

Δυνάμεις του 2

$$\begin{aligned}
 2^0 &= 1 \\
 2^1 &= 2 \\
 2^2 &= 4 \\
 2^3 &= 8 \\
 2^4 &= 16 \\
 2^5 &= 32 \\
 2^6 &= 64 \\
 2^7 &= 128 \\
 2^8 &= 256 \\
 2^9 &= 512 \\
 2^{10} &= 1024 = 1K \\
 2^{11} &= 2048 = 2K
 \end{aligned}$$

$$\begin{aligned}
 2^{12} &= 4096 = 4K \\
 2^{13} &= 8192 = 8K \\
 2^{14} &= 16384 = 16K \\
 2^{15} &= 32K \\
 2^{16} &= 64K \\
 2^{17} &= 128K \\
 2^{18} &= 256K \\
 2^{19} &= 512K \\
 2^{20} &= 1024K = 1M \\
 2^{21} &= 2M \\
 2^{22} &= 4M \\
 2^{23} &= 8M \\
 2^{24} &= 16M
 \end{aligned}$$

$$\begin{aligned}
 2^{25} &= 32M \\
 2^{26} &= 64M \\
 2^{27} &= 128M \\
 2^{28} &= 256M \\
 2^{29} &= 512M \\
 2^{30} &= 1024M = 1G \\
 2^{31} &= 2G \\
 2^{32} &= 4G \\
 2^{33} &= 256G \\
 2^{34} &= 1T \\
 2^{35} &= 1P
 \end{aligned}$$

Deca	da	10^1	10
Hecto	H	10^2	100
Kilo	K	10^3	1.000
Mega	M	10^6	1.000.000
Giga	G	10^9	1.000.000.000
Tera	T	10^{12}	1.000.000.000.000

1 Producer	2 consumer	ΙΟΥΝΙΟΣ 2018 (έχω χαρτί μισπίνο)
2 Producer	2 consumer	ΙΟΥΝΙΟΣ 2020 (δίκα 3, έχω χαρτί μισπίνο)
2 Producer	3 consumer	ΤΕΛΕΥΤΑΙΑ ΓΕΓΓΙΔΑ
2 Producer	1 consumer	ΣΕΠΤΕΜΒΡΙΟΣ 2023 δίκα 2 (ημέρα 6/20) ΣΕΠΤΕΜΒΡΙΟΣ 2023 δίκα 2 (ημέρα 6/20)

ΘΕΜΑΤΑ

Σεντέρπιος 2023	✓
Σεντέρπιος 2021	✓
Ιούνιος 2023	✓
Ιούνιος 2018	✓
Ιούνιος 2020	✓
Σεντέρπιος 2020	✓
Σεντέρπιος 2022	✓
Φεβρουάριος 2022?	
Φεβρουάριος 2023	✓
Φεβρουάριος 2021(εινυζό)	

ΚΩΔΙΚΑΣ

Για να προστασέψει το
εργαστήριο σημάτα τε semaphore
Semaphore: mutex = 1

C1
down(mutex)
// κρίσιμη σημάτα

up(mutex)

C2

down(mutex)
// κρίσιμη σημάτα
up(mutex)

- Και έχει C1, C2 δευτεροβάθμια προτοτάξη για την εκτέλεση

Χρονοδιαγράμμην

ATT → Μέσος χρόνος Εντεροφής

WTT → Μέσος Αντίτερος χρόνος Εντεροφής

WT → Μέσος χρόνος Ανατονής

WW → Μέσος Αντίτερος χρόνος Ανατονής

FCFS-FIFO

TT_{i,...} = χρόνος ολοκλήρωσης - χρόνος αρχής

UTT_{i,...} = TT_{i,...} / χρόνος εκτέλεσης (run time)

WT_{i,...} = πότε σειράσει τη διεργασία - αρχή διεργασίας (•)
(δηλαδή πότε χρόνο λειτένει από την αρχή της)

WTI_{i,...} = WT_{i,...} / χρόνος εκτέλεσης (run time)

SRTN*

* Όσοι οι άλλοι σύνορι μέσοι τε FIFO * και [Round Robin]

WT_{i,...} = TT_{i,...} - χρόνος εκτέλεσης (run time)

priority < 20
 $Q = (140 - \text{priority}) * 5$
 else:
 if priority < 120

Σεντέρμπρος 2023

Όρια 10

Σινεργία είναι γύρουσα Linux και 4 διεργασίες A, B, Γ, Δ τε αριθμούς προτεραιοτήτων 103, 105, 116, 126 είναι αυξιζόμενα. Οι διεργασίες μαζεύθηκαν σε χρόνο t=0 τε ανθεκτική σειρά. Να διεβεβαιωθεί τον χρόνο τελετών κάθε διεργασίας και προστασία από αυτές έχει χρόνο Εκτέλεσης (Run Time) 1100ms. Ο χρόνος υπολογιστή των πειραιών αυτονομείται στην πολιτική του D(1) χρονοδρομούσης και οι διεργασίες είναι CPU bound (δεν μπαίνουν 1/10)

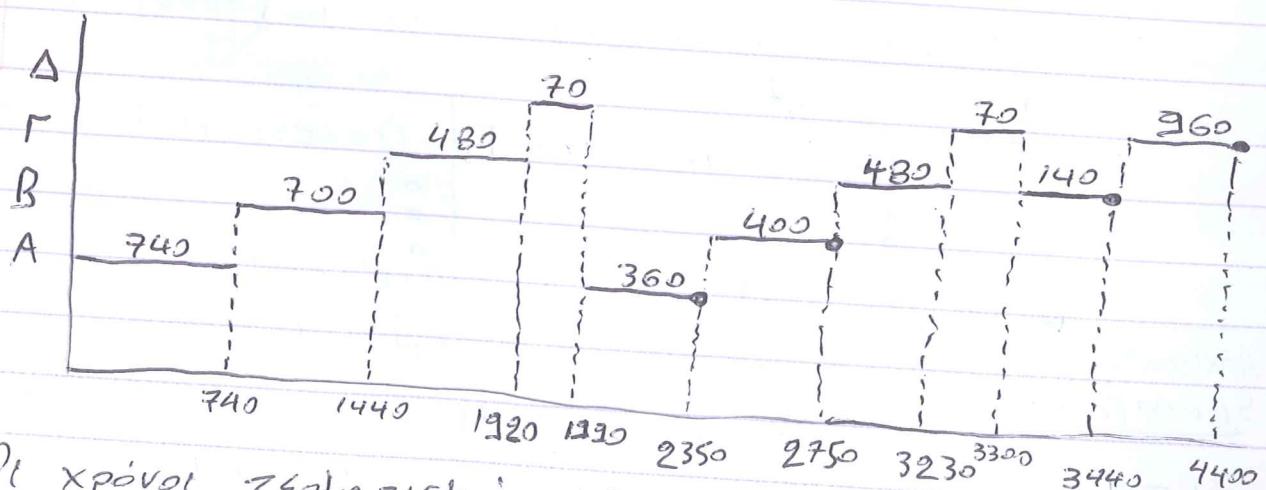
Λύση:

$$103 : A = (140 - 103) * 20 = 37 * 20 = 740 \text{ ms}$$

$$105 : B = (140 - 105) * 20 = 35 * 20 = 700 \text{ ms}$$

$$116 : \Gamma = (140 - 116) * 20 = 24 * 20 = 480 \text{ ms}$$

$$126 : \Delta = (140 - 126) * 5 = 14 * 5 = 70 \text{ ms}$$



Πιο χρόνοι τελετών κάθε διεργασίας είναι:

$$A \rightarrow 2350$$

$$B \rightarrow 2750$$

$$\Gamma \rightarrow 3440$$

$$\Delta \rightarrow 4400$$

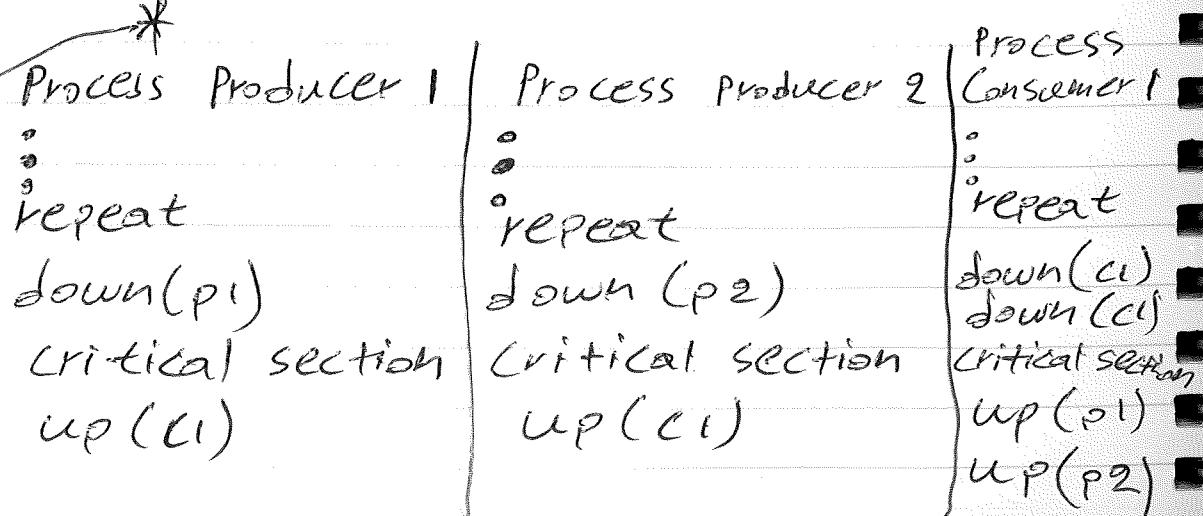
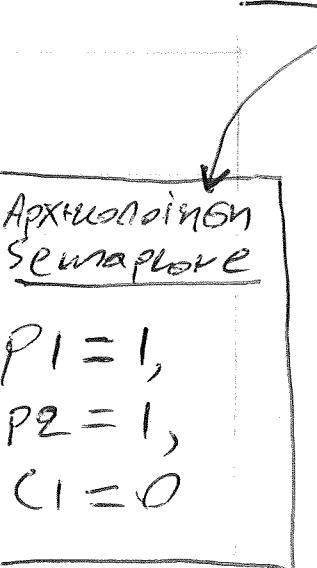
Θέμα 20

Έγω ότι δίνεται η ανάγκη για προβλημάτων για παραγωγή και μαζανάτωση τε συ παραγωγής και ένα μαζανάτωση που επεξεργάζεται σε αύξηψη και ζαυξή χρόνου. Το συγκεντρώνεται στην παραγωγή περιορισμού;

- 1) Καθε διεργασία έχει ένα υπόγιφο στήριγμα που προσδέζεται σε έγαγει ένα στοιχείο από ή εναντίο αποκρυψών.
- 2) Ο μαζανάτωσης δεν πρέπει να είναι ένα υπόγιφο στήριγμα που προσδέζεται σε έγαγει στοιχείο για την παραγωγή και πάνω στον μαζανάτωση.
- 3) Κανένας από τους παραγωγούς δεν πρέπει να επανεισιγγίθει (loop) ένα υπόγιφο στήριγμα που προσδέζεται σε έγαγει στοιχείο για την παραγωγή.
- 4) Η μαζανάτωση πρέπει να επεξεργάζεται σε αύξηψη και ζαυξή χρόνου.

Xenodochonizinge semaphores για να ενισχύεται ασφαλίστηκε αποτελεστό και εγκρινιστό την διεργασία

ΑΙΓΑΝ



Defa 3

- Υποθέτετε ότι η αρίθμηση των γελίδων είναι 4 και η αρίθμηση των πρώτων είναι 3. Αναγνωρίζεται ότι η γελίδα με την αρίθμηση 3 είναι η πιο πρόσφατη. Διάφορες στρατηγικές για την αντιμετώπιση της απόβασης είναι:
- a) FIFO και LRU
 - b) Η γελίδα με την αρίθμηση 3 είναι η πιο πρόσφατη. Διάφορες στρατηγικές για την αντιμετώπιση της απόβασης είναι:

Λύση

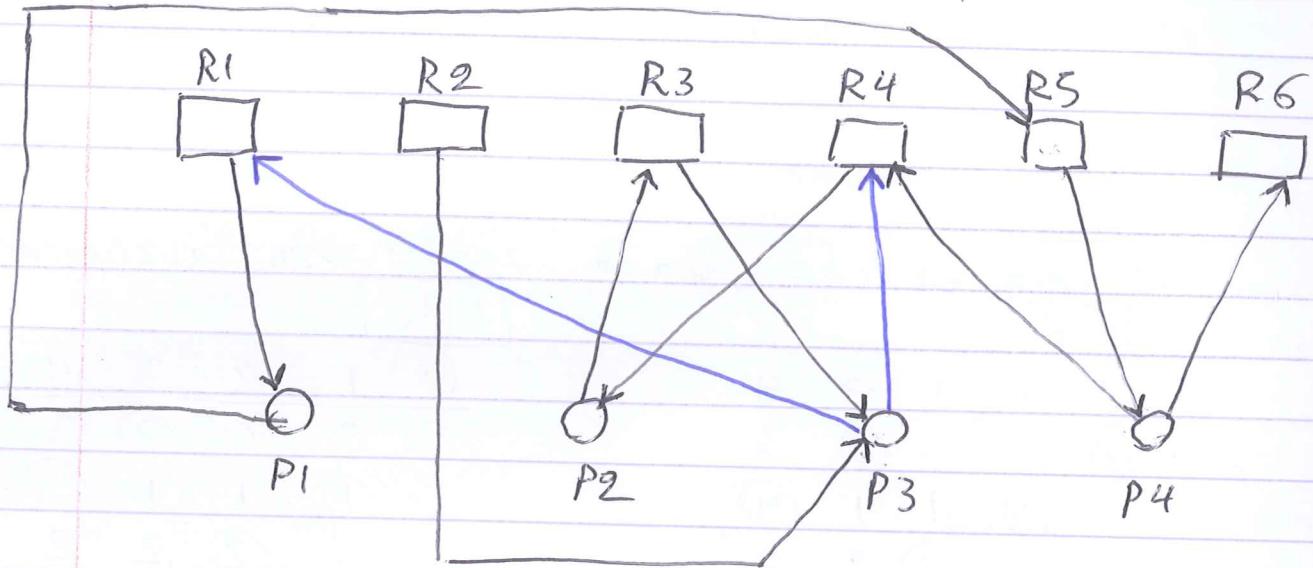
a) ίδια γελίδα μετά την πρώτη αντιμετώπιση				
FIFO		LRU		
0	1	2	3	4
① 0	0	③ 3		
① 1	1	④		
② 2	2			

b) Διαφορετικές γελίδες μετά την πρώτη αντιμετώπιση				
FIFO		LRU		
0	1	2	0	3
① 0	0	0	③ 3	
① 1	1	1	④	
② 2	2	2	2	2

Είτα 4

Διδεζαι η παραπάνω γράφητα κακαστή πόρων

- Λιγες είναι παραδειγματα αιτησης που θα προκαλέσει αδιέξοδο και για τις διεργασιες
- Λιγες ειςις είναι παραδειγματα αιτησης που θα προκαλέσει αδιέξοδο σε δυο μόνο διεργασιες.



Φ

- $P_3 \rightarrow R_1$
- $P_3 \rightarrow R_4$

Παραπάνω είναι αδιέξοδο, ^{εις η διεργασιες} η αιτηση που θα προκαλέσει αδιέξοδο είναι η $P_3 \rightarrow R_1$. Η αιτηση $P_3 \rightarrow R_4$ δεν έχει ισχύ σε όλη την γράφητα κακαστή πόρων, αλλα μόνο στη διεργασια P_3 .

Παραπάνω είναι αδιέξοδο για 2 διεργασιες η αιτηση που θα προκαλέσει αδιέξοδο είναι η $P_3 \rightarrow R_1$. Η αιτηση $P_3 \rightarrow R_4$ δεν έχει ισχύ σε όλη την γράφητα κακαστή πόρων.

Δέρα 5

ΔΙΒΕΖΑΙ ο παρακάτω πίνακας για την παραγωγή (segment table)

Segment	Base	Miros
0	219	609
1	2300	14
2	90	100
3	1327	580
4	1952	96

Όλοι οι αριθμοί είναι σε δεκαδικό. Νοις είναι φυσικές διεύθυνσης για τις παραγωγές διεύθυνσης:

a. 0, 350

b. 1, 10

c. 1, 14

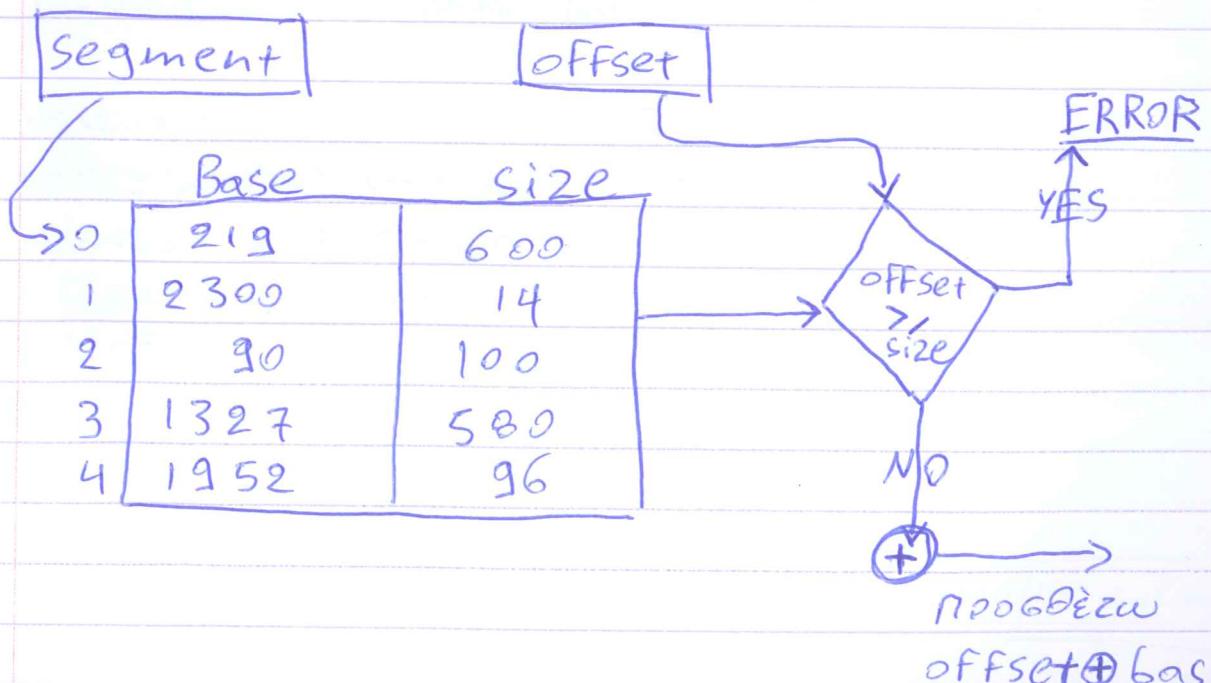


d. 2, 480

e. 3, 190

f. 4, 112

Λύση



Ελέγχω τα offset > size:

a) $350 > 600 \rightarrow \text{NO} \rightarrow 350 + 219 = \boxed{569}$

b) $10 > 14 \rightarrow \text{NO} \rightarrow 10 + 2300 = \boxed{2310}$

c) $14 > 14 \rightarrow \text{YES} \rightarrow \boxed{\text{ERROR}}$

d) $480 > 100 \rightarrow \text{YES} \rightarrow \boxed{\text{ERROR}}$

e) $190 > 580 \rightarrow \text{NO} \rightarrow 190 + 1327 = \boxed{1517}$

f) $112 > 96 \rightarrow \text{YES} \rightarrow \boxed{\text{ERROR}}$

a) Για την ιδεατή διεύθυνση 0,350 η φυσική διεύθυνση είναι 569.

b) Για την ιδεατή διεύθυνση 1,10 η φυσική διεύθυνση είναι 2310.

c) Για την ιδεατή διεύθυνση 1,14 η διεύθυνση είναι η έγκρη, καθώς υπερβαίνει το size του segment

d) Ιδεατή $\rightarrow 2,480 \rightarrow$ η έγκρη

e) Ιδεατή $\rightarrow 3,190 \rightarrow$ φυσική: 1517

f) Ιδεατή $\rightarrow 4,112 \rightarrow$ η έγκρη

Είρα 6

Στον αδημό διέκου φαίνουν μετρήσεις για
τους μιλινόρους 18, 11, 6, 22, 3, 34, 28 και 42
τε αυτήν την σειρά. Μια αναγνώση χρειάζεται
6 msec για να δε τελειώνει την γενούν
μιλινόρο. Νόμος ευοδικούς χρόνου αναγνώσης
ανατέται τε τους παραπάνω αδημίδιους;

- a. Πρώτη εισερχόμενη - Πρώτη εξυπηρετούμενη
- b. Εγγυητέρος μιλινόρος πρώτος
- c. Αδημίδιος του ανελκυστήρα (αρχική μετάδοση AN2)

Σε όλες τις περιπτώσεις ο βραχιόνας βρίσκεται
αρχικά στον μιλινόρο 20.

Λύση

a) Αρχικώς μιλινόρο 20.

18, 11, 6, 22, 3, 34, 28, 42

$$20 - 18 = 2 \text{ cylinders}$$

$$18 - 11 = 7 \text{ cylinders}$$

$$11 - 6 = 5 \text{ cylinders}$$

$$22 - 6 = 16 \text{ cylinders}$$

$$22 - 3 = 19 \text{ cylinders}$$

$$34 - 3 = 31 \text{ cylinders}$$

$$34 - 28 = 6 \text{ cylinders}$$

$$42 - 28 = 14 \text{ cylinders}$$

$$\text{Σύνολο} = 2 + 7 + 5 + 16 + 19 + 31 + 6 + 14 = 100 \text{ cylinders}$$

$$\text{Ευοδικός χρόνος αναγνώσης} = 100 * 6 \text{ msec} = 600 \text{ msec}$$

20

Πάνω στον αυτισμό τους

20, 22, 18, 11, 6, 3, 28, 34, 42
2 4 7 5 3 25 6 8

b)

$$18, 11, 6, 22, 3, 34, 28, 42$$

• Αρχισε από τη μέση της 20.

$$22 - 20 = 2$$

$$22 - 18 = 4$$

$$18 - 11 = 7$$

$$11 - 6 = 5$$

$$6 - 3 = 3$$

$$28 - 3 = 25$$

$$34 - 28 = 6$$

$$42 - 34 = 8$$

$$\Sigma \nuόλο (cylinders) = 60$$

$$\text{Ευρισκός χρόνος αναγνώρισης} = 60 * 6 = \boxed{360 \text{ msec}}$$

20

$$8) 18, 11, 6, 22, 3, 34, 28, 42$$

Ο αριθμός του ανελκυστήρα για αυτήν την περίοδο διαδέχεται αναδιικά από το 20 μέχι τών 215 κιλών (τελευταία δείπνα) + έχει ραγίζει τον αναλογικό όρο της αναγνώρισης από την προηγούμενη διαδέξιμη αναδιικά (τελευταία δείπνα) έως την εγκατάσταση των κιλών.

• Αρχισε από τη μέση της 20.

$$22 - 20 = 2$$

$$28 - 22 = 6$$

$$34 - 28 = 6$$

$$42 - 34 = 8$$

$$42 - 18 = 24$$

$$18 - 11 = 7$$

$$11 - 6 = 5$$

$$6 - 3 = 3$$

$$\Sigma \nuόλο (cylinders) = 61$$

$$\text{Ευρισκός χρόνος αναγνώρισης} =$$

$$61 \text{ cylinders} * 6 \text{ msec} =$$

$$\boxed{366 \text{ msec}}$$

Επεζεύγματα 2021

Περα 1

Θεωρήστε ότι η παρακάτω λογισμική έχει ενσημειώσει την απόβλητη του αποτελείας αναδειγμάτων για 2 διεργασίες:

Type

TurnType = (one, two, Either);

Var

Turn: TurnType; { Κοινή γενική μεταβλητή για τις 2 διεργασίες }

Processi { Τοι ο ειναι 1 ή 2 }

:

Var

Ours, Theirs : TurnType; { Τοινη σεν προσεσσι. Σεν
διεργασια 1, ours=one και theirs=two
Σεν διεργασια 2 ours=two και
theirs=one }

Begin { Processi }

:

Begin { Αρχι αναδειγματος }

if Turn=Theirs Then

wait

else

Turn=Ours;

