**Μέρος 2**

**Ερώτημα Α:**

i)

ii)

cobegin

begin

cobegin

I2

I3

coend;

I5

end;

begin

I1

I4

end;

coend;

I6

I7

iii)

var s1,s2,s3,s4,s5,s6

s1=s2=s3=s4=s5=s6=0

cobegin

begin I1; up(s1); end;

begin I2; up(s2); end;

begin I3; up(s3); end;

begin down(s1); I4; up(s4); end;

begin down(s2); down(s3); I5; up(s5); end;

begin down(s4); down(s5); I6; up(s6); end;

begin down(s6); I7; end;

coend;

**Ερώτημα Β:**

i)

Τα δυνατά αποτελέσματα είναι: Υ, ΧΥ, ΥΧ, -

Το αποτέλεσμα Χ δεν είναι δυνατόν θα εμφανιστεί.

ii)

var s1,s2;

s1=0, s2=0

cobegin

begin begin

x=1 wait(s2)

signal(s2) x=2

y=2 signal(s1)

wait(s1) y=2

if(x==y) if(x==y)

print “X”; wait(s2)

signal(s2) print “Y”;

end; end;

coend;

**Ερώτημα Γ:**

Ο κώδικας μπορεί να γραφτεί και ως εξής:

Shared Boolean flag[2];

Shared turn[2];

**Δ0**

flag[0]=TRUE

turn[0]=turn[1]

while(flag[1]=TRUE and turn[0]=turn[1])

{ noop}

<ΚΡΙΣΗΜΟ ΤΜΗΜΑ>

flag[0]=FALSE;

**Δ1**

flag[1]=TRUE

turn[1]=1-turn[0]

while(flag[0]=TRUE and turn[1]≠turn[0])

{ noop}

<ΚΡΙΣΗΜΟ ΤΜΗΜΑ>

flag[1]=FALSE;

Ο παραπάνω κώδικας έτσι όπως έχει διαμορφωθεί είναι παρόμοιος με την λύση του Peterson ο οποίος επιτυγχάνει τον αμοιβαίο αποκλεισμό. Πιο συγκεκριμένα η τελευταία εντολή η οποία αποτελεί το leave\_region κομμάτι του κώδικα είναι κοινή με την αντίστοιχη που έχουμε στην λύση του Peterson. Αντί για other και process έχουμε τα 1 και 0 αντίστοιχα. Παρατηρούμε ότι και το enter\_region είναι αντίστοιχο καθώς στην θέση της εντολής other=1-process που χρησιμοποιείται στην λύση του Peterson ο παραπάνω κώδικας βάζει 0 και 1 στα flag και turn με παρόμοιο τρόπο. Συνοψίζοντας η εντολή flag[0]=TRUE(ίδια με την interested[process]=TRUE) μπλοκάρει την άλλη διεργασία και η turn[0]=turn[1](turn=process) μπλοκάρει τον εαυτό της. Συνεπώς αφού ο παραπάνω κώδικας ταυτίζεται με την λύση του Peterson πετυχαίνει αμοιβαίο αποκλεισμό.

**Θέμα 3ο**

**Ερώτημα Α:**

Var s2, s3, done

s2=0, s3=0, done

cobegin

**Student\_1 Student\_2 Student\_3**

wait(s2) signal(s2) signal(s3)

wait(s3) wait(done) wait(done)

signal(done)

signal(done)

**Ερώτημα B:**

Insertion:

Binary semaphore insert=1;

while(TRUE) {

down(insert);

down(mutex);

count = count +1;

if(count==1) down(library);

up(mutex);

Insert\_Book();

down(mutex);

count=count-1;

if(count==0) up(library);

up(mutex);

up(insert);

}

Ο σημαφόρος insert δεν επιτρέπει άλλες εισαγωγές όσο μια είναι ήδη ενεργή. Οι δυο κρίσιμες περιοχές επιτρέπουν την παραλληλία και τον κατάλληλο συγχρονισμό με τις Searching και Deletion.

