



# Λειτουργικά Συστήματα

2024 - 2025

## 1η Εργαστηριακή Άσκηση (Project)

### 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 του project θα πρέπει να αποτελούνται έως 4 άτομα.

### 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».

Θα παραδοθούν:

- η αναφορά σε pdf στην εργασία «1ο project 2024-2025, Αναφορά Ομάδων»
- τα αρχεία του κώδικας σας σε μορφή zip στην εργασία «1ο project 2024-2025, 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, στις θεωρητικές απαιτείται σκεπτικό αλγορίθμικής προσέγγισης, χρησιμοποιώντας διαγράμματα ή/και ψευδοκώδικα όπου θεωρείτε απαραίτητο.

Επίσης, θα περιγράφει εν συντομίᾳ 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 κώδικα 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 2024-2025 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 ελέγχετε πως δουλεύει η παράδοση ή εάν δουλεύει ορθά μέσω του eclass, μπορείτε καθόλη τη διάρκεια του χρονικού διαστήματος από την εκφώνηση 1ου 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 απα-γο-ρεύ-ε-ται!! Τι σημαίνει «αντιγραφή»; Αντιγραφή από έναν συμφοιτητή σας, αντιγραφή από web sites (π.χ., Stack Overflow), βιόθεια από «φροντιστήρια». Θα ελέγχουμε με αυτόματο τρόπο για αντιγραφή. Το να μετονομάσετε μεταβλητές ή συναρτήσεις, δεν ξεγελάει τον έλεγχο! Σε περίπτωση αντιγραφής μηδενίζεστε στην εν λόγω άσκηση ΚΑΙ αφαιρέται μία ακόμη μονάδα από τον τελικό βαθμό. Δηλαδή, η αντιγραφή σε ΜΙΑ άσκηση στοιχίζει ΔΥΟ ασκήσεις! Αυτό ισχύει για όλους τους εμπλεκόμενους φοιτητές!»

Οιδήπτοτε 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

I. 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  $B\Gamma \geq 5$

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

$$\text{Τελικός Βαθμός Μαθήματος} = \text{Βαθμός Γραπτής Εξέτασης} * 75\% + \text{Project X} * 25\%, \text{ όπου } X=1 \text{ ή } 2$$

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

$$\text{Τελικός Βαθμός Μαθήματος} = \text{Project 1} * 50\% + \text{Project 2} * 50\%$$

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.



# 1. Shell Scripting (30 Μονάδες)

Θα προσομοιώσουμε μια κατάσταση όπου σε μια άσκηση εγκατάλειψης πλοίου, είναι επιθυμητό να πραγματοποιηθεί μια διαχείριση και ανάλυση των δεδομένων των επιβαινόντων που συμμετέχουν στην άσκηση. Να γράψετε ένα shell script που θα επεξεργάζεται ένα αρχείο με πληροφορίες επιβατών και θα παράγει αναφορές. Το όνομα του αρχείου του shell script θα είναι:

**processes\_ipc.sh**

Οι λειτουργίες θα εκτελούνται ως εξής:

- [1] Εισαγωγή δεδομένων στην εφαρμογή

Το πρόγραμμα ζητά εισαγωγή δεδομένων. Αν δε δοθεί κάποιο όνομα αρχείου- συμπεριλαμβανομένου και του path που βρίσκεται, θα ζητάει τα δεδομένα από το πληκτρολόγιο. Ένα δείγμα αρχείου είναι το αρχείο passengers.csv (βρίσκεται στο eclass). Στο αρχείο αυτό βρίσκονται αποθηκευμένα τα στοιχεία των επιβαινόντων, όπου σε κάθε γραμμή βρίσκεται ένας με την κάτωθι μορφή:

```
[code], [fullname], [age], [country], [status (Passenger/Crew)], [rescued  
(Yes/No)]
```

Εάν διαβάζει τα δεδομένα από το πληκτρολόγιο, να τα αποθηκεύει στο αρχείο passengers.csv.

- [2] Προβολή στοιχείων επιβαίνοντα από το αρχείο, όπου θα εισάγετε το όνομα ή το επώνυμο του επιβαίνοντα, θα αναζητά τα στοιχεία του και τα δείχνει στην οθόνη.

- [3] Άλλαγή στοιχείων επιβαίνοντα, θα ζητά τον κωδικό ή το όνομα ή το επώνυμο του επιβαίνοντα και το πεδίο που χρειάζεται αλλαγή. Ο χειριστής θα μπορεί να επιλέξει ένα πεδίο για διόρθωση ή όλη την εγγραφή. Η επιλογή του συγκεκριμένου πεδίου πρέπει να γίνεται δίνοντας ως όρισμα το όνομα του πεδίου και τη νέα τιμή που θα λάβει. Με όρισμα record θα επιλέγει όλη την εγγραφή. Ο διαχωρισμός θα πραγματοποιείται με :. Θα αλλάζει/ διορθώνει τα στοιχεία και προβάλλει στην οθόνη τα παλαιά και τα νέα στοιχεία του επιβαίνοντα. Ακολουθεί ένα υπόδειγμα μορφής των ορισμάτων:

```
./processes_ipc <κωδικό ή το όνομα ή το επώνυμο του επιβαίνοντα>  
fullname:<νέα είσοδος>
```

```
./processes_ipc <κωδικό ή το όνομα ή το επώνυμο του επιβαίνοντα>  
record:<νέα είσοδος>
```

- [4] Προβολή αρχείου, θα δείχνει στην οθόνη όλους τους επιβαίνοντες και τα στοιχεία τους. Το πρόγραμμα θα πρέπει να σταματά την εκτύπωση όταν έχει γεμίσει η οθόνη, αναμένοντας τον χρήστη να πατήσει το <space> για να δείξει περισσότερες εγγραφές.

