



Λειτουργικά Συστήματα

2023 - 2024

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 από 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 AM 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 2023-2024, Αναφορά Ομάδων»
- τα αρχεία του κώδικα σας σε μορφή zip στην εργασία «1ο project 2023-2024, 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 2023-2024 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

I. Για το τρέχον 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 >= 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 (30 Μονάδες, 5 μονάδες ανά υποερώτημα)

Να γράψετε ένα πρόγραμμα με χρήση του BASH shell για τη δημιουργία και συντήρηση ενός αρχείου επιχειρήσεων. Το πρόγραμμα θα πρέπει να δείχνει στον χρήστη ένα μενού μέσω του οποίου θα εκτελούνται οι κάτωθι λειτουργίες:

- [1] Επιλογή αρχείου επιχειρήσεων
- [2] Προβολή στοιχείων επιχείρησης
- [3] Αλλαγή στοιχείου επιχείρησης
- [4] Προβολή αρχείου
- [5] Αποθήκευση αρχείου
- [6] Έξοδος

Οι λειτουργίες θα εκτελούνται ως εξής:

- [1] Επιλογή αρχείου επιχειρήσεων

Το πρόγραμμα ζητά το path ενός αρχείου. Αν δεν δοθεί κάποιο όνομα θεωρείται σαν να έχει δοθεί το αρχείο **Businesses.csv** (βρίσκεται στο eclass μαζί με την εκφώνηση). Στο αρχείο αυτό βρίσκονται αποθηκευμένα τα στοιχεία των επιχειρήσεων (σε κάθε σειρά από ένας) με την κάτωθι μορφή:

[κωδικός], [όνομα], [οδός], [πόλη], [ταχ. κώδικας], [γεωγρ. μήκος], [γεωγρ. πλάτος]

- [2] Προβολή στοιχείων επιχείρησης

Το πρόγραμμα ζητά την είσοδο του κωδικού μιας επιχείρησης. Με βάση το στοιχείο αυτό, αναζητά τα στοιχεία της και τα δείχνει στην οθόνη.

- [3] Αλλαγή στοιχείου επιχείρησης

Το πρόγραμμα ζητά την είσοδο του κωδικού της επιχείρησης και του στοιχείου που χρειάζεται αλλαγή. Με βάση το στοιχείο αυτό, αλλάζει το στοιχείο και προβάλλει στην οθόνη τα παλαιά και τα νέα στοιχεία της επιχείρησης

- [4] Προβολή αρχείου



Το πρόγραμμα δείχνει στην οθόνη όλες τις επιχειρήσεις 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 (δείτε την εντολή more του Minix)

[5] Αποθήκευση αρχείου

Το πρόγραμμα ζητά το path ενός αρχείου στο οποίο θα αποθηκεύσει το πελατολόγιο. Αν δεν δοθεί κάποιο όνομα, το πελατολόγιο αποθηκεύεται στο αρχείο **Businesses.csv**.

[6] Έξοδος

Τερματισμός του προγράμματος.

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

Μην ξεχάσετε στο αρχείο που θα υποβάλλετε να δώσετε υπό τη μορφή σχολίου στην ΑΡΧΗ του κώδικα τα ΑΜ και ονόματα των ατόμων που συνεργάστηκαν για την υποβολή.

Παράδειγμα:

```
#####  
# 1234 Aristidis Ilias  
# 1235 Christos Makris  
#####  
  
#!/bin/bash
```

Ερώτημα 2: Διεργασίες (25 Μονάδες)

