

Αναφορά εργαστηριακής άσκησης 1

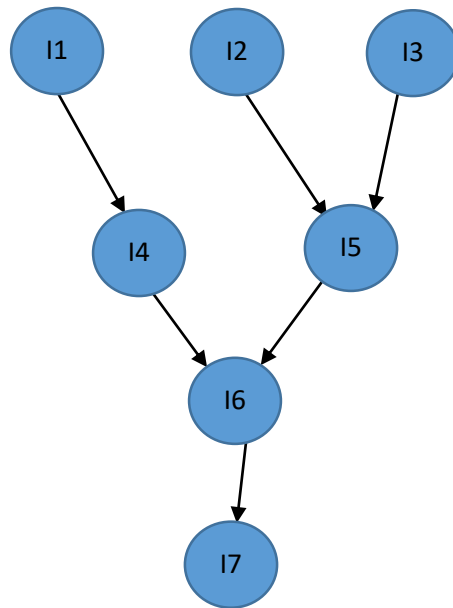
Δελημιχάλης Αλέξανδρος Α.Μ. 1054324

Κωστορρίζος Δημήτριος Α.Μ. 1054419

Θέμα 2

Ερώτημα Α

i)



ii)

cobegin

begin

cobegin

I2;

I3;

coend

I5;

end

begin

```

        l1;

        l4;

    end
coend

    l6;

    l7;

iii)
binary semaphore s1,4 = 0, s2,5 = 0, s3,5 = 0, s4,6 = 0, s5,6 = 0, s6,7 = 0;
cobegin;

    begin l1; up(s1,4); end;

    begin l2; up(s2,5); end;

    begin l3; up(s3,5); end;

    begin; down(s1,4); l4; up(s4,6); end;

    begin; down(s2,5); down(s3,5); l5; up(s5,6); end;

    begin; down(4,6); down(5,6); l6; up(s6,7); end;

    begin; down(s6,7); l7; end;

coend;

```

Ερώτημα Β

i) Τα τελικά αποτελέσματα που είναι δυνατόν να εμφανιστούν μετά το πέρας της παράλληλης εκτέλεσης των διεργασιών Δ1 και Δ2 είναι, το “ΧΥ”, το “ΥΧ”, το “Υ” και το κενό “ ”, ανάλογα με την σειρά εκτέλεσης της κάθε εντολής, όπως φαίνεται στα παρακάτω σενάρια εκτέλεσης:

- “ΧΥ” -> x = 1; x = 2; y = 2; y = 2; print “Χ”; print “Υ”;
- “ΥΧ” -> x = 1; x = 2; y = 2; y = 2; print “Υ”; print “Χ”;
- “Υ” -> x = 2; y = 2; print “Υ”; x = 1; y = 2;
- “ ” -> x = 2; y = 2; x = 1; y = 2;

ii)

binary semaphore s1 = 0, s2=0, s3=0;

Διεργασία Δ1	Διεργασία Δ2
x = 1;	down(s1);

<pre>up(s1); down(s2); y = 2; if (x == y) print "X"; up(s3);</pre>	<pre>x = 2; y = 2; up(s2); down(s3); if (x == y) print "Y";</pre>
--	---

Αρχικά, εκτελείται η εντολή $x = 1$; και έπειτα αυξάνεται ο σημαφόρος $s1$ στην τιμή 1, προκειμένου να κατέβει στην τιμή 0 από την διεργασία Δ2 και να εκτελεστούν οι εντολές $x = 2$; $y = 2$; . Με την ολοκλήρωσή τους, αυξάνεται ο σημαφόρος $s2$ στην τιμή 1, με σκοπό να συνεχιστεί η εκτέλεση της διεργασίας Δ1, κατεβάζοντάς τον στην τιμή 0, εκτυπώνοντας το "X" και αυξάνοντας τον σημαφόρο $s3$. Έπειτα, εκτελείται το υπόλοιπο τμήμα της διεργασίας Δ2, η οποία κατεβάζει τον σημαφόρο $s3$ στην τιμή 0 και εκτυπώνει το "Y", εμφανίζοντας έτσι το επιθυμητό αποτέλεσμα "XY".

