



Λειτουργικά Συστήματα

2022 - 2023

2η Εργαστηριακή Άσκηση

Σκοπός

Σκοπός της εξαμηνιαίας εργαστηριακής άσκησης (project) είναι να γνωρίσουν, να κατανοήσουν και να εξοικειωθούν οι φοιτητές με όσα θέματα πραγματεύεται το μάθημα και να αναγνωρίσουν τις δυνατότητες της συνεργασίας σε ομάδες. Τις ομάδες θα έχετε την ευκαιρία να τις επιλέξετε εσείς και να τις οργανώσετε όπως θέλετε. Η εμπειρία εργασίας σε ομάδες και η συνεργασία είναι σημαντική δεξιότητα που δεν πρέπει να αμελήσετε να καλλιεργήσετε γιατί, όπως θα διαπιστώσετε ως απόφοιτοι, είναι σημαντικό προσόν ενός μηχανικού.

Ομάδα Project

Οι ομάδες project θα πρέπει να αποτελούνται από 1 έως 4 άτομα, και προτείνεται να είναι ίδια η σύνθεση των ομάδων στις δύο ασκήσεις.

Παραδοτέα Project

Τα παραδοτέα κάθε ομάδας είναι:

- Η αναφορά του project της ομάδας, σε μορφή pdf.
- Τα αρχεία του κώδικα της υλοποίησης σε συμπιεσμένη μορφή zip. **Στην αρχή του αρχείου του κώδικα, σε σχόλια, θα τοποθετείτε τα ονοματεπώνυμα και τους ΑΜ των μελών της ομάδας ως εξής:**

```
/* Spiridon Sioutas, 2345678 */  
/* Christos Makris, 4567890 */  
/* Panagiotis Hadjidoukas, 3456789 */  
/* Aristidis Ilias, 1234567 */
```

Τα παραδοτέα θα υποβληθούν μέσω του eclass στις «Εργασίες», από το ίδιο μέλος της ομάδας που είχε παραδώσει τα παραδοτέα του πρώτου project.

Θα παραδοθούν:

- η αναφορά σε pdf στην εργασία «2ο project 2022-2023, Αναφορά Ομάδων»
- τα αρχεία του κώδικα σας σε μορφή zip στην εργασία «2ο project 2022-2023, Κώδικας Ομάδων»

Στο περιεχόμενο της αναφοράς της κάθε ομάδας, η τεκμηρίωση είναι απαραίτητη και θα αξιολογηθεί. Στο εξώφυλλο της αναφοράς, πρέπει να περιλαμβάνει τα ονόματα των μελών της ομάδας, τους αριθμούς μητρώου και το email τους.

Θα αποτελείται από μια σύντομη περιγραφή του σχεδιασμού της υλοποίησης κάθε ομάδας, χρησιμοποιώντας διαγράμματα η/και ψευδοκώδικα όπου θεωρείτε απαραίτητο.

Επίσης, θα περιγράφει εν συντομία τα προβλήματα που αντιμετωπίσατε κατά την υλοποίηση της άσκησης και τις προσεγγίσεις της ομάδας για την επίλυση τους. Εάν εντοπίσατε περισσότερες από μια προσεγγίσεις για την επίλυση, να τις αναφέρετε και να τεκμηριώσετε γιατί επιλέξατε όποια επιλέξατε.

Ειδικά και επιπρόσθετα, στις ασκήσεις που πρέπει να αναπτύξετε κάποιας μορφής κώδικα/πρόγραμμα, ο κώδικας σας πρέπει να απαντά ακριβώς σε ό,τι σας ζητείτε, χωρίς αφαιρέσεις ή προσθήκες, π.χ., αν ζητείται να υλοποιηθεί κάτι με διεργασίες (processes), αλλά επειδή δυσκολευτήκατε ή δεν καταλάβατε κάτι καλά, το υλοποιήσατε με νήματα (threads), δεν θα πάρετε τα credits που αναλογούν. Οι απαντήσεις που αφορούν κώδικα μέσα στο pdf της αναφοράς θα πρέπει να τεκμηριώνονται με screenshots από την εκτέλεση των προγραμμάτων.



