



Λειτουργικά Συστήματα

2022 - 2023

1η 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 μέχρι 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

#Panagioris Chatzidoukas, 3456789

#Aristidis Ilias, 1234567

Τα παραδοτέα θα υποβληθούν μέσω του eclass στις «Εργασίες», όπου θα παραδώσει ένα μέλος της ομάδας και θα είναι το ίδιο μέλος για όλα τα παραδοτέα (και για το δεύτερο project).

Θα παραδοθούν:

- η αναφορά σε pdf στην εργασία «1ο project 2022-2023, Αναφορά Ομάδων»
- τα αρχεία του κώδικα σας σε μορφή zip στην εργασία «1ο 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, στις θεωρητικές απαιτείται σκεπτικό αλγοριθμικής προσέγγισης, διαγράμματα και ψευδοκώδικας.

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 υλοποιήσατε 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 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 α-πα-γο-ρεύ-ε-ται!! 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 δικής 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

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

$$\text{Τελικός Βαθμός Μαθήματος} = \text{Βαθμός Γραπτής Εξέτασης} * 50\% + \text{Project 1} * 25\% + \text{Project 2} * 25\%$$

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.



Εργαστηριακές Ασκήσεις

Ερώτημα 1: Shell Scripting (42 Μονάδες, 6 μονάδες ανά υποερώτημα)

Καλείστε να γράψετε ένα πρόγραμμα το οποίο θα ονομάσετε `logparser.sh` σε BASH shell. Στην αρχή του αρχείου του κώδικα, σε σχόλια, θα τοποθετείτε τα ονοματεπώνυμα και τους ΑΜ των μελών της ομάδας. Το πρόγραμμα προορίζεται για τη διαχείριση του αρχείου καταγραφής `access.log` (σας δίνεται συνημμένα με την εκφώνηση).

Μια εγγραφή αρχείου καταγραφής για το `access.log`, θα μοιάζει κάπως έτσι:

```
127.0.0.1 - userid - [10/Feb/2022:13:55:36 +0300] "GET /server-status HTTP/1.1" 200 2326
```

ή

```
127.0.0.1 - - [29/Mar/2022:17:02:15 +0300] "POST /phpmyadmin/index.php?route=/navigation&ajax_request=1 HTTP/1.1" 200 3941 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0) Gecko/20100101 Firefox/98.0"
```

ή

```
:::1 - - [19/Apr/2022:10:22:23 +0300] "GET /phpmyadmin/themes/pmahomme/img/b_bookmark.png HTTP/1.1" 200 637
```

όπου τα πεδία στο παραπάνω δείγμα εγγραφής αντιπροσωπεύουν τα εξής:

- 127.0.0.1 - Διεύθυνση IP του πελάτη που υπέβαλε το αίτημα (στη συγκεκριμένη περίπτωση localhost) και το ::1 είναι η διεύθυνση loopback στο ipv6, ταυτόσημη με 127.0.0.1 στο ipv4.
- Η πρώτη παύλα ορίζει το δεύτερο πεδίο στο αρχείο καταγραφής που είναι η ταυτότητα του πελάτη. Εάν το αίτημα δεν ελέγχθηκε με έλεγχο ταυτότητας, η καταχώριση στο αρχείο καταγραφής για την τιμή αυτή είναι -.
- Userid- η ταυτότητα του ατόμου που ζητά τον πόρο.
- [10/Feb/2022:13:55:36 +0300]- ημερομηνία, ώρα και χρονική ζώνη του αιτήματος [ημέρα 2 ψηφία, μήνας 3 χαρακτήρες, έτος 4 ψηφία, ώρα 2 ψηφία, λεπτά 2 ψηφία, δευτερόλεπτα 2 ψηφία, ζώνη ('+' | '-') 4 ψηφία]

Ειδικά για τη ζώνη της ώρας, οι σφραγίδες ώρας περιέχουν τα σύμβολα '+' και '-' για να υποδηλώνουν τις ζώνες μετατοπισμένες από το UTC (Coordinated Universal Time) και εάν η ζώνη ώρας είναι μπροστά από το UTC (προς τα ανατολικά) θα έχει πρόσημο + (θετικό), ενώ προς τα δυτικά θα έχει πρόσημο - (αρνητικό). Ο ίδιος ο αριθμός αντιπροσωπεύει τη ζώνη ώρας και τη μετατόπισή της από την ώρα UTC, δηλαδή πόσες ώρες μπροστά ή πίσω από το UTC βρίσκεται το σύστημα.