Ερώτημα Γ

Η διεργασία Δ0 αλλάζει αρχικά το $Flag[0]$ σε True παράλληλα το ίδιο κάνει και η Δ1 για το $Flag[1]$. Έπειτα, η Δ0 θέτει το $turn[0]$ σε 0 και η Δ1 το $turn[1]$ σε 1, άρα η Δ0 ξεφεύγει πρώτη από το βρόχο λόγω της δεύτερης συνθήκης, εκτελείται το κρίσιμο τμήμα της και όταν τελειώσει η εκτέλεση της αλλάζει το $flag[0]$ σε false. Έπειτα ξεφεύγει η Δ1 από τον βρόχο και λόγω της πρώτης συνθήκης και εκτελείται το κρίσιμο τμήμα της. Παράλληλα η Δ0 αλλάζει ξανά το $flag[0]$ σε true, και μετά το $turn[0]$ αυξάνεται σε 1 και είναι πλέον ίδιο με της Δ1, οπότε πλέον και οι δυο συνθήκες είναι αναληθείς. Έτσι, η Δ0 περιμένει την Δ1 να βγει από το κρίσιμο σημείο της, αλλάζει το $flag[1]$ σε false και μετά μπαίνει η ίδια στο κρίσιμο σημείο της. Με παρόμοιο τρόπο επιτυγχάνεται η παραπάνω διαδικασία, στην περίπτωση όπου αρχίζαμε με την διεργασία Δ1. Επομένως, εύλογα συμπεραίνουμε ότι εξασφαλίζεται αμοιβαίος αποκλεισμός για τις διεργασίες Δ0 και Δ1. Στην περίπτωση όπου εκτελεστεί οποιαδήποτε από τις δύο διεργασίες εξ' ολοκλήρου, χωρίς αναμείξεις τις άλλης, οι συνθήκες των εντολών until, δεν μπορούν να ισχύουν ταυτόχρονα, εξασφαλίζοντας έτσι τον αμοιβαίο αποκλεισμό των διεργασιών Δ1 και Δ2.

Θέμα 3

Ερώτημα Α

binary semaphore $s1,2 = 0, s1,3 = 0, s2,1 = 0, s2,3 = 0, s3,1 = 0, s3,2 = 0$;

Διεργασία Student_1	Διεργασία Student_2	Διεργασία Student_3
<pre>Search_Book(); up(s1,2); up(s1,3);</pre>	<pre>Search_Book(); up(s2,1); up(s2,3);</pre>	<pre>Search_Book(); up(s3,1); up(s3,2);</pre>

down(s2,1); down(s3,1); Study_Project();	down(s1,2); down(s3,2); Study_Project();	down(s1,3); down(s2,3); Study_Project();
--	--	--

Ερώτημα Β

Binary semaphore insert =1;

int flag=0;

Insertion()

While(TRUE)

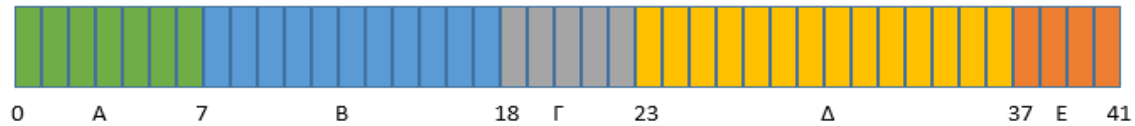
```
{
    down(insert);
    down(mutex);
    if(count==0)
        count = count + 1;
    flag=1;
    down(library);
    up(mutex);
    Insert_Book();
    if(flag==1)
        down(mutex);
        count = count - 1;
        if(count==0)
            Up(library);
        up(mutex);
        flag=0;
    Up(insert);
}
```

Χρησιμοποιούμε έναν επιπλέον σημαφόρο, τον δυαδικό σημαφόρο insert και μία επιπλέον μεταβλητή flag. Κατά την εκτέλεση της εντολής Insertion(), απενεργοποιείται ο σημαφόρος insert, προκειμένου να μην μπορούν να εκτελεστούν άλλες εντολές Insertion(). Ελέγχεται αν η

μεταβλητή count ισούται με 0, δηλαδή πόσες εντολές Search() εκτελούνται. Εάν δεν εκτελούνται, τότε αυξάνεται το count κατά 1, απενεργοποιείται ο σημαφόρος library και ενημερώνεται το flag γεγονότος. Έπειτα, εκτελείται το κρίσιμο τμήμα Insert_Book(); . Αν το flag ισούται με 1, δηλαδή έχει συμβεί η παραπάνω αλλαγή, τότε μειώνεται ξανά το count και η μεταβλητή flag επιστρέφει στην αρχική της τιμή. Αν, η μεταβλητή count ισούται με 0, δηλαδή δεν εκτελούνται άλλες εντολές Search(), τότε ενεργοποιείται ο σημαφόρος library, υποδηλώνοντας την απουσία ενεργών διεργασιών εκείνη την χρονική περίοδο.

Θέμα 4

- FCFS(First Come First Served)



$$X_{\Delta A} = 7 - 0 = 7$$

$$X_{A A} = 7 - 7 = 0$$

$$X_{\Delta B} = 18 - 3 = 15$$

$$X_{A B} = 15 - 11 = 4$$

$$X_{\Delta \Gamma} = 23 - 9 = 14$$

$$X_{A \Gamma} = 14 - 5 = 9$$

$$X_{\Delta \Delta} = 37 - 12 = 25$$

$$X_{A \Delta} = 25 - 14 = 11$$

$$X_{\Delta E} = 41 - 14 = 27$$

$$X_{A E} = 27 - 4 = 23$$

$$MX_{\Delta} = \frac{7 + 15 + 14 + 25 + 27}{5} = 17,6 \text{ χρονικές μονάδες}$$