Στην αναφορά (pdf), στο μέρος που αντιστοιχεί στις υλοποιήσεις σας, πρέπει να αναφέρει στην αρχή της τεκμηρίωσης:

- ποια τμήματα της άσκησης έχετε υλοποιήσει και ποια όχι
- ποια δουλεύουν σωστά και ποια δεν δουλεύουν.

ΜΗΝ συμπεριλάβετε στην αναφορά σας την εκφώνηση της άσκησης και τον κώδικά σας αυτούσιο, αλλά καλείστε να περιγράψετε λεκτικά τον αλγόριθμο, κώδικα και προσέγγιση που ακολουθήσατε.

Επικοινωνία και παράδοση του project

Όλη η επικοινωνία του μαθήματος και κατά συνέπεια και όλη η επικοινωνία για το project και η παράδοση του project θα γίνεται αποκλειστικά μέσω του eClass του μαθήματος.

Οι απορίες θα συζητούνται **πρωτίστως στο μάθημα** και δευτερευόντως με μήνυμα («Μηνύματα» του eclass) μέσω του eClass.

Η παράδοση θα γίνει μέσω eClass, μέσα από τις «Εργασίες» του μαθήματος.

Απορίες με e-mail και γενικά κάθε επικοινωνία μέσω e-mail θα αγνοείται.

Η οριστική ημερομηνία παράδοσης του project θα είναι ίδια για όλους χωρίς εξαιρέσεις και θα εμφανισθεί στο eclass (Εργασίες) και ορίζεται η ημέρα εξέτασης του μαθήματος στην εξεταστική του χειμερινού εξαμήνου 2022-2023 στις 23:59. **Τα παραδοτέα δεν μπορούν να αλλαχθούν μετά την καταληκτική ημερομηνία παράδοσης μέσω του eclass.**

Μετά την παρέλευση της παράδοσης του project, κανένα project δε θα γίνεται δεκτό μέσω e-mail, με οποιαδήποτε δικαιολογία. Εάν θέλετε να ελέγξετε πως δουλεύει η παράδοση ή εάν δουλεύει ορθά μέσω του eclass, μπορείτε καθόλη τη διάρκεια του χρονικού διαστήματος από την εκφώνηση του 2ου project μέχρι την εξέταση του μαθήματος, να υποβάλλεται όσες φορές επιθυμείτε εργασίες, αλλά θα προσμετρηθεί μόνο η τελευταία (οπότε δε θα υπάρχει δικαιολογία ότι κάτι δεν έγινε καλά εάν δραστηριοποιηθείτε έγκαιρα).

Περιπτώσεις αντιγραφής και λογοκλοπής

Projects που είναι προϊόντα αντιγραφής ή λογοκλοπής θα μηδενίζονται και ανάλογα με τη σοβαρότητα της περίπτωσης το θέμα θα παραπέμπεται στη Επιτροπή Δεοντολογίας του Τμήματος.

Όπως αναφέρεται και στις διαφάνειες της εισαγωγικής διάλεξης για τα διαδικαστικά του μαθήματος: «Η αντιγραφή απαγορεύεται!! Τι σημαίνει «αντιγραφή»; Αντιγραφή από έναν συμφοιτητή σας, αντιγραφή από web sites (π.χ., Stack Overflow), βοήθεια από «φροντιστήρια». Θα ελέγχουμε με αυτόματο τρόπο για αντιγραφή. Το να μετονομάσετε μεταβλητές ή συναρτήσεις, δεν ξεγελάει τον έλεγχο! Σε περίπτωση αντιγραφής μηδενίζετε στην εν λόγω άσκηση ΚΑΙ αφαιρείται μία ακόμη μονάδα από τον τελικό βαθμό. Δηλαδή, η αντιγραφή σε ΜΙΑ άσκηση στοιχίζει ΔΥΟ ασκήσεις! Αυτό ισχύει για όλους τους εμπλεκόμενους φοιτητές!»

Οτιδήποτε αναφέρετε στο project θα πρέπει να έχει αναφορά στις πηγές σας με τον ορθό τρόπο. Αν κάτι δεν αναφέρεται, θεωρείται αυτόματα ότι είναι προϊόν δική σας πνευματικής εργασίας. Αν όμως εντοπιστεί ότι κάτι τέτοιο δεν ισχύει, τότε το project μηδενίζεται αυτόματα, ασχέτως από την έκταση του φαινομένου (δηλαδή αν αφορά μόνο ένα σχήμα, μια εικόνα, ή ένα ολόκληρο παραδοτέο).