- [5] Δημιουργία αναφορών με τα ακόλουθα χαρακτηριστικά:

- Εύρεση και προβολή των ηλικιακών ομάδων των επιβατών (π.χ., 0-18, 19-35, 36-50, 51+).
- Υπολογισμός του ποσοστού των συμμετεχόντων στη διάσωση για κάθε ηλικιακή ομάδα.
- Υπολογισμός μέσης ηλικίας ανά κατηγορία επιβατών.
- Φιλτράρισμα των δεδομένων ώστε να δημιουργηθεί ένα νέο αρχείο που θα περιέχει μόνο τους «διασωθέντες».

Η δημιουργία αναφορών θα πραγματοποιείται με την εκτέλεση:

```
./processes_ipc reports
```

Η εξαγωγή των αναφορών θα πραγματοποιείται στην στα αρχεία:

- ages.txt
- percentages.txt
- avg.txt
- rescued.txt

## Απαιτούμενα



- Χρησιμοποιήστε την AWK, για την αναλυτική επεξεργασία και τον υπολογισμό στατιστικών.
- Χρησιμοποιείστε τις SED για ενημέρωση εγγραφών στα αρχεία δεδομένων και GREP για φίλτραρισμα.
- Δημιουργείστε τις συναρτήσεις:
  - insert\_data
  - search\_passenger
  - update\_passenger
  - display\_file
  - generate\_reports

## 2. Συγχρονισμός Διεργασιών και Σημαφόροι (30 Μονάδες)

Θα προσομοιώσουμε μια κατάσταση όπου σε μια άσκηση εγκατάλειψης πλοίου, θα γίνει προσπάθεια να βρεθεί ένας καλός τρόπος για τη διάσωση των επιβαίνοντων. Στο σκάφος οι διαθέσιμες λέμβοι είναι επίσης περιορισμένες και πολύ λιγότερες από το μέγιστο πλήθος επιβαίνοντων και σε κάθε λέμβο διάσωσης, επίσης, ο αριθμός θέσεων των διαθέσιμων θέσεων είναι περιορισμένος. Επειδή η στεριά είναι κοντά, εάν υπάρξει ο κατάλληλος συντονισμός, τότε όλοι οι επιβαίνοντες θα διασωθούν σε πραγματικές συνθήκες. Οι επιβαίνοντες προσπαθούν να επιβιβαστούν με ασφαλή και συγχρονισμένο τρόπο. Ο αριθμός αυτών που μπορούν να επιβιβαστούν θα καθορίζεται από έναν αριθμητικό περιορισμό μέσω σημαφόρου.

Στόχος είναι να επιτρέψουμε στις διεργασίες να επιβιβάζονται μέχρι τη μέγιστη χωρητικότητα της λέμβου και να περιμένουν εάν αυτή να είναι πλήρης. Οι επιβαίνοντες δημιουργούν μια ουρά προς τις λέμβους και αρχίζουν την επιβιβαση. Όταν ωστόσο ένας επιβαίνων αλλάζει γνώμη και δεν επιβιβαστεί τότε μετακινείται στο τέλος της ουράς και μια θέση αδειάζει και επιπρέπει σε κάποιον άλλον να επιβιβαστεί. Μόλις μια λέμβος γεμίσει αναχωρεί. Επίσης εάν μια λέμβος επιστρέψει από αποβίβαση στη στεριά τοποθετείται στην ουρά για να μπορέσει να εξυπηρετείται επιβάτες.

Να κατασκευάσετε πρόγραμμα που να συγχρονίζει την παραπάνω διαδικασία εγκατάλειψης του πλοίου. Η απάντηση σας θα αποτελείται από τα εξής αρχεία:

1. ipc\_utils.h, για τη ρύθμιση των σημαφόρων και βασικές βιοθητικές συναρτήσεις,
2. passenger.c, διεργασία για τους επιβαίνοντες που προσπαθούν να επιβιβαστούν στη λέμβο, και
3. launch.c, το αρχείο εκκίνησης που- μεταξύ άλλων- θα διαβάζει το πλήθος των επιβαίνοντων, των λέμβων και των διαθέσιμων θέσεων σε αυτές και θα ρυθμίζει τις θέσεις στη λέμβο.

