

Λειτουργικά Συστήματα

Απαντήσεις 2ης εργασίας

Μάριος Στεφανίδης (1067458) & Μαύρα Πολυδώρου (1064885)

Μέρος Α

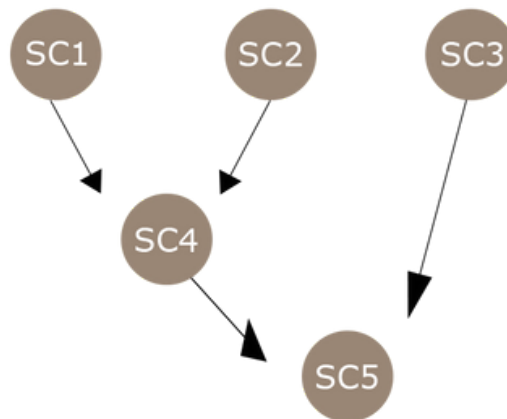
Ερώτημα Α: Ο κώδικας που υλοποιεί το ζητούμενο περιλαμβάνεται στο ίδιο Archive με αυτό το pdf στον φάκελο MerosA_QuestionA και περιέχει λεπτομερή σχόλια.

Ερώτημα Β: Ο κώδικας που υλοποιεί το ζητούμενο περιλαμβάνεται στο ίδιο Archive με αυτό το pdf στον φάκελο MerosA_QuestionB και περιέχει λεπτομερή σχόλια.
Κάποιες παρατηρήσεις:

- Η μεταβλητή data είναι αυτή που θα προσπελαύνει και θα διαβάζει κάθε ανάγνωση που θα πραγματοποιείται ενώ κάθε εγγραφή θα διπλασιάζει το περιεχόμενο της. Το αποτέλεσμα έπειτα από κάθε ενέργεια εμφανίζεται με printf().
- Ο κώδικας έχει υλοποιηθεί σύμφωνα με αυτά που έχουν αναφερθεί στις διαλέξεις, ώστε αναγνώστες και εγγραφείς να προσπελαύνουν κάθε φορά την "βάση" εναλλάξ.
- Για την επίλυση του προβλήματος χρησιμοποιήσαμε 2 εγγραφείς και 2 αναγνώστες.

```
marios@SINGLES:~/Desktop/Marios - CEID/Λειτουργικά Συστήματα/Project2/MerosA_Que  
stionB$ ./code  
Reader 1: read data as 1  
Writer 1 modified data to 2  
Reader 2: read data as 2  
Writer 2 modified data to 4
```

Ερώτημα Γ: Αρχικά, σύμφωνα με τους δοθέντες περιορισμούς προκύπτει το παρακάτω γράφημα προτεραιοτήτων:



Ο κώδικας που υλοποιεί το ζητούμενο περιλαμβάνεται στο ίδιο Archive με αυτό το pdf στον φάκελο MerosA_QuestionC. Κάποιες παρατηρήσεις:

- Χρησιμοποιήθηκαν **2 σημαφόροι (S12 και S34)**, οι οποίοι αρχικοποιήθηκαν με -1, προκειμένου να εξασφαλιστεί ο ελάχιστος δυνατός αριθμός σημαφόρων. Ο S12 αφορά τις κλήσεις συστήματος SC1, SC2 και ο S34 τις SC3, SC4.
- Σύμφωνα με το παραπάνω γράφημα οι κλήσεις συστήματος ομαδοποιούνται ως εξής:

Διεργασία P

begin
SC1;
SC3;
SC5;
end

Διεργασία Q

begin
SC2;
SC4;
end

```
marios@SINGLES:~/Desktop/Marios - CEID/Λειτουργικά Συστήματα/Project2/MerosA_Quest
stionCS$ ./semaphores
This is SC2
This is SC1
This is SC4
This is SC3
This is SC5
Child13778 terminated with return value 0
marios@SINGLES:~/Desktop/Marios - CEID/Λειτουργικά Συστήματα/Project2/MerosA_Quest
ionCS$
```

