους, startorder, finishorder, startcolding και finishcolding και αρχικοποιεί ordertime και coldtime με 0. Επίσης αρχικοποιεί μεταβλητές που αφορούν τον αριθμό από πίτσες που επιθυμεί ο πελάτης, δηλαδή τη num_pizzas που υπολογίζεται μέσω της Random και επιστρέφει αριθμό ανάμεσα σε Norderlow και Norderhigh, μετά ανάλογα με την πιθανότητα Pplain μέσω της Propability υπολογιζεται το πλήθος των απλών πιτσών και των σπέσιαλ και αποθηκεύεται στις μεταβλητές tmp_plain και tmp_spec αντίστοιχα. Τέλος υπολογίζεται ο χρόνος παρασκευής και πακεταρίσματος ανάλογα με τον αριθμό από πίτσες της παραγγελίας και του Tprep και Tpack αντίστοιχα.

Ξεκινά ο χρόνος με τη clock_gettime, ώστε να προσδιοριστούν στο τέλος τα απαιτούμενα ερωτήματα. Ύστερα, καλείται η sleep που κοιμίζει το πρόγραμμα για χρόνο μεταξύ Tpaymentlow και Tpaymenthigh, αναπαριστώντας το χρόνο που απαιτείται για την πληρωμή της παραγγελίας. Η πληρωμή έχει πιθανότητα Pfail να αποτύχει, η οποία προσδιορίζεται από την Propability και σε κάθε περίπτωση, αποτυχίας ή μη, ενημερώνονται οι αντίστοιχες μεταβλητές sumFail και sumSucc που μετράνε το πλήθος των αποτυχημένων και επιτυχημένων παραγγελιών. Στην πρώτη περίπτωση το νήμα επιστρέφει και ενημερώνεται ο χρήστης για την αποτυχία της συγκεκριμένης παραγγελίας με τη βοήθεια του αναγνωριστικού threadid. Στην άλλη περίπτωση, κρατούνται τα συνολικά έσοδα όλου του προγράμματος στην μεταβλητή totalRev, προστίθενται στο συνολικό αριθμό των παραγγελιών απλής πίτσας, sumPlain, ο αριθμός από απλές πίτσες που

ζητήθηκε σε αυτή την παραγγελία και στο στο συνολικό αριθμό των παραγγελιών σπέσιαλ πίτσας, sumSpec, ο αριθμός από σπέσιαλ πίτσες αυτής της παραγγελίας.

Στη συνέχεια για να εξασφαλιστεί ότι μόνο ένα νήμα έχει πρόσβαση σε κάποιο κομμάτι κώδικα χρησιμοποιείται η pthread_mutex_lock. Αρχικά εφαρμόζεται στο mutexCook, όπου αν έχει κλειδωθεί από άλλο νήμα, το αρχικό νήμα μέσω της pthread_mutex_lock μπλοκάρεται, μέχρι να ελευθερωθεί από το άλλο νήμα. Αν δεν έχει κλειδωθεί από άλλο τότε το κλειδώνει το αρχικό νήμα ώστε να διασφαλίσει ότι μόνο αυτό θα έχει πρόσβαση στον κώδικα. Η αρχική συνθήκη που πρέπει να ικανοποιηθεί για να ξεκινήσει η διαδικασία είναι να υπάρχει τουλάχιστον ένας παρασκευαστής, που ο αριθμός τους διατηρείται στη μεταβλητή AvailableCooks, ο οποίος θα αναλάβει μία παραγγελία και θα ετοιμάσει τις πίτσες για να ψηθούν. Αν δεν υπάρχουν, το νήμα καλεί τη συνάρτηση pthread_cond_wait και μπλοκάρεται, μέχρις ότου να δεχτεί σήμα condCook. Όσο αναμένει για παρασκευαστές το mutexCook ξεκλειδώνει για να χρησιμοποιηθεί από άλλο νήμα. Μόλις λάβει σήμα για υπάρχοντες παρασκευαστές τότε ενημερώνεται ανάλογα η μεταβλητή AvailableCooks. Μέσω της pthread_mutex_lock απελευθερώνεται το mutexCook για να χρησιμοποιηθεί από άλλο νήμα. Η διαδικασία καθυστερεί μέχρι να ετοιμαστούν οι πίτσες από τον παρασκευαστή, καλώντας την sleep για όσο χρόνο υπολογίστηκε παραπάνω ότι απαιτείται για να προετοιμαστούν οι πίτσες της παραγγελίας, δηλαδή για num_prep.

