

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

Υποχρεωτικό Μάθημα 4^{ου} εξαμήνου

Εαρινό Εξάμηνο 2019-2020

1^η Προγραμματιστική Εργασία

Σκοπός: Στις εργασίες σας θα υλοποιήσετε ένα σύστημα παραγγελιών και διανομής πίτσας με χρήση του πακέτου νημάτων POSIX threads (pthreads). Στο σύστημα αυτό εισάγονται παραγγελίες, κάθε παραγγελία ετοιμάζεται και στο τέλος γίνεται η διανομή στον πελάτη. Στην πρώτη εργασία θα κατασκευάσετε ένα απλουστευμένο σύστημα, το οποίο στη δεύτερη εργασία θα γίνει πιο σύνθετο. Στα συστήματα αυτά έχουμε έναν μεγάλο αριθμό παραγγελιών οι οποίοι εξυπηρετούνται από έναν 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. ανακοίνωση στο eclass). Η εργασία είναι ομαδική και είναι σχεδιασμένη για ομάδες τριών (3) ατόμων, μπορείτε όμως να την υλοποιήσετε και σε μικρότερες ομάδες. Ο βαθμός της 1^{ης} εργασίας θα είναι το 20% του τελικού βαθμού (δηλαδή, 2 μονάδες στις 10).

Αντικείμενο: Η πιτσαρία διαθέτει N_{cook} παρασκευαστές και N_{oven} φούρνους. Στην 1^η εργασία, η πιτσαρία δεν κάνει διανομή, δηλαδή οι πελάτες παραλαμβάνουν μόνο από το κατάστημα. Η πρώτη παραγγελία γίνεται τη χρονική στιγμή 0, και κάθε επόμενη παραγγελία γίνεται μετά από ένα τυχαίο ακέραιο χρονικό διάστημα στην περιοχή $[T_{\text{orderlow}}, T_{\text{orderhigh}}]$. Κάθε παραγγελία περιλαμβάνει έναν τυχαίο ακέραιο αριθμό από πίτσες στο διάστημα $[N_{\text{orderlow}}, N_{\text{orderhigh}}]$. Η παραγγελία αρχικά περιμένει μέχρι ένας παρασκευαστής να γίνει διαθέσιμος. Όταν συμβεί αυτό, ο παρασκευαστής χρειάζεται T_{prep} χρόνο να ετοιμάσει κάθε πίτσα. Στη συνέχεια, ο παρασκευαστής περιμένει μέχρι ένας φούρνος να γίνει διαθέσιμος. Όταν συμβεί αυτό, όλες οι πίτσες της παραγγελίας μπαίνουν στον ίδιο φούρνο και ψήνονται για χρόνο T_{bake} , ενώ ο παρασκευαστής παρακολουθεί τον φούρνο. Όταν ολοκληρωθεί το ψήσιμο, ο παρασκευαστής αφήνει την έτοιμη παραγγελία στον πάγκο παράδοσης και απελευθερώνεται. **Ο παρασκευαστής ασχολείται με μία μόνο παραγγελία σε κάθε στιγμή, από την παραλαβή ως την ολοκλήρωσή της.**

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

- $N_{\text{cook}}=6$ παρασκευαστές
- $N_{\text{oven}}=5$ φούρνοι
- $T_{\text{orderlow}}=1$ λεπτό
- $T_{\text{orderhigh}}=5$ λεπτά
- $N_{\text{orderlow}}=1$ πίτσες
- $N_{\text{orderhigh}}=5$ πίτσες
- $T_{\text{prep}}=1$ λεπτό
- $T_{\text{bake}}=10$ λεπτά

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

Έξοδος εργασίας: Για κάθε παραγγελία, τη στιγμή της παραλαβής θα τυπώνεται στην οθόνη ένα μήνυμα της μορφής:

- **Η παραγγελία με αριθμό <oid> ετοιμάστηκε σε <X> λεπτά.**

Η σειρά των γραμμών θα είναι τυχαία, αλλά οι γραμμές δεν πρέπει να μπλέκονται μεταξύ τους. Ο χρόνος <X> συμβολίζει το διάστημα από την εισαγωγή της παραγγελίας μέχρι την ολοκλήρωσή της. Στο τέλος της εκτέλεσης, το σύστημα θα τυπώνει τα ακόλουθα:

- Μέσο και μέγιστο χρόνο ολοκλήρωσης των παραγγελιών.

Δομή κώδικα: Το αρχικό νήμα του προγράμματός σας θα δημιουργεί ένα νήμα ανά παραγγελία (συνολικά N_{cust} νήματα) στα οποία θα μεταβιβάζετε έναν αριθμό νήματος (από 1 έως N_{cust}) ώστε να τα διακρίνετε. 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 και 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.
- Μία ακέραιη μεταβλητή και ένα `mutex` για να μετράτε το πλήθος των διαθέσιμων φούρνων και μία μεταβλητή συνθήκης για να συγχρονίσετε τις παραγγελίες με τους φούρνους, έτσι ώστε όταν δεν υπάρχουν διαθέσιμοι φούρνοι να μπλοκάρονται οι παραγγελίες.
- Πραγματικές μεταβλητές και τα σχετικά `mutex` για τους χρόνους ολοκλήρωσης.
- Ένα `mutex` για κλείδωμα της οθόνης όταν τυπώνετε την έξοδο.

Προσοχή στο να τερματίζετε σωστά τα νήματα, να τα περιμένετε όπου χρειάζεται και (κυρίως!) να απελευθερώνετε σωστά όση μνήμη δεσμεύετε.

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)`. Με αυτή τη μέθοδο μπορείτε να δημιουργείτε και τις παραγγελίες από το αρχικό νήμα.
- Για να παράγετε μία σειρά ψευδοτυχαίων αριθμών, θα χρησιμοποιήσετε την `int rand_r(unsigned int *seedp)`. Δίνοντας διαφορετικούς αριθμούς στο σπόρο, θα έχετε διαφορετικές ακολουθίες εκτέλεσης. 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 οι παρασκευαστές είναι απασχολημένοι έως το τέλος του ψησίματος.

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-pizza1.h` για τις δηλώσεις, `p3x-p3y-p3z-pizza1.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-pizza1.pdf`. Τέλος, θα πρέπει να συμπεριλάβετε ένα αρχείο με όνομα `test-res1.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-pizza1.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 τη Δευτέρα 4/5/2020 και ώρα 23:59.

Βαθμολόγηση και εξέταση: Αρχεία που δεν θα έχουν την *ακριβή* ονοματολογία που ζητείται παραπάνω *θα μηδενιστούν*. Εργασίες με ομοιότητες που υποδεικνύουν αντιγραφή *θα μηδενιστούν* όλες (θα γίνει έλεγχος με ειδικό πρόγραμμα). Δειγματοληπτικά, θα κληθούν ορισμένες ομάδες για προφορική εξέταση στα CSLAB. Μέλη ομάδων, ή και ολόκληρες ομάδες, που δεν θα προσέλθουν στην προφορική εξέταση, *θα μηδενιστούν*.