

## ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

### Υποχρεωτικό Μάθημα 4<sup>ου</sup> εξαμήνου

Εαρινό Εξάμηνο 2022-2023

### Προγραμματιστική Εργασία

**Σκοπός:** Στην εργασία σας θα υλοποιήσετε ένα σύστημα παραγγελιών και διανομής πίτσας με χρήση του πακέτου νημάτων POSIX threads (pthreads). Στο σύστημα αυτό οι παραγγελίες δίνονται και πληρώνονται ηλεκτρονικά, και στη συνέχεια κάθε παραγγελία προετοιμάζεται, ψήνεται και τελικά διανέμεται στον πελάτη. Στα συστήματα αυτά έχουμε μεγάλο αριθμό παραγγελιών οι οποίες εξυπηρετούνται από περιορισμένο αριθμό σημείων εξυπηρέτησης, συνεπώς το πρόγραμμά σας πρέπει να υλοποιεί αμοιβαίο αποκλεισμό (με mutexes) και συγχρονισμό (με μεταβλητές συνθήκης). Ο κώδικάς σας θα πρέπει να λειτουργεί σωστά στην εικονική μηχανή που διατίθεται στα CSLAB και στο διαδίκτυο (βλ. το wiki του eclass). Η εργασία είναι ομαδική και είναι σχεδιασμένη για ομάδες δύο (2) ή τριών (3) ατόμων. Ο βαθμός της εργασίας θα είναι το 30% του τελικού βαθμού (δηλαδή, 3 μονάδες στις 10).

**Αντικείμενο:** Η πιτσαρία φτιάχνει μόνο δύο είδη πίτσας, απλή και σπέσιαλ. Διαθέτει ηλεκτρονικό σύστημα παραγγελιών και πληρωμών,  $N_{\text{cook}}$  παρασκευαστές,  $N_{\text{oven}}$  φούρνους,  $N_{\text{packer}}$  υπάλληλους πακεταρίσματος και  $N_{\text{deliverer}}$  διανομείς. Ο πρώτος πελάτης συνδέεται στο σύστημα παραγγελιών τη χρονική στιγμή 0, και κάθε επόμενος πελάτης συνδέεται μετά από ένα τυχαίο ακέραιο χρονικό διάστημα στην περιοχή  $[T_{\text{orderlow}}, T_{\text{orderhigh}}]$ . Όταν ένας πελάτης συνδεθεί με το σύστημα παραγγελιών, παραγγέλλει έναν τυχαίο ακέραιο αριθμό από πίτσες στο διάστημα  $[N_{\text{orderlow}}, N_{\text{orderhigh}}]$ . Κάθε μία από αυτές έχει πιθανότητα  $P_{\text{plain}}$  να είναι απλή, αλλιώς είναι σπέσιαλ. Το σύστημα απαιτεί ένα τυχαίο ακέραιο χρονικό διάστημα στην περιοχή  $[T_{\text{paymentlow}}, T_{\text{paymenthigh}}]$  για να χρεώσει την πιστωτική κάρτα του πελάτη. Με πιθανότητα  $P_{\text{fail}}$  η χρέωση αποτυγχάνει και η παραγγελία ακυρώνεται, αλλιώς η παραγγελία καταχωρείται, τα έσοδα του καταστήματος αυξάνονται κατά  $C_{\text{plain}}$  ή  $C_{\text{special}}$  ευρώ ανά πίτσα, ανάλογα με τις πίτσες της παραγγελίας και, τέλος, ενημερώνουμε το πλήθος πωλήσεων ανά τύπο πίτσας. Η καταχωρημένη παραγγελία αρχικά περιμένει μέχρι ένας παρασκευαστής να γίνει διαθέσιμος. Όταν συμβεί αυτό, ο παρασκευαστής χρειάζεται  $T_{\text{prep}}$  χρόνο για να ετοιμάσει κάθε πίτσα. Στη συνέχεια, ο παρασκευαστής περιμένει μέχρι να γίνουν αρκετοί φούρνοι διαθέσιμοι, ώστε όλες οι πίτσες να ψηθούν παράλληλα, γιατί κάθε φούρνος χωράει μία μόνο πίτσα. Όταν αρκετοί φούρνοι είναι διαθέσιμοι, κάθε πίτσα μπαίνει σε έναν φούρνο και ο παρασκευαστής αναλαμβάνει την επόμενη παραγγελία. Οι πίτσες ψήνονται για χρόνο  $T_{\text{bake}}$ . Όταν ολοκληρωθεί το ψήσιμο, οι φούρνοι κλείνουν αυτόματα και περιμένουν έναν υπάλληλο πακεταρίσματος να βγάλει τις πίτσες από τους φούρνους και να τις πακετάρει, πράγμα που παίρνει χρόνο  $T_{\text{pack}}$  ανά πίτσα. Σε αυτή τη στιγμή και οι φούρνοι απελευθερώνονται. Όταν ένας διανομέας γίνει διαθέσιμος, παραλαμβάνει μία πακεταρισμένη παραγγελία και την παραδίδει στον πελάτη, με την παράδοση να διαρκεί ένα τυχαίο ακέραιο χρονικό διάστημα στην περιοχή  $[T_{\text{dellow}}, T_{\text{delhigh}}]$ . Μετά την παράδοση, ο διανομέας χρειάζεται το ίδιο ακριβώς διάστημα ώστε να επιστρέψει στο κατάστημα και να αναλάβει την επόμενη παραγγελία. **Προσοχή:** κάθε παρασκευαστής ασχολείται με μία μόνο παραγγελία μέχρι να την βάλει στο φούρνο, οι φούρνοι αποδεσμεύονται μόνο όταν τελειώσει το πακετάρισμα, και κάθε διανομέας μεταφέρει μία μόνο παραγγελία σε κάθε διαδρομή.