Παρόμοια διαδικασία ακολουθείται και για το mutexOven, το οποίο δεσμεύεται από το νήμα για να αποτρέψει άλλα νήματα να εκτελέσουν το κομμάτι του κώδικα. Ύστερα ακολουθεί έλεγχος για τη διαθεσιμότητα των φούρνων, στη μεταβλητή AvailableOvens, ώστε να υπάρχουν τουλάχιστον τόσοι φούρνοι όσες είναι και οι πίτσες της παραγγελίας. Αν δεν είναι διαθέσιμος ο αριθμός των φούρνων που απαιτείται μέσω της pthread_cond_wait μπλοκάρεται το νήμα αυτό και απελευθερώνεται το mutexOven για να μπορέσει να χρησιμοποιηθεί από άλλο νήμα. Όταν γίνει διαθέσιμος ο απαιτούμενος αριθμός φούρνων ενημερώνεται η μεταβλητή AvailableOvens και ξεκλειδώνει το mutexOven με την pthread_mutex_unlock. Όταν οι πίτσες είναι έτοιμες για να μπουν για ψήσιμο τότε αποδεσμεύεται ο παρασκευαστής και αρχικά κλειδώνει το mutexCook με την pthread_mutex_lock, ενημερώνει τη μεταβλητή AvailableCooks με το πλήθος των παρασκευαστών, δίνει σήμα condCook για τη διαθεσιμότητα παρασκευαστή μέσω της pthread_cond_signal και ξεκλειδώνει το mutexCook με την pthread_mutex_unlock για να χρησιμοποιηθεί από άλλο νήμα. Μετά καλείται η sleep για να περιμένει μέχρι να ψηθούν οι πίτσες, δηλαδή για χρόνο Tbake.

Αφού ψηθούν οι πίτσες, μέσω της clock_gettime ξεκινά ο χρόνο καταγραφής για την ώρα που κρυώνει η παραγγελία. Μετά μέσω της pthread_mutex_lock κλειδώνεται το mutexPacker και ελέγχεται με τη μεταβλητή AvailablePackers, αν υπάρχει τουλάχιστον ένας υπάλληλος πακεταρίσματος ο οποίος θα βγάλει τις πίτσες από τον φούρνο και θα τις πακετάρει. Αν δεν υπάρχει η pthread_cond_wait μπλοκάρει το νήμα και απελευθερώνει το mutexPacker. Όταν βρεθεί υπάλληλος πακεταρίσματος ενημερώνεται η μεταβλητή AvailablePackers, ώστε να γίνει γνωστό ότι απασχολείται ένας υπάλληλος και το mutexPacker ξεκλειδώνει με την pthread_mutex_unlock. Μετά από αναμονή για το πακετάρισμα χρόνου num_pack με τη sleep, δεσμεύεται το mutexPacker με τη

pthread_mutex_lock ώστε να ενημερωθεί η μεταβλητή AvailablePackers ότι τελείωσε τη δουλεία του ο υπάλληλος που δεσμεύτηκε πριν, δίνεται σήμα με τη pthread_cond_signal για το condPacker ώστε να ενημερώσει για τη διαθεσιμότητα επιπλέον υπαλλήλου και απελευθερώνεται το mutexPacker με τη pthread_mutex_unlock. Όμοια γίνεται η διαδικασία και για την ενημέρωση διαθεσιμότητας των φούρνων και ξεκλειδώματος του αντίστοιχου mutex του.