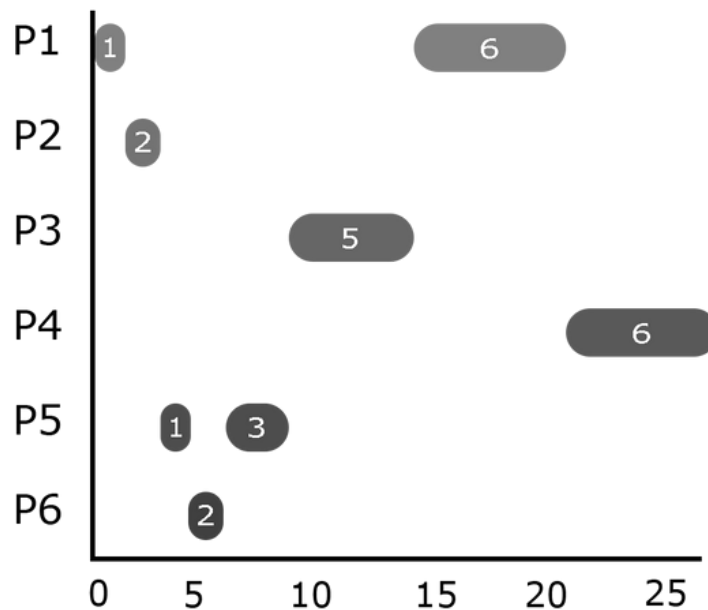
Ερώτημα Δ:

Το πρόβλημα δημιουργούται εξαιτίας του τρόπου με τον οποίο ελέγχονται και λειτουργούν οι δύο διαμοιραζόμενες μεταβλητές στη διαδικασία `Leave_p(free_p)`. Υποθέτουμε πως σε κάποια χρονική στιγμή δεν υπάρχουν ελεύθερες θέσεις και ένας πελάτης εισέρχεται για έκδοση εισιτηρίου σταύθμευσης. Η αντίστοιχη διεργασία (έστω K) θα εκτελέσει την διαδικασία `Enter_p()` και θα μπλοκαριστεί στην πρώτη κρίσιμη περιοχή υπό συνθήκη `await(free_s > 0)`, αφού `free_s = 0` εξαιτίας της υπόθεσης. Αν αργότερα, κάποιο σταθμευμένο αυτοκίνητο αναχωρήσει, τότε η αντίστοιχη διεργασία (έστω Λ) θα εκτελέσει την διαδικασία `Leave_p(free_p)` και η μεταβλητή `free_s` θα γίνει ίση με 1. Ας υποθέσουμε πως η διεργασία Λ χάνει τον έλεγχο της CPU πριν εκτελεστεί η δεύτερη κρίσιμη περιοχή της διαδικασίας `Leave_p(free_p)`, δηλαδή πριν χαρακτηριστεί η συγκεκριμένη θέση σταύθμευσης ως ελεύθερη. Αυτό θα έχει ως αποτέλεσμα, η διεργασία K να αφυπνιστεί, να θέσει την μεταβλητή `free_s` ίση με 0 και να εκτελέσει την δεύτερη κρίσιμη περιοχή της διαδικασίας `Enter_p()`. Ωστόσο, η συνάρτηση `Επιλογή_Θέσης()` δεν θα επιστρέψει μια έγκυρη θέση αφού προηγουμένως η διεργασία Λ δεν κατάφερε να ενημερώσει τον πίνακα `free_a[]`. Συνεχίζοντας, ο έλεγχος θα περάσει στη διεργασία Λ , η οποία θα ολοκληρώσει την εκτέλεση της διαδικασίας `Leave_p(free_p)` εκτελώντας την δεύτερη κρίσιμη περιοχή. Παρόλο βέβαια, σύμφωνα με τα παραπάνω, που έχει ελευθερωθεί μια θέση, αυτή δεν κατέστη δυνατόν να διατεθεί στον πελάτη που περιμένε.