**Είσοδος και δεδομένα:** Οι ακόλουθες σταθερές θα ορίζονται σε ένα αρχείο δηλώσεων:

- $N_{\text{cook}}=2$  παρασκευαστές
- $N_{\text{oven}}=15$  φούρνοι

- $N_{\text{packer}}=2$  υπάλληλοι πακεταρίσματος
- $N_{\text{deliverer}}=10$  διανομείς
- $T_{\text{orderlow}}=1$  λεπτό
- $T_{\text{orderhigh}}=3$  λεπτά
- $N_{\text{orderlow}}=1$  πίτσες
- $N_{\text{orderhigh}}=5$  πίτσες
- $P_{\text{plain}}=60\%$
- $T_{\text{paymentlow}}=1$  λεπτό
- $T_{\text{paymenthigh}}=3$  λεπτά
- $P_{\text{fail}}=10\%$
- $C_{\text{plain}}=10$  ευρώ
- $C_{\text{special}}=12$  ευρώ
- $T_{\text{prep}}=1$  λεπτό
- $T_{\text{bake}}=10$  λεπτά
- $T_{\text{pack}}=1$  λεπτό
- $T_{\text{dellow}}=5$  λεπτά
- $T_{\text{delhigh}}=15$  λεπτά

Το πρόγραμμά σας θα δέχεται δύο (ακριβώς) παραμέτρους με το πλήθος των πελατών προς εξυπηρέτηση,  $N_{\text{cust}}$ , και έναν τυχαίο σπόρο για τη γεννήτρια των τυχαίων αριθμών.

**Έξοδος εργασίας:** Για κάθε παραγγελία, θα τυπώνονται τουλάχιστον τα παρακάτω μηνύματα στην οθόνη στις κατάλληλες χρονικές στιγμές:

- Η παραγγελία με αριθμό <oid> απέτυχε / καταχωρήθηκε. [Τη στιγμή αποτυχίας ή καταχώρησης της παραγγελίας.]
- Η παραγγελία με αριθμό <oid> ετοιμάστηκε σε <X> λεπτά. [Τη στιγμή που ολοκληρώνεται το πακετάρισμα μιας επιτυχημένης παραγγελίας.]
- Η παραγγελία με αριθμό <oid> παραδόθηκε σε <Y> λεπτά. [Τη στιγμή που ολοκληρώνεται η παράδοση μιας επιτυχημένης παραγγελίας.]