## 3. Χρονοπρογραμματισμός Διεργασιών και Διαχείριση Μνήμης (20 Μονάδες)

Σε ένα υπολογιστικό σύστημα με 512 KB συνολικό χώρο μνήμης για τις διεργασίες των χρηστών φθάνουν οι πέντε διεργασίες με μεταβλητούς χρόνους άφιξης, διάρκεια και απαιτούμενη μνήμη την οποία θα διαβάζει από το πληκτρολόγιο. Η μνήμη είναι μεταβλητών διαμερίσεων και για τη χρονοδρομολόγηση εφαρμοστεί ο αλγόριθμος Round Robin με κενάτο χρόνου 3mses και για την τοποθέτηση στη μνήμη εφαρμοστεί ο αλγόριθμος του πρώτου ταιριάσματος, αναφέρετε σε κάθε χρονική στιγμή την κατάσταση της CPU και της μνήμης για την εξυπηρέτηση των παραπάνω των διεργασιών.

### Υπόδειγμα:

Κάθε διεργασία έχει αναγνωριστικό (pid), χρόνο άφιξης (arrival\_time), διάρκεια (duration), υπολειπόμενο χρόνο εκτέλεσης (remaining\_time), απαιτούμενη μνήμη (memory\_needed), και ένδειξη αν βρίσκεται στη μνήμη (in\_memory).

Η μνήμη μοντελοποιείται ως πίνακας MemoryBlock με το κάθε στοιχείο να αναπαριστά 1KB.

Το κάθε μπλοκ έχει θέση έναρξης (start), μέγεθος (size), αν είναι ελεύθερο (free), και τον pid της διεργασίας που το κατέχει.

Όλη η μνήμη ξεκινά ελεύθερη.



## 4. Χρονοπρογραμματισμός Διεργασιών (20 Μονάδες)

Θεωρείστε έξι (6) διεργασίες με χρόνους άφιξης (XA) και εκτέλεσης (XE) όπως δίνονται στον ακόλουθο πίνακα (οι χρόνοι στον πίνακα αναφέρονται στην ίδια χρονική μονάδα π.χ. ms):

Διεργασία	Χρόνος άφιξης (XA)	Διάρκεια εκτέλεσης (XE)	PID
A	0	6	3
B	2	4	1
Γ	3	1	2
Δ	4	3	5
Ε	5	5	4
Ζ	6	7	1

(α) Σχεδιάστε το διάγραμμα Gantt για τους αλγόριθμους χρονοδρομολόγησης:

- FCFS (First Come First Served). (**1 Μονάδα**)
- SJF (Shortest Job First). (**1 Μονάδα**)
- SRTF (Shortest Remaining Time First). (**1 Μονάδα**)
- RR (Round Robin) με κβάντο χρόνου 2 χρονικές μονάδες. Θεωρείστε πώς αν μια χρονική στιγμή μια διεργασία αφήνει την KME και την ίδια χρονική στιγμή καταφθάνει μια νέα διεργασία στο σύστημα, τότε η διεργασία που αφήνει την KME εισέρχεται πρώτη στην ουρά των διεργασιών. (**1 Μονάδα**)

Γενικά, μεταξύ των διεργασιών που ενδέχεται να αφιχθούν την ίδια χρονική στιγμή, θεωρείστε ότι προηγείται αυτή που θα αφιχθεί πρώτη στο σύστημα. Ο χρόνος θεματικής εναλλαγής (content switch) είναι αμελητέος.

(β) Για καθένα από τους παραπάνω αλγόριθμους υπολογίστε:

- το μέσο χρόνο αναμονής (**1 Μονάδα**)
- μέσο χρόνο απόκρισης (**1 Μονάδα**)
- μέσο χρόνο ολοκλήρωσης (**1 Μονάδα**)
- το πλήθος των θεματικών εναλλαγών των διεργασιών (**1 Μονάδα**)

(γ) Σχεδιάστε έναν προεκχωρητικό αλγόριθμο χρονοδρομολόγησης διεργασιών, ο οποίος θα εξυπηρετεί πρώτες αυτές με το μεγαλύτερο χρόνο που απομένει. Η προεκχώρηση πραγματοποιείται λαμβάνοντας υπόψη τις προτεραιότητες που έχουν οι διεργασίες. Τον αλγόριθμο τον ονομάζουμε Longest Remaining Time First Preemptive-LRTFP. Ο αλγόριθμος εξετάζει πρώτα την ιδιότητα LRTF για την προεκχώρηση και σε περιπτώσεις ισοδυναμίας ενεργοποιείται η PID προτεραιότητα, δίνοντας προτεραιότητα στη διαδικασία με το μικρότερο αναγνωριστικό διεργασίας (PID- Process ID). Να:

- σχεδιάστε το διάγραμμα Gantt για τον LRTFP (**4 Μονάδα**)
- το μέσο χρόνο αναμονής του (**2 Μονάδα**)
- μέσο χρόνο απόκρισης (**2 Μονάδα**)
- μέσο χρόνο ολοκλήρωσης (**2 Μονάδα**)
- το πλήθος των θεματικών εναλλαγών των διεργασιών. (**2 Μονάδα**)

Καλή επιτυχία!!!