**Θέμα 4ο**

**First Come First Serve**: Οι διεργασίες θα εκτελεστούν με την σειρά που φτάνουν στο σύστημα δηλαδή Α, Β, Γ, Δ, Ε.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Α | Β | Γ | Δ | Ε |

7 18 23 37 41

Εάν t1= χρονική στιγμή άφιξης και t2=χρονική στιγμή εξόδου και tcpu=χρόνος που χρειάζεται η διεργασία για εξυπηρέτηση από την CPU: ΧΔ=t2-t1 ΧΑ=t2-t1-tcpu

ΧΔΑ=7-0=7

ΧΔΒ=18-3=15

ΧΔΓ=23-9=14

ΧΔΔ=37-12=25

ΧΔΕ=41-14=27

ΜΧΔ=(7+15+14+25+27)/5

**ΜΧΔ=17,6**

ΧΑΑ=7-0-7=0

ΧΑΒ=18-3-11=4

ΧΑΓ=23-9-5=9

ΧΑΔ=37-12-14=11

ΧΑΕ=41-14-4=23

ΜΧΔ=(0+4+9+11+23)/5

**ΜΧΑ=9,4**

**Shortest Job First**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | Β | Ε | Γ | Δ |

7 18 22 27 41 ΧΔΑ=7-0=7

ΧΔΒ=18-3=15

ΧΔΓ=27-9=18

ΧΔΔ=41-12=29

ΧΔΕ=22-14=8

ΜΧΔ=(7+15+18+29+8)/5

**ΜΧΔ=15,4**

ΧΑΑ=7-0-7=0

ΧΑΒ=18-3-11=4

ΧΑΓ=27-9-5=13

ΧΑΔ=41-12-14=15

ΧΑΕ=22-14-4=4

ΜΧΔ=(0+4+13+15+4)/5

**ΜΧΑ=7,2**

**Shortest Remaining Time First**:

Γ:2 Β:9

Α:4 Β:9 Β:9 Δ:14 Β:9

Β:11 Β:11 Γ:5 Δ:14 Ε:4 Δ:14 Δ:14

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | A | B | Γ | Γ | Ε | Β | Δ |

ΧΔΑ=7-0=7

ΧΔΒ=27-3=24

ΧΔΓ=14-9=5

ΧΔΔ=41-12=29

ΧΔΕ=18-14=4

ΜΧΔ=(7+24+5+29+4)/5

**ΜΧΔ=13,8**

ΧΑΑ=7-0-7=0

ΧΑΒ=(7-3)+(18-9)=13

ΧΑΓ=14-9-5=0

ΧΑΔ=41-12-14=15

ΧΑΕ=18-14-4=0

ΜΧΔ=(13+15)/5

**ΜΧΑ=5,6**

**Priority Scheduling-non preemptive**:

ΧΔΑ=7-0=7

ΧΔΒ=18-3=15

ΧΔΓ=41-9=32

ΧΔΔ=32-12=20

ΧΔΕ=36-14=22

ΜΧΔ=(7+15+32+20+22)/5

**ΜΧΔ=19,2**

ΧΑΑ=7-0-7=0

ΧΑΒ=18-11-3=4

ΧΑΓ=41-9-5=27

ΧΑΔ=32-12-14=6

ΧΑΕ=36-14-4=18

ΜΧΔ=(0+4+27+6+18)/5

**ΜΧΑ=11**

**Round Robin**:

Β:3 Γ:1 Δ:10

Ε:4 Β:3 Γ:1 Δ:10

Α:3 Β:7 Γ:5 Δ:14 Ε:4 Β:3 Γ:1 Δ:10

Β:11 Α:3 Β:7 Γ:5 Δ:14 Ε:4 Β:3 Γ:1 Δ:10 Δ:6 Δ:2

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | A | Β | Γ | Δ | Ε | Β | Γ | Δ | Δ | Δ |

ΧΔΑ=11-0=11

ΧΔΒ=30-3=27

ΧΔΓ=31-9=22

ΧΔΔ=41-12=29

ΧΔΕ=27-14=13

ΜΧΔ=(11+27+22+29+13)/5

**ΜΧΔ=20,4**

ΧΑΑ=8-4=4

ΧΑΒ=(11-8)+(4-3)+(27-15)=16

ΧΑΓ=(15-9)+(30-19)=17

ΧΑΔ=(19-12) +(31-23)=15

ΧΑΕ=(23-14)=9

ΜΧΔ=(4+16+17+15+9)/5

**ΜΧΑ=12,2**