End; { Αρχις αναδειγματος }

upi githn nseproxh

:

Begin { Τελος αναδειγματος }

if other process is suspended Then

Begin

Turn=Theirs

resume other process

End

Else

Turn=Either;

End; { Τελος αναδειγματος }

H ιδέα είναι να επιχειφώντες είσοδο δύο υπότιτρο τύπων οζαν είναι η "εισιτήριας" (our turn) ή "αναμενόμενης εισιτήριας" (either turn). H γιρή "Either" μαζαπρετι και ανατίθεται να επαλλάσσονται οι διεργασίες που θα παίνουν δύο υπότιτρο σημεία τους. Αρχικά, turn=Either. Λοιπόν είναι ΝΑΙ γιατί; Αν ΟΧΙ γιατί;

Λύση

Αρχική μαζαπραση:

- Turn = Either (και οι 2 διεργασίες προστίνα και εισιτήριας δύο υπότιτρο σημεία)
- Process 1 → Ours = One, Theirs = Two
- Process 2 → Ours = Two, Theirs = One

- a. H (process 1) είδεξει Turn = Either, η γενικότερης έξι οζι προπει να προχωρήσει δύο υπότιτρο σημεία
- b. H (process 2) ενησις είδεξει την γιρή του Turn οζι είναι Either, και η γενικότερη και ουδέτερη προπει να προχωρήσει δύο υπότιτρο σημεία.

- c. Και οι 2 διεργασίες θα εισέλθουν δύο υπότιτρο σημεία σχεδόν ταυτόχρονα, διότι η απόφαση τους βασίστηκε στην γιρή Turn.

Ευκίνηση

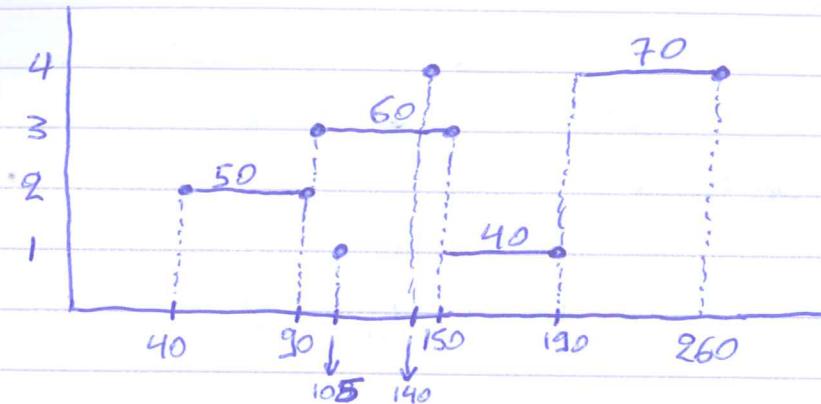
Το σφάλμα να είναι, οζι η ταυτόχρονη προσβαση συν διεργασιών οζο υπότιτρο σημεία παραβιάζει την αρχή του αρχιβαΐου ανοικεστήρα, το υπότιτρο σημεία πρέπει να είναι προβασικό ανα τια διεργασία για να αποφύγει δυοχροίσεις και race conditions.

* * AÜ6n · Ausznp's Evadagris:

```
while (true) {  
    while ((turn) <> 0) { wait }  
    critical_section  
    turn = 1  
    non_critical_section  
}
```

```
while (true) {  
    while (Turn <> 1) { wait }  
    critical_section  
    turn = 0  
    non_critical_section  
}
```