Συνεχίζοντας, κλειδώνεται με την pthread_mutex_lock το mutexDeliverer ώστε να γίνει ο τελευταίος έλεγχος που αφορά το αν είναι διαθέσιμος τουλάχιστον ένας διανομέας με τη μεταβλητή AvailableDeliverers. Αν δεν είναι μπλοκάρεται με την pthread_cond_wait και περιμένει για σήμα ενημέρωσης στο condDeliverer. Όταν υπάρχει διαθέσιμος διανομέας ενημερώνεται η μεταβλητή που αφορά το πλήθος τους και ξεκλειδώνεται το mutexDeliverer με την pthread_mutex_unlock ώστε να χρησιμοποιηθεί και από άλλα νήματα. Ο χρόνος παράδοσης είναι time_delivery, που υπολογίζεται με τη Random και βρίσκεται μεταξύ Tdellow και Tdelhigh. Μετά τη χρήση της sleep για time_delivery χρόνο, η παραγγελία έχει παραδοθεί και σταματάνε τα ρολόγια καταγραφής του χρόνου κρυώματος και όλη της διαδικασίας εξυπηρέτησης της παραγγελιάς με την clock_gettime. Για να αποδεσμευτεί ο διανομέας απαιτείται η κλήση της sleep πάλι για time_delivery χρόνο καθώς χρειάζεται τον ίδιο χρόνο που θα κάνει για να παραδώσει την παραγγελία και για να γυρίσει. Με την pthread_mutex_lock κλειδώνεται το mutexDeliverer ώστε να ενημερωθεί η AvailableDeliverers, δίνεται σήμα διαθεσιμότητας condDeliverer με την pthread_cond_signal και ξεκλειδώνεται το mutexDeliverer με την pthread_mutex_unlock.

Κλείνοντας την Order, υπολογίζονται και εμφανίζονται πληροφορίες για την κάθε παραγγελία, δηλαδή ο χρόνος εξυπηρέτησης της παραγγελίας ordertime και ο χρόνος κρυώματός της coldtime, και γίνονται και οι ανάλογοι υπολογισμοί για τον μέγιστο χρόνο εξυπηρέτησης και κρυώματος παραγγελιάς και για τους μέσους όρους τους. Το νήμα αυτής της παραγγελίας παύει να υπάρχει μετά την pthread_exit.

Όταν γίνει η ολοκλήρωση για όλα τα νήματα, δηλαδή όταν εξυπηρετηθούν όλες οι παραγγελίες, τυπώνονται τα συνολικά έσοδα totalRev, ο αριθμός των απλών και σπέσιαλ πιτσών, sumPlain και sumSpec αντίστοιχα, ο αριθμών επιτυχημένων και αποτυχημένων παραγγελιών, sumSucc και sumFail αντίστοιχα, ο μέσος όρος για εξυπηρέτηση παραγγελιών avOrder και κρυώματος avCold, και ο μέγιστος χρόνος εξυπηρέτησης maxOrder και κρυώματος maxCold. Τέλος με την pthread_mutex_destroy καταστρέφονται όλα τα mutexes που χρησιμοποιήθηκαν, δηλαδή mutexCook, mutexOven, mutexPacker, mutexDeliverer και mutexScreen, και με την pthread_cond_destroy καταστρέφονται όλα τα cond που χρησιμοποιήθηκαν, δηλαδή τα condCook, condPacker, condOven και condDeliverer και απελευθερώνεται ο χώρος που δεσμεύτηκε από τη malloc για τα threads και threadid μέσω της free.

Περιορισμοί και Πρόσθετα Χαρακτηριστικά

Για την υλοποίηση του προγράμματος ακολουθήθηκαν οι οδηγίες της εκφώνησης. Έχουν μόνο προστεθεί έλεγχοι κατά την κλήση pthread_mutex_lock, pthread_mutex_unlock και

pthread_mutex_destroy για τα mutexes, pthread_cond_wait, pthread_cond_signal και pthread_cond_destroy για τα cond, και clock_gettime για το χρόνο, ώστε να εκτυπώνεται ένα μήνυμα σφάλματος και το πρόγραμμα τερματίζεται με έξοδο -1 σε περίπτωση αποτυχίας υλοποίησης της αντίστοιχης διαδικασίας.