Η σειρά των γραμμών θα είναι τυχαία, αλλά οι γραμμές δεν πρέπει να μπλέκονται μεταξύ τους. Το <X> συμβολίζει το διάστημα από την εμφάνιση του πελάτη μέχρι το πακετάρισμα της παραγγελίας, ενώ το <Y> το διάστημα από την εμφάνιση του πελάτη μέχρι την παράδοση της παραγγελίας. Στο τέλος της εκτέλεσης, το σύστημα θα τυπώνει τα ακόλουθα:

- Τα συνολικά έσοδα από τις πωλήσεις, πόσες πίτσες πουλήθηκαν από κάθε τύπο και το πλήθος επιτυχημένων και αποτυχημένων παραγγελιών.
- Το μέσο και μέγιστο χρόνο εξυπηρέτησης των πελατών (από τη στιγμή που εμφανίζεται ο πελάτης, μέχρι να παραδοθεί η παραγγελία) – **μόνο για τις επιτυχημένες παραγγελίες. !**
- Το μέσο και μέγιστο χρόνο κρυώματος των παραγγελιών (από τη στιγμή που ολοκληρώνεται το ψήσιμο μέχρι τη στιγμή που παραδίδεται η παραγγελία) – **μόνο για τις επιτυχημένες παραγγελίες.**

**Δομή κώδικα:** Το αρχικό νήμα του προγράμματός σας θα δημιουργεί ένα νήμα ανά πελάτη/παραγγελία (συνολικά  $N_{\text{cust}}$  νήματα) στο οποίο θα μεταβιβάζετε έναν αριθμό νήματος (από 1 έως  $N_{\text{cust}}$ ) ώστε να τον χρησιμοποιείτε ως αριθμό παραγγελίας. Κάθε νήμα στη συνέχεια θα εκτελεί τα παραπάνω βήματα μέχρι να ολοκληρωθεί η παραγγελία και θα τυπώνει την κατάλληλη έξοδο. Τέλος, **το αρχικό νήμα θα τυπώνει την τελική έξοδο.** Θα χρειαστείτε τουλάχιστον τα ακόλουθα:

- Μία ακέραιη μεταβλητή κι ένα mutex για να μετράτε το πλήθος των διαθέσιμων παρασκευαστών και μία μεταβλητή συνθήκης για να συγχρονίσετε τις παραγγελίες με τους παρασκευαστές, έτσι ώστε όταν δεν υπάρχουν διαθέσιμοι παρασκευαστές να μπλοκάρονται οι παραγγελίες. Με ανάλογο τρόπο θα πρέπει να χειριστείτε τους φούρνους, τους υπαλλήλους πακεταρίσματος και τους διανομείς.
- Μεταβλητές και τα σχετικά mutex για τα έσοδα και τα στατιστικά.

- Ένα mutex για κλείδωμα της οθόνης όταν τυπώνετε την έξοδο.