a) FCFS



$$TT_1 = 190 - 105 = 85$$

$$TT_3 = 150 - 90 = 60$$

$$TT_2 = 90 - 40 = 50$$

$$TT_4 = 260 - 140 = 120$$

$$ATT = 85 + 50 + 60 + 120 = 295 / 4 = 73.75$$

$$WTT_1 = 85 / 40 = 2.125$$

$$WTT_3 = 60 / 60 = 1$$

$$WTT_2 = 50 / 50 = 1$$

$$WTT_4 = 120 / 70 = 1.71$$

$$AWTT = (2.125 + 1 + 1 + 1.71) / 4 = 1.459$$

$$WT_1 = 150 - 105 = 45$$

$$WT_3 = 90 - 90 = 0$$

$$WT_2 = 40 - 40 = 0$$

$$WT_4 = 190 - 140 = 50$$

$$AWT = (45 + 0 + 0 + 50) / 4 = 23.75$$

$$WWT_1 = 45 / 40 = 1.125$$

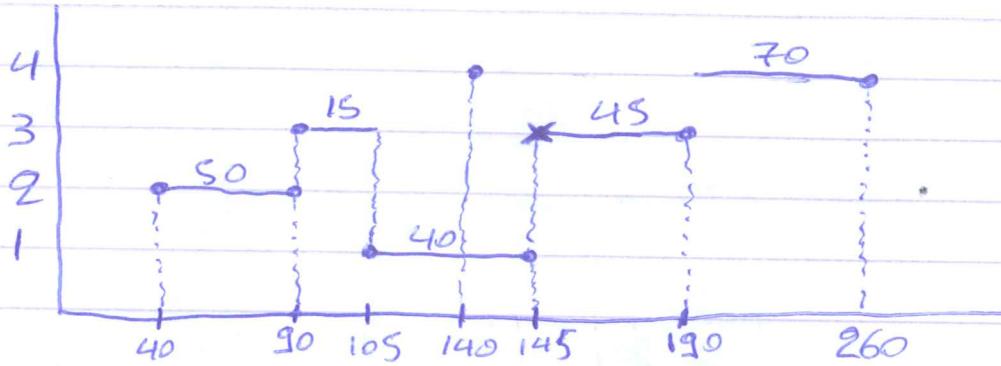
$$WWT_3 = 0 / 60 = 0$$

$$WWT_2 = 0 / 50 = 0$$

$$WWT_4 = 50 / 70 = 0.714$$

$$AWWT = (1.125 + 0 + 0 + 0.714) / 4 = 0.459$$

b) SRTN



$$TT_1 = \cancel{140 - 40} = 40$$

$$TT_2 = 90 - 40 = 50$$

$$TT_3 = 190 - 90 = 100$$

$$TT_4 = 260 - 140 = 120$$

$$ATT = \frac{40 + 50 + 100 + 120}{4}$$

$$ATT = 77,5$$

$$WTT_1 = 40 / 40 = 1$$

$$WTT_2 = 50 / 50 = 1$$

$$WTT_3 = 100 / 60 = 1,66$$

$$WTT_4 = 120 / 70 = 1,71$$

$$AWTT = \frac{1 + 1 + 1,66 + 1,71}{4}$$

$$AWTT = 1,34$$

$$WT_1 = \cancel{40 - 40} = 0$$

$$WT_2 = \cancel{50 - 50} = 0$$

$$WT_3 = 90 - 90 = 100 - 60 = 40$$

$$WT_4 = 120 - \cancel{70} = 50$$

$$AWT = \frac{0 + 0 + 40 + 50}{4} = 22,5$$

$$WWT_1 = 0 / 40 = 0$$

$$WWT_2 = 0 / 50 = 0$$

$$WWT_3 = 40 / 60 = 0,66$$

$$WWT_4 = 50 / 70 = 0,71$$

$$AWWT = \frac{0 + 0 + 0,66 + 0,71}{4}$$

$$= 0,34$$

Ωδη 3

Σενού αδηρό δίβου φλάνου αιτησεις για zoos κειμενού
 7, 20, 10, 36, 5, 22, 42, ή ε αυτή στην δείρα.
 Η γεγανικότητα της μεφοράς σε γειωνιαδ
 υλικό χρειάζεται 4msec. Ήπος γυναικών
 χρόνος αναγνώσης ανατέται περίπου πάνω από
 αλγόριθμους.

- a) Πρώην Εισερχόμενη - Πρώην Εξυπέρερχη (FCFS)
- b) Εγγύζερος υλικός Πρώτος
- c) Αλγόριθμος του ανελκυστήρα (αρχική μεταδόνα AN)

Εε ίδες της περιπτώσεις, ο βραχιόνας βρίσκεται
 αρχικά σενού υλικό 20.

a) Αιτησεις: 20 7 20 10 36 5 22 42
 Βιταζα: 13 13 10 26 31 17 20

Εύρος Βιταζών: 130 cylinders

Ευροτικός Χρόνος Αναγνώσης: $130 * 4 = \boxed{520 \text{ msec}}$

b) Αιτησεις: 20 22 20 10 7 5 36 42
 Βιταζα: 2 2 10 3 2 31 6

Εύρος Βιταζών: 56 cylinders

Ευροτικός Χρόνος Αναγνώσης: $56 * 4 = \boxed{224 \text{ msec}}$

c) ~~Αρχιτεκτονική~~

Αιτησεις: 20 22 36 42 20 10 7 5
 Βιταζα: 2 14 6 22 10 3 2

Εύρος Βιταζών: 59 cylinders

Ευροτικός Χρόνος Αναγνώσης: $59 * 4 = \boxed{236 \text{ msec}}$

Θέμα 4

Ένα δειγματικό πρόβλημα χρησιμοποιεί ηδεσή virtual και' αναίνετη (virtual memory και demand-based paging). Αναδιδικά, τια σελίδα δεν έρχεται στην virtual αν δεν γίνεται. Έτσι, αρχικά δεν φορώνεται καθίστα σελίδα στην virtual και έχουμε εφάπτα σελίδας αυτά και τελικά πρώην διεύθυνση που γίνεται. Το γύρω μέσο χρησιμοποιεί τον αδιόριθτο FIFO για την αντικατάσταση των σελίδων. Δινέται η παρακάτω σειρά αισθησών σελίδων (αριθμοί σελίδων που γίνονται).

1, 2, 3, 4, 5, 3, 5, 3, 0, 2, 4, 1, 5, 4, 5, 5, 2, 3

Πόσα ~~page~~ page faults έχουμε τελικά στην σειρά αυτή (τελευταία και τα αρχικά) αν:

- Δινορται στην διεργασία γρία πλαστικής
- Δινορται στην διεργασία γεννητρια πλαστικής
- Δινορται στην διεργασία γρία πλαστικής και χρησιμοποιείται ο αδιόριθμος LRU.

Λύση

a)

1	2	3	4	5	3	5	3	0	2	4	1	5	4	5	5	2	3
①	1	1	④	4	4	4	4	4	②	2	2	⑤	5	5	5	5	5
②	2	2	⑤	5	5	5	5	5	④	4	4	4	4	4	4	②	2
③	3	3	3	3	3	3	3	⑥	0	0	①	1	1	1	1	1	③
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	

Έχουμε 12 εφάπταση και 6 hits.

6)

1	2	3	4	5	3	5	3	0	2	4	1	5	4	5	5	2	3	
①	1	1	1	1	⑤	5	5	5	5	5	5	5	5	5	5	4	4	4
②	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	5	5	5
③	3	3	3	3	3	3	3	3	②	2	2	2	2	2	2	2	2	2
④	4	4	4	4	4	4	4	4	4	4	①	1	1	1	1	1	1	1
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	

Έχουμε 11 σφάλματα και 7 hit.

7)

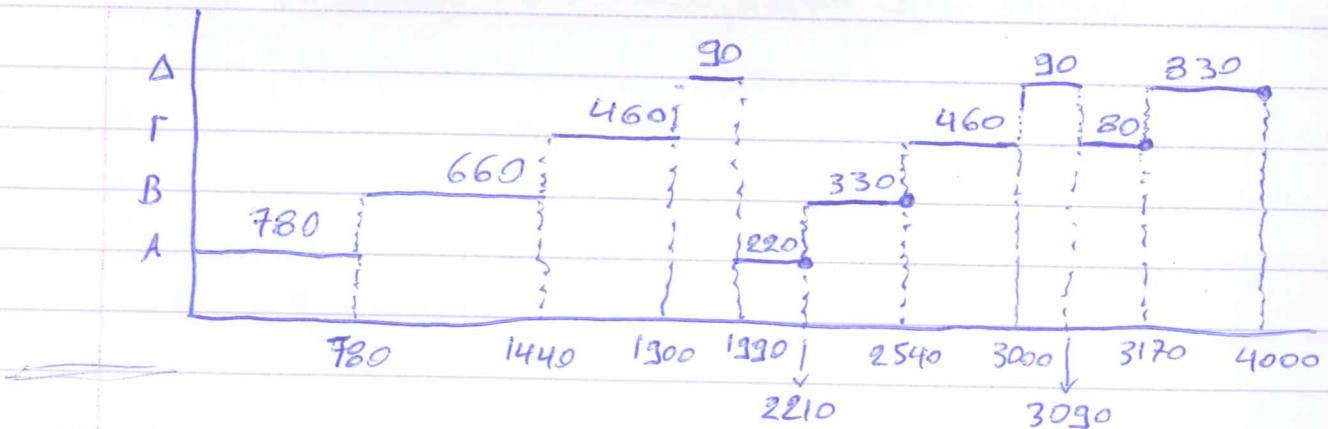
1	2	3	4	5	3	5	3	0	2	4	1	5	4	5	5	2	3
①	1	1	④	4	4	4	4	⑥	0	0	①	1	1	1	1	②	2
②	2	2	⑤	5	⑤	5	5	②	2	2	⑤	5	⑤	5	5	⑧	5
③	3	3	③	3	③	3	3	④	4	4	④	4	4	4	4	③	3
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	