- "GET /server-status HTTP/1.1"- τύπος αιτήματος και πόρος.
- 200- Κωδικός κατάστασης απόκρισης HTTP.
- 2326- μέγεθος του αντικειμένου που επιστράφηκε στον πελάτη.
- "http://localhost/"- είναι η αναφορά HTTP, η οποία αντιπροσωπεύει τη διεύθυνση από την οποία προήλθε το αίτημα για τον πόρο.
- "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, όπως Gecko) Chrome/78.0.3904.108 Safari/537.36"- παράγοντας χρήστη, ο οποίος προσδιορίζει πληροφορίες σχετικά με το πρόγραμμα περιήγησης που χρησιμοποιεί ο πελάτης πρόσβαση στον πόρο.

Το `logparser.sh` θα πρέπει να εκτελεί τις ακόλουθες λειτουργίες, κατά τις οποίες θα ελέγχεται εάν το <filename> έχει επέκταση ονόματος .log και εφόσον δεν είναι αυτής της μορφής θα εμφανίζει το μήνυμα «Wrong File Argument»:

- i. Όταν εκτελείτε χωρίς καμία παράμετρο

```
./logparser.sh
```

θα εμφανίζει τον ΑΜ των μελών της ομάδας σας, χωρισμένους μεταξύ τους με «|», χωρίς κανέναν άλλο χαρακτήρα, για παράδειγμα:

```
1234567|1245678|1345654|1343425
```

- ii. Όταν εκτελείτε:



```
./logparser.sh <filename>
```

όπου <filename> είναι το access.log, θα εμφανίζει όλα τα περιεχόμενα του αρχείου που δόθηκε ως όρισμα στο <filename>, στην οθόνη ανά γραμμή.

iii. Όταν εκτελείτε:

```
./logparser.sh <filename> --usrid
```

θα πρέπει να σας εμφανίζει (μετράει) πόσες φορές επαναλαμβάνεται κάθε χρήστης- συμπεριλαμβανομένου και του κενού χρήστη- στα περιεχόμενα του αρχείου και δίπλα το όνομα του χρήστη αλφαβητικά ταξινομημένους, π.χ.

```
7 -  
4 admin  
2 root  
8 user1  
5 user2
```

Για την λειτουργία αυτή θα πρέπει να υλοποιήσετε μια συνάρτηση, την mining_usernames().

Όταν εκτελείτε:

```
./logparser.sh <filename> --usrid <user_id>
```

όπου <user_id> το όνομα του χρήστη, θα πρέπει να ζητήσετε να σας εμφανίζει τα περιεχόμενα του αρχείου που εμφανίζουν κάποιον χρήστη, όπως για παράδειγμα η εκτέλεση:

```
./logparser.sh access.log --usrid root
```

θα εμφανίζει:

```
127.0.0.1 - root - [29/Mar/2022:16:47:20 +0300] "GET /dashboard/images/fastly-  
logo.png HTTP/1.1" 200 1770 "http://localhost/dashboard/" "Mozilla/5.0 (Windows  
NT 10.0; Win64; x64; rv:98.0) Gecko/20100101 Firefox/98.0"  
  
::1 - root - [19/Apr/2022:10:20:45 +0300] "GET  
/phpmyadmin/themes/pmahomme/img/b_routines.png HTTP/1.1" 200 306
```

iv. Όταν εκτελείτε:

```
./logparser.sh <filename> -method <method_name>
```

όπου <method_name> θα είναι GET ή POST, θα εμφανίζει τα περιεχόμενα του αρχείου που δόθηκε στο <filename> τα οποία αντιστοιχούν στη μέθοδο που δόθηκε ως όρισμα στο -method. Εάν δε συμπληρωθεί το <method_name> ή συμπληρωθεί εσφαλμένα, να εμφανίζεται το μήνυμα «Wrong Method Name».

v. Όταν εκτελείτε:

```
./logparser.sh <filename> --servprot <Network_Protocol>
```

όπου όρισμα στην παράμετρο --servprot θα δίνεται IPv4 ή IPv6 ως <Network_Protocol> και θα εμφανίζει τα περιεχόμενα του αρχείου που δόθηκε ως όρισμα στο <filename> τα οποία αντιστοιχούν στο ένα ή στο άλλο πρωτόκολλο. Εάν το <Network_Protocol> είναι κενό ή οτιδήποτε άλλο, θα εμφανίζει το μήνυμα «Wrong Network Protocol».

vi. Όταν εκτελείτε:

```
./logparser.sh <filename> --browsers
```

θα καλείται η συνάρτηση count_browsers στην οποία θα εμπεριέχεται και η συνάρτηση match(), που εμφανίζει τους browsers Mozilla, Chrome, Safari και Edg και μετά από κάθε browser τον αριθμό των φορών που χρησιμοποιήσαν, όπως για παράδειγμα:

```
Mozilla 234  
Chrome 456  
Safari 789  
Edg 876
```

vii. Όταν εκτελείτε:

```
./logparser.sh <filename> --datum <Date>
```



όπου το <Date> θα πρέπει να προβλεφθεί ότι μπορεί να είναι:

- μήνας, θα αποδέχεται μόνο αλφαριθμητικά της μορφής Jan...Dec και θα εμφανίζει όλες τις εγγραφές του αρχείου access.log που πραγματοποιήθηκαν μέσα σε αυτόν το μήνα ανεξαρτήτως έτους,
- κενό ή οτιδήποτε άλλο, θα πρέπει να εμφανίζει το μήνυμα «Wrong Date».

Απαιτούμενα συνολικά για την άσκηση: Όπου είναι δυνατόν θα πρέπει να γίνει χρήση case, sed και awk.



Ερώτημα 2: Διεργασίες (20 Μονάδες)

Υπολογισμός ολοκληρώματος συνάρτησης

Ο παρακάτω κώδικας υπολογίζει το ολοκλήρωμα της συνάρτησης $f(x) = \ln(x) \cdot \sqrt{x}$ στο διάστημα $[1, 4]$. Ο κώδικας είναι διαθέσιμος στα προγραμματιστικά παραδείγματα του μαθήματος για διαδιεργασιακή επικοινωνία (κατάλογος `os_ipc/4_shmem`). Στον ίδιο κατάλογο είναι διαθέσιμη και η έκδοση του κώδικα που χρησιμοποιεί πολλαπλές διεργασίες και κοινή μνήμη.

```
// integral_seq.c numerical integration - sequential code
#include <stdio.h>
#include <math.h>
#include <sys/time.h>

double get_wtime(void)
{
    struct timeval t;
    gettimeofday(&t, NULL);
    return (double)t.tv_sec + (double)t.tv_usec*1.0e-6;
}

double f(double x)
{
    return log(x)*sqrt(x);
}

// WolframAlpha: integral_1^4 log(x) sqrt(x) dx = 4/9 (4 log(64)-7)
// -> 4.28245881486164
int main(int argc, char *argv[])
{
    double a = 1.0;
    double b = 4.0;
    unsigned long const n = 1e9;
    const double dx = (b-a)/n;

    double S = 0;

    double t0 = get_wtime();
    for (unsigned long i = 0; i < n; i++) {
        double xi = a + (i + 0.5)*dx;
        S += f(xi);
    }
    S *= dx;
    double t1 = get_wtime();

    printf("Time=%lf seconds, Result=%.8f\n", t1-t0, S);

    return 0;
}
```

Ζητούμενο της άσκησης είναι η υλοποίηση του κώδικα χρησιμοποιώντας πολλαπλές διεργασίες και ουρές μηνυμάτων (message queues) για την απαραίτητη επικοινωνία μεταξύ τους. Ο αριθμός των διεργασιών θα πρέπει να ορίζεται δυναμικά, π.χ. να δίνεται ως όρισμα στη γραμμή εκτέλεσης ή να διαβάζεται κατά τον χρόνο εκτέλεσης.



Ερώτημα 3: Διαδιεργασιακή Επικοινωνία (15 Μονάδες)

Σε μια φαρμακοβιομηχανία υπάρχει μια γραμμή παραγωγής για τη συσκευασία φαρμάκων σε φιαλίδια, όπου λαμβάνουν χώρα οι παρακάτω εργασίες:

- Ανάγνωση προγραμματισμένης παραγόμενης ποσότητας φιαλιδίων με φάρμακο (E1)
- Παραγωγή γυάλινων φιαλιδίων (E2)
- Παραγωγή καπακίων για τα φιαλίδια (E3)
- Παραγωγή ετικετών για τα φιαλίδια και χαρτόκουτα (E4)
- Παραγωγή χαρτόκουτων συσκευασίας (E5)
- Απολύμανση γυάλινων φιαλιδίων και καπακίων (E6)
- Εισαγωγή ορού στα φιαλίδια (E7)
- Εισαγωγή δραστικής ουσίας στα φιαλίδια (E8)
- Τοποθέτηση καπακίων στα φιαλίδια (E9)
- Κόλληση ετικετών στα φιαλίδια (E10)
- Κόλληση ετικετών στις συσκευασίες χαρτόκουτου (E11)
- Συσκευασία των φιαλιδίων σε χαρτόκουτα (E12)