Υπολογισμός ολοκληρώματος συνάρτησης

Ο παρακάτω κώδικας υπολογίζει το ολοκλήρωμα της συνάρτησης $f(x) = \sin(\cos(x))$ στο διάστημα $[0, 1]$, χρησιμοποιώντας τη μέθοδο Monte Carlo. Ο κώδικας του ερωτήματος είναι διαθέσιμος στο αντίστοιχο αρχείο zip, στο e-class του μαθήματος. Στο αρχείο είναι διαθέσιμο το Makefile αλλά και δύο κενά αρχεία κώδικα τα οποία θα πρέπει να συμπληρώσετε.

```
// integral_mc_seq.c: Monte Carlo integration - sequential code  
// https://en.wikipedia.org/wiki/Monte_Carlo_integration  
#include <stdio.h>  
#include <stdlib.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;  
}  
  
inline double f(double x) {  
    return sin(cos(x));  
}  
  
// WolframAlpha: integral sin(cos(x)) from 0 to 1 = 0.738643  
// 0.73864299803689018  
// 0.7386429980368901838000902905852160417480209422447648518714116299  
  
int main(int argc, char *argv[]) {  
    double a = 0.0;  
    double b = 1.0;  
    unsigned long n = 24e7; // or e8  
    const double h = (b-a)/n;  
    const double ref = 0.73864299803689018;
```



```
const long tseed = 10;
double res = 0;
double t0, t1;

if (argc == 2) {
    n = atol(argv[1]);
}

srand48(tseed);

t0 = get_wtime();
for (unsigned long i = 0; i < n; i++) {
    double xi;
    xi = drand48();
    res += f(xi);
}
res *= h;
t1 = get_wtime();
printf("Result=%.16f Error=%e Rel.Error=%e Time=%lf seconds\n",
    res, fabs(res-ref), fabs(res-ref)/ref, t1-t0);
return 0;
}
```

Ζητούμενο της άσκησης είναι η υλοποίηση του κώδικα χρησιμοποιώντας πολλαπλές διεργασίες και κοινή/διαμοιραζόμενη μνήμη (shared memory) για την απαραίτητη επικοινωνία μεταξύ τους. Η κοινή μνήμη θα δημιουργηθεί με την κλήση συστήματος `mmap`. Οι διεργασίες πρέπει να διαμοιράζονται τις επαναλήψεις του βρόγχου, να υπολογίζουν το μερικό αποτέλεσμα και να χρησιμοποιούν την κοινή μνήμη για να ενημερώσουν τη διεργασία-πατέρα. 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.

Συγκεκριμένα, θα πρέπει να υλοποιηθούν οι παρακάτω δύο περιπτώσεις:

- `integral_mc_shm.c`: κάθε διεργασία-εργάτης θα γράψει το αποτέλεσμά της σε μία ξεχωριστή θέση ενός πίνακα που έχει δεσμευτεί σε κοινή μνήμη. Στο τέλος, η πατρική διεργασία θα αθροίσει τα μερικά αποτελέσματα για να υπολογίσει το συνολικό.
- `integral_mc_shm_sem.c`: κάθε διεργασία-εργάτης θα γράψει το αποτέλεσμά της σε μία μοναδική θέση μνήμης (π.χ. πίνακα ενός μόνο στοιχείου) που έχει δεσμευτεί σε κοινή μνήμη. Η ανανέωση της κοινής θέσης μνήμης αποτελεί κρίσιμη περιοχή που πρέπει να προστατευτεί με σηματοφόρο

Παρατηρήσεις:

- Βεβαιωθείτε πως ο πολυδιεργασιακός κώδικας τρέχει σωστά και ο χρόνος εκτέλεσης μειώνεται όταν χρησιμοποιούνται περισσότερες διεργασίες. Αυτό προϋποθέτει πως το υπολογιστικό σύστημα θα είναι αδρανές, άρα θα τρέχει μόνο το δικό σας εκτελέσιμο, και πως ο αριθμός των διεργασιών-εργατών δεν είναι μεγαλύτερος από τον αριθμό των διαθέσιμων επεξεργαστικών πυρήνων.
- Κάθε διεργασία-εργάτης πρέπει να αρχικοποιήσει τη γεννήτρια τυχαίων αριθμών με διαφορετικό seed.

Ερώτημα 3: Συγχρονισμός Διεργασιών (20 Μονάδες)