Ο κώδικας που υλοποιεί το ζητούμενο περιλαμβάνεται στο ίδιο Archive με αυτό το pdf στον φάκελο `MerosA_QuestionD` και περιέχει λεπτομερή σχόλια.

Μέρος Β

Ερώτημα Α: Στο παρακάτω διάγραμμα Gantt υποδεικνύουμε τον τρόπο με τον οποίο θα εκτελεστούν οι διεργασίες σύμφωνα με τον αλγόριθμο SRTF (Shortest Remaining Time First):



Ο πίνακας στον οποίο έχουν καταγραφεί τα ζητούμενα περιλαμβάνεται στο ίδιο Archive με αυτό το pdf στον φάκελο MerosB_QuestionA.

Ερώτημα Β:

α):

Μέγεθος σελίδας/πλαisiού $\rightarrow 1\text{kbytes} = 2^{10}\text{ bytes} = 1024\text{ bytes}$

Διεργασία $\rightarrow 6100\text{ bytes}$

Βρίσκω πόσα frames θα χρειαστούν $\rightarrow 6100/1024 = 5.957 \approx 6$

Αυτό σημαίνει πως θα χρειαστούν 6 frames, συνεπώς στο τελευταίο frame η διεργασία θα καταλαμβάνει: $6100 - 5 \times 1024 = 980\text{ bytes}$. Οπότε, η εσωτερική κλασματοποίηση ισούται με $1024 - 980 = \mathbf{44\text{ bytes}}$.

β): Παρακάτω παρατίθεται ο πίνακας με τις δοθέντες εγγραφές

Αριθμός Σελίδας	Αριθμός Πλαισίου
00	0B
01	0C
02	01
03	0F
04	-
05	08

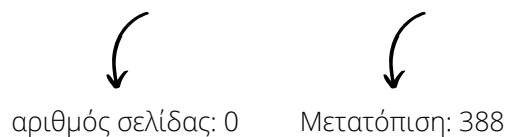
Μέγεθος σελίδας/πλαίσιου $\rightarrow 1 \text{ kbytes} = 2^{10} \text{ bytes}$

Εύρος λογικών διευθύνσεων $\rightarrow 20 \text{ bits}$

Φυσική μνήμη συστήματος $\rightarrow 4 \text{ MB} = 2^{20} \times 2^2 = 2^{22} \text{ bytes}$

Αριθμός σελίδας $= 20 - 10 = 10 \text{ bits}$

$(00388)_H = 0000 \ 0000 \ 0011 \ 1000 \ 1000$



Συνεπώς, η ζητούμενη φυσική διεύθυνση είναι: **002F88**.

$(0125F)_H = 0000 \ 0001 \ 0010 \ 0101 \ 1111$



Η συγκεκριμένη λογική σελίδα δεν βρίσκεται στην φυσική μνήμη.

$(015A4)_H = 0000 \ 0001 \ 0101 \ 1010 \ 0100$



Συνεπώς, η ζητούμενη φυσική διεύθυνση είναι: **0021A4**.

Ερώτημα Γ: Παρακάτω παρατίθεται ο ζητούμενος πίνακας.

Η ακολουθία αναφοράς είναι η εξής:

2	5	8	1	8	7	5	1	8	2	4	2	1	3	6	4	7	5	3	7
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	<u>2</u>	2	2	2	2	<u>7</u>	7	7	7	<u>2</u>	2	2	2	2	2	<u>4</u>	4	4	4	4
1		<u>5</u>	5	5	5	5	5	5	5	5	<u>4</u>	4	4	4	<u>6</u>	6	6	6	<u>3</u>	3
2			<u>8</u>	8	8	8	8	8	8	8	8	8	8	<u>3</u>	3	3	3	<u>5</u>	5	5
3				<u>1</u>	1	1	1	1	1	1	1	1	1	1	1	1	<u>7</u>	7	7	7

Υπογραμμίζονται και βρίσκονται σε κόκκινο φόντο τα δεδομένα που ήρθαν στην μνήμη εκείνη την χρονική στιγμή.