Θεωρήστε ότι για τις άνω αναφερόμενες εργασίες, η γραμμή παραγωγής προγραμματίζεται για την απαιτούμενη παραγόμενη ποσότητα φιαλιδίων με φάρμακο. Ακολουθεί η παραγωγή της ανάλογης ποσότητας, σύμφωνα με τον προγραμματισμό της παραγόμενης ποσότητας, των γυάλινων φιαλιδίων, των καπακίων, των ετικετών φιαλιδίων και των χαρτόκουτων συσκευασίας. Απολυμαίνονται τα γυάλινα φιαλίδια και τα καπάκια. Εισάγεται ο ορός στα φιαλίδια και στη συνέχεια η δραστική ουσία σε αυτά. Τοποθετούνται τα καπάκια στα φιαλίδια και μετά επικολλάται η ετικέτα τους. Επικολλώνται οι ετικέτες στα χαρτόκουτα συσκευασίας και πραγματοποιείται η συσκευασία ανά 12-άδες των φιαλιδίων με φάρμακο σε χαρτόκουτα.

- (α) Να σχεδιάσετε το γράφημα προτεραιότητας (precedence graph) που περιγράφει την εκτέλεση των ανωτέρω εργασιών με βάση τους περιορισμούς που προαναφέρθηκαν. **(4 Μονάδες)**
- (β) Να δώσετε τον παράλληλο ψευδοκώδικα συγχρονισμού των ανωτέρω εργασιών, χωρίς τη χρήση σημαφόρων. **(3 Μονάδες)**
- (γ) Να δώσετε, βασιζόμενοι στο γράφο προτεραιότητας που σχεδιάσατε και χρησιμοποιώντας κατάλληλους σημαφόρους, τον ψευδοκώδικα συγχρονισμού των ανωτέρω εργασιών. **(3 Μονάδες)**
- (δ) Χρησιμοποιήστε το ελάχιστο πλήθος σημαφόρων και ορθές αρχικοποιήσεις σε αυτούς **(5 Μονάδες)**

Υπόδειγμα: Για τη σχεδιαστική-θεωρητική προσέγγιση, χρησιμοποιήστε εντολές cobegin ... coend/ begin ... end, διαδικασίες P/ wait/ down) και V/ signal/ up και σημαφόρους. Για την υλοποίηση θα καλείτε τις διεργασίες E_i (όπου $i=1, \dots, 12$) χωρίς να χρειάζεται να υλοποιήσετε το περιεχόμενο τους. Αντ' αυτού για περιεχόμενο μπορείτε απλά να τυπώνετε ένα μήνυμα κειμένου



Ερώτημα 4: Χρονοπρογραμματισμός Διεργασιών (23 Μονάδες)

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

Διεργασία	Χρόνος άφιξης (ΧΑ)	Διάρκεια εκτέλεσης (ΧΕ)	PID
A	0	6	3
B	0	4	1
Γ	2	1	2
Δ	3	3	5
E	4	5	4
Z	5	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. Θεωρήστε πώς αν μια χρονική στιγμή μια διεργασία αφήνει την ΚΜΕ και την ίδια χρονική στιγμή καταφθάνει μια νέα διεργασία στο σύστημα, τότε η διεργασία που αφήνει την ΚΜΕ εισέρχεται πρώτη στην ουρά των διεργασιών. **(1 Μονάδα)**

Γενικά, μεταξύ των διεργασιών που ενδέχεται να αφιχθούν την ίδια χρονική στιγμή, θεωρήστε 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) Υπολογίστε το ποσοστό μεταβολής (αρνητικής/ θετικής) του μέσου χρόνου διεκπεραίωσης που φέρνει κάθε αλγόριθμος σε σχέση με το μέσο χρόνο διεκπεραίωσης του αλγόριθμου FCFS. **(1 Μονάδες)**

(δ) Σχεδιάστε έναν προεκχωρητικό αλγόριθμο χρονοδρομολόγησης διεργασιών, ο οποίος θα εξυπηρετεί πρώτες αυτές με το μεγαλύτερο χρόνο που απομένει, σε κάθε χρονική στιγμή άφιξης ή εκτέλεσης. 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 προτεραιότητας διεργασίας (PID- Priority ID) και σύμφωνα με τις ουρές αφίξεων και εναλλαγών. Να:

- σχεδιάσετε το διάγραμμα Gantt για τον LRTFP, επισημαίνοντας πότε ενεργοποιείται ο LRTFP, η προτεραιότητα και η επίδραση των ουρών στη σειρά εναλλαγής **(5 Μονάδες)**
- το μέσο χρόνο αναμονής του **(2 Μονάδες)**
- μέσο χρόνο απόκρισης **(2 Μονάδες)**
- μέσο χρόνο ολοκλήρωσης **(3 Μονάδες)**
- το πλήθος των θεματικών εναλλαγών των διεργασιών. **(2 Μονάδες)**