Στην περίπτωση που κάποιο τμήμα του έργου έχει βασιστεί σε κάτι που αναφέρετε, θα πρέπει να διευκρινίζεται λεπτομερώς και με τρόπο που δεν επιδέχεται αμφισβήτηση τι ακριβώς διαφοροποιεί η δική σας δουλειά.

Βαθμολογία

Ι.Για το τρέχων πρόγραμμα σπουδών η βαθμολογία στο μάθημα θα προέρχεται:

Τελικός Βαθμός Μαθήματος = Βαθμός Γραπτής Εξέτασης * 50% + Project 1 * 25% + Project 2 * 25%

ή

$$TB = BG * 50\% + Pr1 * 25\% + Pr2 * 25\%,$$

Όπου υποχρεωτικά μόνο ο $BG \geq 5$



Επομένως, εάν κάποιος ΔΕΝ παραδώσει ασκήσεις, τότε μπορεί να δώσει γραπτή εξέταση διεκδικώντας τον ανάλογο βαθμό σύμφωνα με το παραπάνω τυπολόγιο υπολογισμού.

Υπενθυμίζεται ότι σε ότι αφορά τη γραπτή εξέταση στο τέλος του εξαμήνου, το 4.75 στρογγυλοποιείται στο 5. Το 4.5, όμως, όχι! Επίσης, το μάθημα είναι αδιαίρετο (όχι «à la carte») κι εάν δεν το περάσετε στο τρέχον ακαδημαϊκό έτος, δίνετε τα project εκ νέου σε επόμενο ακαδημαϊκό έτος.

II. Για το παλιό πρόγραμμα σπουδών

- Λειτουργικά συστήματα I

Τελικός Βαθμός Μαθήματος = Βαθμός Γραπτής Εξέτασης * 75% + Project X * 25%, όπου X=1 ή 2

ή

$$TB = BG * 75\% + PrX * 25\%, \text{ , όπου } X=1 \text{ ή } 2$$

Αυτό σημαίνει ότι υποχρεωτικό είναι μόνο ένα από τα δύο projects κι εφόσον κάποιος παραδώσει και τα δύο θα μετρήσει ο καλύτερος βαθμός.

- Εργαστήριο Λειτουργικών Συστημάτων

Τελικός Βαθμός Μαθήματος = Project 1 * 50% + Project 2 * 50%

ή

$$TB = Pr1 * 50\% + Pr2 * 50\%$$

Που σημαίνει υποχρέωση παράδοσης και των δύο projects και δε θα υπάρχει τελική εξέταση, αλλά ο μέσος όρος των ασκήσεων είναι ο τελικός βαθμός στο εργαστήριο.



Εργαστηριακή Άσκηση

Χρονοπρογραμματιστής Διεργασιών

Εισαγωγή

Το ζητούμενο στην εργασία είναι η υλοποίηση ενός περιβάλλοντος χρονοπρογραμματισμού σε λειτουργικό σύστημα Unix. Συγκεκριμένα, πρέπει να υλοποιηθεί ένας δρομολογητής ο οποίος παίρνει σαν είσοδο τις εφαρμογές που πρέπει να εκτελέσει, διαβάζοντας ένα αρχείο με τα ονόματά τους, τις εισάγει σε μία κατάλληλη δομή δεδομένων (λίστα) και κατόπιν τις δρομολογεί εφαρμόζοντας μία από τις ακόλουθες πολιτικές:

- **Εξυπηρέτηση με βάση τη σειρά άφιξης (FCFS):** οι εφαρμογές εκτελούνται με τη σειρά εισαγωγής τους στη λίστα, δηλαδή με τη σειρά που αναγράφονται στο αρχείο εισόδου
- **Εξυπηρέτηση με βάση τη μικρότερη διάρκεια (SJF):** κάθε εφαρμογή χαρακτηρίζεται από έναν στατικό χρόνο εκτέλεσης (αριθμό), ο οποίος καθορίζει τη σειρά εκτέλεσής της. Οι εφαρμογές με το μικρότερο αριθμό έχουν τη μεγαλύτερη προτεραιότητα. Αν δύο εφαρμογές έχουν τον ίδιο αριθμό τότε προτεραιότητα έχει η διεργασία που εμφανίζεται πρώτη στο αρχείο εισόδου.
- **Δρομολόγηση εκ περιτροπής (RR):** οι εφαρμογές εκτελούνται εκ περιτροπής για χρόνο ίσο με προκαθορισμένο κβάντο δρομολόγησης.
- **Δρομολόγηση με βάση την προτεραιότητα (PRIO):** κάθε εφαρμογή χαρακτηρίζεται από μια **στατική** προτεραιότητα (αριθμό). Οι εφαρμογές με το μικρότερο αριθμό έχουν μεγαλύτερη προτεραιότητα και εκτελούνται πρώτα. Μεταξύ εφαρμογών με τον ίδιο βαθμό προτεραιότητας, εφαρμόζεται δρομολόγηση εκ περιτροπής.

Η υλοποίηση του δρομολογητή πρέπει να πραγματοποιηθεί με τέτοιο τρόπο ώστε οι βασικές δομές δεδομένων και οι εσωτερικές του λειτουργίες να είναι όσο το δυνατόν περισσότερο συμπαγείς και ανεξάρτητες της πολιτικής που εφαρμόζει. Π.χ., η πολιτική θα μπορούσε να μεταβάλλεται δυναμικά κατά τον χρόνο εκτέλεσης. Η παρούσα ενότητα αυτή περιλαμβάνει μια γενική περιγραφή των βασικών στοιχείων της υλοποίησης του δρομολογητή που θα πρέπει να λάβετε υπόψη.

Περιγραφές εφαρμογής

Κάθε εφαρμογή που πρόκειται να εκτελεστεί από τον δρομολογητή θα περιγράφεται με μία κατάλληλη δομή δεδομένων. Μία τέτοια δομή θα δεσμεύεται και αρχικοποιείται κατάλληλα για κάθε εφαρμογή που διαβάζεται από το αρχείο εισόδου και πρόκειται να εκτελεστεί από το δρομολογητή. Στην συνέχεια θα εισάγεται στην ουρά εκτέλεσης. Η δομή αυτή θα περιλαμβάνει πεδία που περιγράφουν χαρακτηριστικά της εφαρμογής, όπως π.χ. όνομα εκτελέσιμου αρχείου, προτεραιότητα, αναγνωριστικό (pid), κατάσταση εκτέλεσης κ.α.

Δυνατές καταστάσεις εκτέλεσης μιας εφαρμογής (διεργασίας) είναι οι εξής:

- **READY:** Η εφαρμογή έχει μόλις εισαχθεί σε ουρά για εκτέλεση. Οι εφαρμογές εκτελούνται μέσω του δρομολογητή, ο οποίος δημιουργεί τη διεργασία για την εκτέλεση μιας εφαρμογής (με *fork-exec*).
- **RUNNING:** Η εφαρμογή – διεργασία είναι ενεργή.
- **STOPPED:** Η εκτέλεση της εφαρμογής έχει διακοπεί.
- **EXITED:** Η εφαρμογή έχει τερματίσει.

Ουρά Εκτέλεσης

Η δομή δεδομένων που περιέχει τους περιγραφείς των εφαρμογών πρέπει να είναι μία διπλά συνδεδεμένη λίστα (ουρά). Εισαγωγή διεργασιών πραγματοποιείται στο τέλος της λίστας ενώ εξαγωγή από την αρχή της. Στην περίπτωση της δρομολόγησης με βάση την προτεραιότητα, μπορείτε να χρησιμοποιήσετε τόσες λίστες όσες και οι δυνατές τιμές προτεραιοτήτων (π.χ. κάποιος μικρός ακέραιος) ή να ακολουθήσετε κάποια άλλη προσέγγιση.

Δρομολόγηση Εφαρμογών

Ο δρομολογητής επιλέγει κάθε φορά την επόμενη προς εκτέλεση εφαρμογή, και εφόσον υπάρχει κάποια εφαρμογή **router** πραγματοποιεί τις κατάλληλες ενέργειες διαφορετικά τερματίζει.

Θεωρούμε ότι οι εφαρμογές που θα δρομολογηθούν περιλαμβάνουν αποκλειστικά υπολογισμούς και καθόλου εντολές εισόδου – εξόδου.