Σε μια εστία προσέρχονται οι ένοικοι στο εστιατόριο για το γεύμα και κάθονται γύρω από ένα μοναδικό και μεγάλο τραπέζι. Όποιος πεινάει βάζει στο πιάτο του μια μερίδα φαγητό από μια μεγάλη κατσαρόλα προετοιμασίας του φαγητού και σερβιρίσματος η οποία είναι τοποθετημένη στο κέντρο του τραπεζιού και από τους συνδαιτημόνες, μόνο ένας μπορεί να σερβιρίρεται κάθε φορά. Η κατσαρόλα αυτή χωράει μέχρι **ΣΧ+5** μερίδες φαγητό. Εάν η κατσαρόλα αδειάσει ενημερώνεται ο μάγειρας να ετοιμάσει πάλι φαγητό και για να γεμίσει η κατσαρόλα χρειάζεται **ΣΥ+3** κιλά φασόλια και **ΣΥ** κιλά καρότα και στην κουζίνα δεν έχει απόθεμα των υλικών, τα οποία πρέπει να έρθουν από την κατάλληλη- για κάθε υλικό- αποθήκη της εστίας. Ο μάγειρας έχει δυο βοηθούς, καθένας από τους οποίους ενημερώνεται να πάει σε διαφορετική αποθήκη για να φέρει τα υλικά που χρειάζονται για το μαγείρεμα και κάθε βοηθός μπορεί να μεταφέρει από την αποθήκη ευθύνης του μέχρι **ΣΥ** κιλά υλικών, ενώ η κάθε αποθήκη έχει απεριόριστη ποσότητα από κάθε υλικό. Ο μάγειρας για να ετοιμάσει το φαγητό πρέπει να έχει όλα τα υλικά για να γεμίσει η κατσαρόλα και μέχρι να συγκεντρωθούν τα υλικά αυτά περιμένει.

Να γράψετε ψευδοκώδικα που να υλοποιεί και να συγχρονίζει τη διαδικασία εστίασης που περιγράφηκε για τους σητιζόμενους, τη δουλειά του μάγειρα και των βοηθών του.

Ο αριθμός **ΣΧ** αντιστοιχεί στο ακέραιο μέρος του μέσου όρου των τελευταίων ψηφίων του επταψήφιου (νέου για τους παλιούς φοιτητές) αριθμού μητρώου των μελών της ομάδας σας και ο **ΣΥ** στον αντίστοιχο μέσο όρο του τρίτου



από δεξιά προς τα αριστερά ψηφίο για τα μέλη της ομάδας σας, οπότε για κάποιο μέλος ο αντίστοιχος **X** και **Y** θα είναι για παράδειγμα αν AM 1234567 τότε $X=7$ και $Y=5$ (1234**Y**6**X**) και θα πρέπει να υπολογίσετε τον μέσο όρο των αντίστοιχων αριθμών για τα μέλη της ομάδας σας.

Υπόδειγμα: Θα υλοποιήσετε τόσες διεργασίες όσες είναι οι οντότητες που συμμετέχουν στην εστίαση για να ολοκληρωθεί σωστά. Η συνάρτηση που υλοποιεί το σερβίρισμα κάθε συνδαιτημόνα θα ονομάζεται **take_food()**; η συνάρτηση για το μαγείρεμα θα ονομάζεται **cook_food()**; και για τη μεταφορά των υλικών από τους βοηθούς **helper()**; και θα τις καλείτε μέσα από τις διεργασίες χωρίς να χρειάζεται να υλοποιήσετε το περιεχόμενο της.

Ερώτημα 4: Χρονοπρογραμματισμός Διεργασιών (25 Μονάδες)

Θεωρήστε τέσσερις (4) 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	8	3
B	3	4	1
Γ	4	4	2
Δ	5	5	5
E	7	7	4

(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 αλγορίθμους υπολογίστε:

- το μέσο χρόνο αναμονής **(4 Μονάδες)**
- μέσο χρόνο απόκρισης **(4 Μονάδες)**
- μέσο χρόνο ολοκλήρωσης **(4 Μονάδες)**
- το πλήθος των θεματικών εναλλαγών των διεργασιών **(4 Μονάδες)**

(γ) Θεωρούμε μια νέα παραλλαγή της εκτέλεση του παραπάνω RR στην οποία ο χρόνος θεματικής εναλλαγής των διεργασιών δε είναι αμελητέος. Να υπολογίσετε το χρόνο εναλλαγής για τον RR **(2,5 Μονάδες)**, σχεδιάζοντας το νέο διάγραμμα Gantt για αυτόν **(2,5 Μονάδες)**, λαμβάνοντας υπόψη ότι ο μέσος χρόνος αναμονής (MXA) στην περίπτωση του RR είναι τριπλάσιος από τον MXA στην περίπτωση του SRTF.