#### Υποδείξεις:

- Για να ολοκληρώνεται σε λογικό χρονικό διάστημα το πρόγραμμά σας, αντιμετωπίστε όλους τους χρόνους που δίνονται σαν δευτερόλεπτα αντί για λεπτά.
- Κατά τη μεταγλώττιση πρέπει να δίνετε την επιλογή `-pthread` για να χρησιμοποιηθεί η βιβλιοθήκη POSIX threads.
- Για να προσομοιώσετε το χρόνο που διαρκεί κάποιο διάστημα (π.χ. ψησίματος) θα χρησιμοποιήσετε την `unsigned int sleep(unsigned int seconds)`.
- Για να παράγετε μία σειρά ψευδοτυχαίων αριθμών, θα χρησιμοποιήσετε την `int rand_r(unsigned int *seedp)`. Χρησιμοποιήστε τον τελεστή `%` για να περιορίσετε το εύρος τιμών των τυχαίων αριθμών.
- Για τον υπολογισμό των χρόνων αναμονής, χρησιμοποιείτε την `int clock_gettime(clockid_t clk_id, const struct timespec *tp)` στην αρχή και τέλος κάθε πράξης, με πρώτη παράμετρο τη σταθερά `CLOCK_REALTIME`. Στην έξοδο, μετατρέψτε τους χρόνους που προκύπτουν από δευτερόλεπτα σε λεπτά.
- Χρησιμοποιήστε έναν βρόχο με `while` για να ελέγχετε τη συνθήκη αναμονής και να κάνετε `pthread_cond_wait` για όσες φορές χρειάζεται.
- Προσοχή στο ότι μπορεί να χρειαστεί να δεσμεύσουμε πολλούς φούρνους για μία παραγγελία.
- Προσοχή στο πότε δεσμεύεται και αποδεσμεύεται κάθε πόρος (παρασκευαστής, φούρνος, υπάλληλος πακεταρίσματος και διανομέας).
- Προσοχή στο να τερματίζετε σωστά τα νήματα, να τα περιμένετε όπου χρειάζεται και (κυρίως!) να απελευθερώνετε σωστά όση μνήμη δεσμεύετε.!

**Παραδοτέα:** Ο κώδικάς σας πρέπει να αποτελείται από ένα αρχείο με δηλώσεις (συμπεριλαμβανομένων των σταθερών) και ένα αρχείο κώδικα C για το πρόγραμμα. Τα αρχεία αυτά πρέπει να έχουν ονόματα της μορφής `p3x-p3y-p3z-pizzeria.h` για τις δηλώσεις, `p3x-p3y-p3z-pizzeria.c` για τον κώδικα C, όπου και `p3x-p3y-p3z` είναι οι αριθμοί μητρώου σας. Εκτός από τον κώδικα, θα πρέπει να γράψετε μία αναφορά η οποία να περιγράφει τη δομή του κώδικά σας και να αναφέρει τυχόν περιορισμούς ή πρόσθετα χαρακτηριστικά που έχετε υλοποιήσει. Η αναφορά πρέπει να είναι ένα αρχείο σε μορφή PDF με όνομα της μορφής `p3x-p3y-p3z-pizzeria.pdf`. Τέλος, θα πρέπει να συμπεριλάβετε ένα αρχείο με όνομα `test-res.sh` το οποίο θα μεταγλωττίζει και θα εκτελεί το πρόγραμμά σας με παραμέτρους 100 πελάτες και αρχικό σπόρο 1000. Αυτά τα τέσσερα αρχεία (και τίποτα άλλο) θα πρέπει να συμπιεστούν σε ένα αρχείο σε μορφή zip με όνομα της μορφής `p3x-p3y-p3z-pizzeria.zip` και να υποβληθούν από ένα μόνο μέλος της ομάδας μέσω της υποβολής εργασιών του eclass.

**Προθεσμία υποβολής:** Τα συμπιεσμένα αρχεία με τις εργασίες σας θα πρέπει να παραδοθούν μέσω του eclass μέχρι την Παρασκευή 9/6/2023 και ώρα 23:59.

**Βαθμολόγηση και εξέταση:** Εργασίες με ομοιότητες που υποδεικνύουν αντιγραφή *θα μηδενιστούν* όλες (θα γίνει έλεγχος με ειδικό πρόγραμμα). Δειγματοληπτικά, θα κληθούν ορισμένες ομάδες για προφορική εξέταση. Μέλη ομάδων, ή και ολόκληρες ομάδες, που δεν θα προσέλθουν στην προφορική εξέταση, *θα μηδενιστούν*. Η βαθμολόγηση εξετάζει τα εξωτερικά χαρακτηριστικά της εργασίας (ονοματολογία αρχείων, αναφορά, μεταγλώττιση χωρίς σφάλματα, κανονική ολοκλήρωση), την ποιότητα του κώδικα (σωστή δόμηση, σχολιασμός, έλεγχος για σφάλματα) και την υλοποίηση των απαιτήσεων (κλειδώματα και συνθήκες, υπολογισμός στατιστικών, έξοδος).