$$MX_A = \frac{0 + 4 + 9 + 11 + 23}{5} = 9,4 \text{ χρονικές μονάδες}$$

- SJF(Shortest Job First)



$$X\Delta_A = 7 - 0 = 7$$

$$XA_A = 7 - 7 = 0$$

$$X\Delta_B = 18 - 3 = 15$$

$$XA_B = 15 - 11 = 4$$

$$X\Delta_\Gamma = 27 - 9 = 18$$

$$XA_\Gamma = 18 - 5 = 13$$

$$X\Delta_\Delta = 41 - 12 = 29$$

$$XA_\Delta = 29 - 14 = 15$$

$$X\Delta_E = 22 - 14 = 8$$

$$XA_E = 8 - 4 = 4$$

$$MX\Delta = \frac{7 + 15 + 18 + 29 + 8}{5} = 15,4 \text{ χρονικές μονάδες}$$

$$MXA = \frac{0 + 4 + 13 + 15 + 4}{5} = 7,2 \text{ χρονικές μονάδες}$$

- SRTF(Shortest Remaining Time First)



$$X\Delta_A = 7 - 0 = 7$$

$$XA_A = 7 - 7 = 0$$

$$X\Delta_B = 27 - 3 = 24$$

$$XA_B = 24 - 11 = 13$$

$$X\Delta_\Gamma = 14 - 9 = 5$$

$$XA_\Gamma = 5 - 5 = 0$$

$$X\Delta_\Delta = 41 - 12 = 29$$

$$XA_\Delta = 29 - 14 = 15$$

$$X\Delta_E = 18 - 14 = 4$$

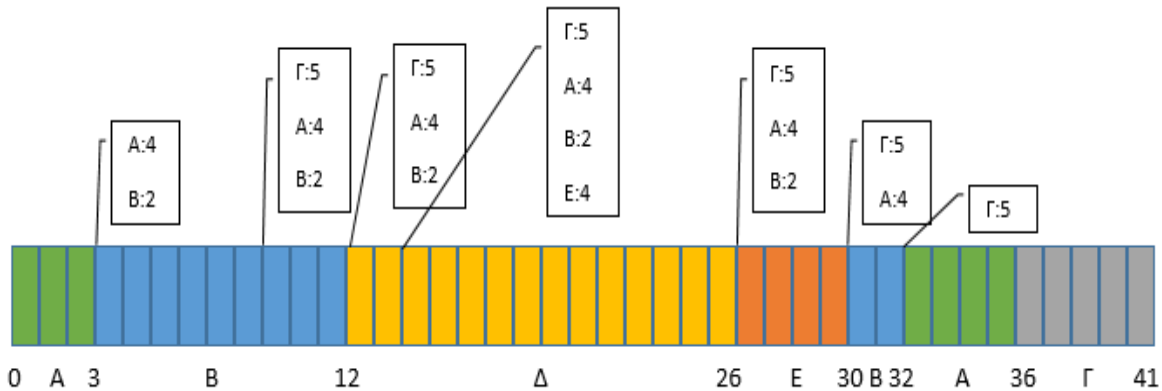
$$XA_E = 4 - 4 = 0$$

$$MX\Delta = \frac{7 + 24 + 5 + 29 + 4}{5} = 13,8 \text{ χρονικές μονάδες}$$

$$MXA = \frac{0 + 13 + 0 + 15 + 0}{5} = 5,6 \text{ χρονικές μονάδες}$$

- Priority Scheduling (non-preemptive priority)

Στα πλαίσια, παρουσιάζεται ο υπολειπόμενος χρόνος εκτέλεσης ανά διεργασία για συγκεκριμένες χρονικές στιγμές, ταξινομημένες κατά αύξουσα τιμή προτεραιότητας.



$$X_{\Delta_A} = 36 - 0 = 36$$

$$X_{A_A} = 36 - 7 = 29$$

$$X_{\Delta_B} = 32 - 3 = 29$$

$$X_{A_B} = 29 - 11 = 18$$

$$X_{\Delta_\Gamma} = 41 - 9 = 32$$

$$X_{A_\Gamma} = 32 - 5 = 27$$

$$X_{\Delta_\Delta} = 26 - 12 = 14$$

$$X_{A_\Delta} = 14 - 14 = 0$$

$$X_{\Delta_E} = 30 - 14 = 16$$

$$X_{A_E} = 16 - 4 = 12$$

$$MX_{\Delta} = \frac{36 + 29 + 32 + 14 + 16}{5} = 25,4 \text{ χρονικές μονάδες}$$

$$MX_A = \frac{29 + 18 + 27 + 0 + 12}{5} = 17,2 \text{ χρονικές μονάδες}$$

- Στα πλαίσια, παρουσιάζεται ο υπολειπόμενος χρόνος εκτέλεσης ανά διεργασία για συγκεκριμένες χρονικές στιγμές, όπως βρίσκονται στην βοηθητική ουρά διεργασιών .



$$ΜΧΑ = \frac{4 + 16 + 17 + 15 + 9}{5} = 12,2 \text{ χρονικές μονάδες}$$