Έχουμε 12 σφάλματα και 6 hit.

Άσκηση 5

Δινέται ένα σύστημα Linux με 4 διεργασίες A, B, Γ, Δ με αριθμούς προτεραιότηταν 101, 107, 117, 122. Οι διεργασίες κατευθύνονται σε χρόνο t=0 με αδιαβίωτη διάρκεια. Να δώσετε τον χρόνο επέκτασης καθε διεργασίας αν καθετία ανά αυτές έχει χρόνο εκτέλεσης (RUN TIME) 1000 ms. Ο χρόνος υπολογίστε στην πλήν των αποδουθειών στην πολιτική του O(1) χρονοδρομολογητή και οι διεργασίες είναι CPU bound (δεν κάνουν I/O).

$$\begin{aligned}101: A &= (140 - 101) * 20 = 39 * 20 = 780 \text{ ms} \\107: B &= (140 - 107) * 20 = 33 * 20 = 660 \text{ ms} \\117: \Gamma &= (140 - 117) * 20 = 23 * 20 = 460 \text{ ms} \\122: \Delta &= (140 - 122) * 5 = 18 * 5 = 90 \text{ ms}\end{aligned}$$



Άρα οι χρόνοι σερφασισμού είναι:

A → 2210

B → 2540

Γ → 3170

Δ → 4000

Ωρα 6

Διεργαί η παραίστων μαζίστων ενώς γυρνήσας

Δοκίμων πόροι		Μέχιστο αιχμήσων		Διαθέσιμων πόροι								
A	B	C	D	A	B	C	D	A	B	C	D	
p ₀	0	0	1	1	0	1	1	2	4	5	0	1
p ₁	1	0	1	2	1	5	2	3				
p ₂	1	5	3	0	2	5	3	7				
p ₃	0	2	7	2	0	3	9	2				
p ₄	3	0	1	4	9	3	1	6				

Anavzingze gis παραίστων ερωτήσεις χρησιμοποιώντας ταν αλγόριθμο αποφυγής αδιέξοδου:

a. Ποια ειναι τα πειράξτερα ταν nivaua "Anofένουες αιχμήσων"

b. Ειναι τα γυρνήσα γε αδιέξοδο;

c. Αν έτσι ειναι ταν διαδικασία p₁ τα αιχμήσων (0,5,0,1)

κανει τα εγγυηστεί σύγχυρα ταν αλγόριθμο;

~~ΕΠΙΛΟΓΕΣ~~

Λύση

a)	Anofένουες				= Μέχιστες - Δοκίμες
	A	B	C	D	
	0	1	0	1	
	0	5	1	1	
	1	0	0	7	
	0	1	2	0	
	6	3	0	2	

⇒

6. Θέμα Anoτένουσες & Διαθέσιτες

Δοστένοι				
	A	B	C	D
p0	0	0	1	1
p1	1	0	1	2
p2	1	5	3	0
p3	0	2	7	2
p4	3	0	1	4

Anoτένουσες				
	A	B	C	D
	0	1	0	1
	0	5	1	1
	1	0	0	7
	0	1	2	0
	6	3	0	2

Διαθέσιτοι				
	A	B	C	D
	4	5	0	1
	+	+	1	1

(ΜΕ ζΩ ΠΟΥ ΚΑΡΙΑΡΗ ΜΑ ΔΙΕΡΓΑΣΙΑ ΒΙΤΦΩΡΑ ήΕ)
ΑΝΟΤΕΝΟΥΣΕΣ \leq ΔΙΑΘΕΣΙΤΕΣ, ΣΠΙΓΡΕΦΗΝ 2003
ΠΟΡΟΥΣ ΖΗΣ ΔΙΑΘΕΣΙΤΕΣ.

Δοστένοι				
	A	B	C	D
p0	0	0	0	0
p1	1	0	1	2
p2	1	5	3	0
p3	0	2	7	2
p4	3	0	1	4

Anoτένουσες				
	i	s	r	s
	1	5	1	2
	1	0	0	7
	0	1	2	0
	6	3	0	2

Διαθέσιτοι				
	A	B	C	D
	4	5	1	2
	+	+	1	2

Δοστένοι				
	A	B	C	D
p0	0	0	0	0
p1	0	0	0	0
p2	1	5	3	0
p3	0	2	7	2
p4	3	0	1	4

* Αναγνωρίζεται στην πόλη της Εύβοιας στην οποία υπάρχουν διαθέσιτες καριάριζες και διεργασίες.

Δοστένοι				
	A	B	C	D
p0	0	0	0	0
p1	0	0	0	0
p2	1	5	3	0
p3	0	0	0	0
p4	3	0	1	4

Διαθέσιτοι				
	A	B	C	D
	5	7	9	6
	+	+	1	2

C.