Κατά συνέπεια, στην περίπτωση των στατικών πολιτικών δρομολόγησης (FCFS, SJF), ο δρομολογητής θα ενεργοποιείται μόνο όταν μια εφαρμογή – διεργασία τερματίζει.

Αντίθετα, στην περίπτωση των δυναμικών πολιτικών δρομολόγησης (RR, PRIO), ο δρομολογητής θα ενεργοποιείται κατά τακτά χρονικά διαστήματα, όπως καθορίζεται από το κβάντο δρομολόγησης που αποτελεί παράμετρο του προγράμματος. Ο μηχανισμός ενεργοποίησης του δρομολογητή θα υλοποιηθεί με χρήση κατάλληλου timer (π.χ. με χρήση της κλήσης **nanosleep**).

Στην κατηγορία των δυναμικών πολιτικών δρομολόγησης, η διαχείριση της εκτέλεσης των εφαρμογών – διεργασιών, δηλαδή η αναστολή και η συνέχιση της εκτέλεσής τους, θα πραγματοποιείται με χρήση των σημάτων **SIGSTOP** και **SIGCONT**.

Τέλος, όταν μια εφαρμογή – διεργασία τερματίζει, ο δρομολογητής θα πρέπει να ενημερώνεται, έχοντας θέσει χειριστή για το σήμα **SIGCHLD**, και να εκτελεί τις απαραίτητες ενέργειες για τη σωστή λειτουργία του.

Σε κάθε περίπτωση, όταν μια εφαρμογή τερματίζει θα πρέπει να υπολογίζεται και να εκτυπώνεται ο συνολικός χρόνος εκτέλεσής της, ξεκινώντας από τη στιγμή που μπήκε στην λίστα εκτέλεσης, αλλά και ο χρόνος που έχει περάσει από την εκκίνηση του δρομολογητή. Για λόγους απλότητας, όλες οι διεργασίες **ektipwsi xronou.txt**

Χρήσιμες υποδείξεις

1. Ο δρομολογητής θα πρέπει να λαμβάνει το σήμα **SIGCHLD** μόνο όταν μια εφαρμογή – παιδί τερματίζει και όχι όταν αναστέλλεται η εκτέλεσή του (βλ. **sigaction**)
2. Είναι χρήσιμο ο δρομολογητής να διατηρεί ιστορικό εκτέλεσης, δηλαδή πληροφορία με την τρέχουσα εκτελούμενη εφαρμογή σε κάθε κβάντο.
3. Κατά τον τερματισμό μιας διεργασίας, ο δρομολογητής θα πρέπει να εμφανίζει πληροφορίες για την εκτέλεση της αντίστοιχης εφαρμογής. Εναλλακτικά, η εκτύπωση των πληροφοριών μπορεί να πραγματοποιηθεί πριν την ολοκλήρωση της εκτέλεσης του δρομολογητή. Ο δρομολογητής θα πρέπει να εμφανίζει και το συνολικό χρόνο εκτέλεσης του συνόλου των εφαρμογών.
4. Για την ανάπτυξη του δρομολογητή, μπορείτε να χρησιμοποιήσετε απλά προγράμματα που επιτρέπουν τον έλεγχο της σωστής λειτουργίας του.
5. Ο κώδικας των παραπάνω προγραμμάτων, χαρακτηριστικά αρχεία εισόδου, ένα αρχείο (script) με παραδείγματα εκτέλεσης του δρομολογητή, ένα ενδεικτικό παράδειγμα εκτύπωσης αποτελεσμάτων, καθώς και ένας ενδεικτικός κώδικας του δρομολογητή, τον οποίο πρέπει να συμπληρώσετε, δίνονται μαζί με την εκφώνηση της άσκησης.

Χρήση

Σύμφωνα με τα παραπάνω, το πρόγραμμα του δρομολογητή θα χρησιμοποιείται ως εξής: **scheduler <policy> [<quantum>] <input_filename>**, όπου

- **scheduler**: το εκτελέσιμο του δρομολογητή που θα υλοποιήσετε.
- **policy**: η πολιτική δρομολόγησης με την οποία θα εκτελεστούν οι εφαρμογές. Δυνατές τιμές είναι οι FCFS, SJF, RR, PRIO.
- **quantum**: Το κβάντο δρομολόγησης σε msec. Απαιτείται μόνο όταν ως πολιτική δρομολόγησης έχει οριστεί μία από τις RR ή PRIO.
- **input_filename**: Το όνομα του αρχείου το οποίο περιέχει το φορτίο εργασιών που θα εκτελεστούν μέσω του δρομολογητή. Κάθε γραμμή του αρχείου περιλαμβάνει το όνομα του εκτελέσιμου μιας εφαρμογής και έναν αριθμό που θα δηλώνει τον απαιτούμενο χρόνο εκτέλεσης (σύμφωνα με κάποια σχετική ή απόλυτη κλίμακα) ή την προτεραιότητά του.

Παραδείγματα χρήσης:

```
$ scheduler FCFS reverse.txt
```

```
$ scheduler SJF reverse.txt
```

```
$ scheduler RR 1000 reverse.txt
```

```
$ scheduler PRIO 500 reverse.txt
```



Καλείστε να συμπληρώσετε το ζητούμενο πρόγραμμα δρομολόγησης το εκτελέσιμο του οποίου θα ονομάσετε scheduler.

Αναλυτικότερα:

1. Υλοποιήστε τις βασικές δομές και λειτουργίες του χρονοδρομολογητή.
2. Υλοποιήστε την πολιτική δρομολόγησης FCFS.
3. Υλοποιήστε την πολιτική δρομολόγησης SJF.
4. Υλοποιήστε την πολιτική δρομολόγησης RR.
5. Υλοποιήστε την πολιτική δρομολόγησης PRIO.
6. Εκτελέστε και αξιολογήστε τις παραπάνω πολιτικές για τα παραδείγματα εκτέλεσης του δρομολογητή όπως ορίζονται στο αρχείο run.sh.
7. Γράψτε την αναφορά σύμφωνα με τις οδηγίες και παρουσιάστε τα αποτελέσματα από τα πειράματά σας.

Θα βαθμολογηθείτε ξεχωριστά για όλα τα παραπάνω ζητούμενα. Αν δεν έχετε υλοποιήσει όλες τις πολιτικές δρομολόγησης, τότε φροντίστε ώστε ο κώδικας σας να εκτυπώνει κατάλληλο μήνυμα για αυτές αλλά και να εκτελείται σωστά για όλες τις υπόλοιπες πολιτικές.

Παρατηρήσεις:

- Μην συμπεριλάβετε εκτελέσιμα αρχεία στο αρχείο zip με τον κώδικα σας.
- Μπορείτε να κάνετε οποιεσδήποτε αλλαγές στον ημιτελή κώδικα του δρομολογητή αλλά θα πρέπει να ακολουθήσετε τη διεπαφή (δηλαδή τις παραμέτρους εκτέλεσης) που ορίζονται στην εκφώνηση και χρησιμοποιούνται στα παραδείγματα εκτέλεσης.
- Ο κώδικας που θα παραδώσετε θα πρέπει να μπορεί να μεταγλωττιστεί και να εκτελέσει scripts αντίστοιχα με το run.sh. Το πρώτο στάδιο αξιολόγησης του κώδικα της εργασίας σας θα πραγματοποιηθεί με αυτό ακριβώς τον τρόπο.
- Τα εκτελέσιμα για τα βοηθητικά προγράμματα και τον δρομολογητή μπορούν να δημιουργηθούν με την εκτέλεση της εντολής make στο τερματικό και ενώ είστε στο αντίστοιχο directory.
- Ο χρόνος εκτέλεσης των βοηθητικών προγραμμάτων (load1, load2, ...) μπορεί να προσαρμοστεί θέτοντας κατάλληλη τιμή στη σταθερά DELAY, όπως φαίνεται και στο αρχείο Makefile.
- Μια υλοποίηση της εργασίας έχει υλοποιηθεί και δοκιμαστεί σε συστήματα Linux (όπως και στο σύστημα diogenis) χωρίς προβλήματα.
- Αν εκτελέσετε τα πειράματά σας σε σύστημα το οποίο χρησιμοποιείται από ταυτόχρονα από πολλούς χρήστες (π.χ. diogenis), τότε ο χρόνος εκτέλεσης των βοηθητικών προγραμμάτων αναμένεται να διαφέρει κάθε φορά, ανάλογα με το συνολικό φόρτο του συστήματος.

Καλή επιτυχία!