Aiznēn

$$\rightarrow \text{Pi}(0, 5, 0, 1)$$

1) Νāw εzn πi διεργασία
 2) Αφαιρίω za voitēra zns aiznēn
 anō nivaka ~~Anoferousges~~ Anoferousges
 kai nivaka ~~Δiaθēsikoi~~ Δiaθēsikoi. kai proctēzē
 ta voitēra zns aiznēn εzo nivaka

Δoσtēvoi

Δoσtēvoi				
	A	B	C	D
p0	0	0	1	1
p1	1	5	1	3
p2	1	5	3	0
p3	0	2	7	2
p4	3	0	1	4

Anoferousges				
	A	B	C	D
	0	1	0	1
	0	0	1	0
	1	0	0	7
	0	1	2	0
	6	3	0	2

Δiaθēsikoi				
	A	B	C	D
	0	0	0	0
	4	0	0	0
	0	0	0	0

H Pi εxei zo struajuta va jniziesi avzous
 zous nōpous. Di nōpol eival diaθēsikoi, àpa
 da progrōinθoukε óz* divoufε zous nōpous.
Euknèpastas:

κatia δiεrgasia δev kapnipesai, àpa δev
 divoufε nōpous kai n aiznēn δev jnoperi
 va eguniprezenθei.

Ioúvios 2023

Díka 1

Ωι αναλογίες διεργασιών επχορτάνται συνεχόμενα
και ευρίσκεται και απέως ευναγωνιστούνται
και στη CPU.

Aριθμός Διεργασίας	Χρόνος Εκτέλεσης
1	70
2	40
3	30
4	60

Να εγγένετε όλους τους εγγίσιους αλγόριθμους χρονοδρομού:

a) SRTN

b) FIFO, στην περίπτωση συνεχόμενων αφίσεων
η χρονοδρομίαν γίνεται ότι αριθμητική σειρά.

c) Round Robin, ότι $\text{ιτάντο} = 10$. (Αρχισε στην
δρομολόγηση ότι αριθμητική σειρά)

d) Round Robin ότι $\text{ιτάντο} = 20$. (>>>)

-Για κάθε περίπτωση υπολογίστε:

1) To χρόνο επιεγραφής κάθε διεργασίας. (TT)

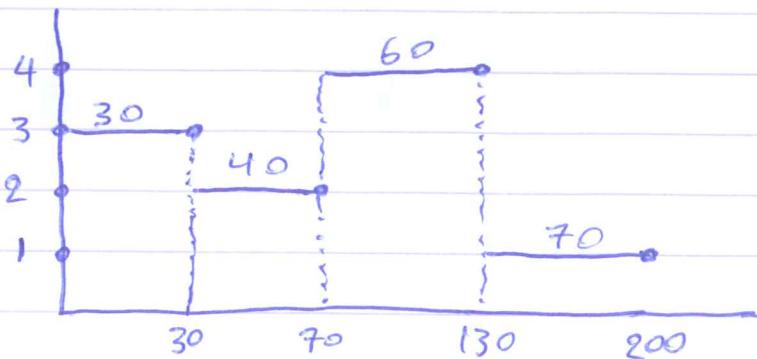
2) To μέσο χρόνο επιεγραφής. (ATT)

3) To χρόνο αναπονίας. (WT)

4) To μέσο χρόνο αναπονίας. (AWT)

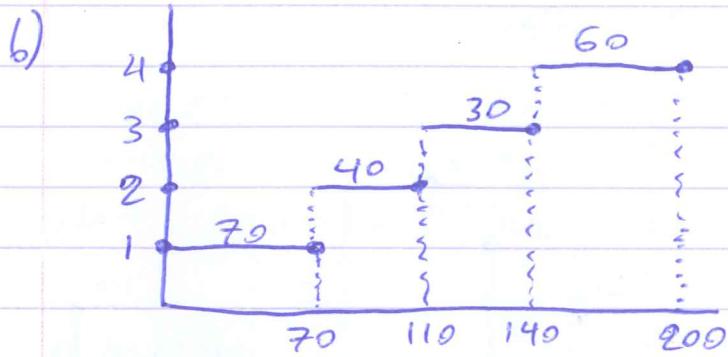
Άσκηση

a)



$$\begin{array}{l}
 TT_1 = 200 - 0 = 200 \\
 TT_2 = 70 - 0 = 70 \\
 TT_3 = 30 - 0 = 30 \\
 TT_4 = 130 - 0 = 130
 \end{array}
 \quad \left| \begin{array}{l}
 ATT = \frac{200 + 70 + 30 + 130}{4} \\
 ATT = 107,5
 \end{array} \right.$$

$$\begin{array}{l}
 WT_1 = 200 - 70 = 130 \\
 WT_2 = 70 - 40 = 30 \\
 WT_3 = 30 - 30 = 0 \\
 WT_4 = 130 - 60 = 70
 \end{array}
 \quad \left| \begin{array}{l}
 AWT = \frac{130 + 30 + 0 + 70}{4} = 57,5
 \end{array} \right.$$

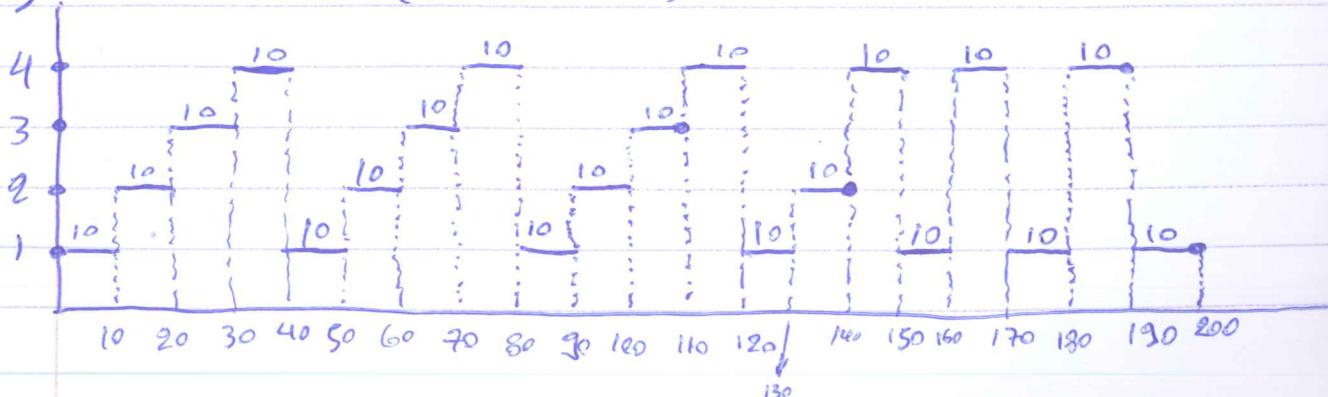


1 → 180 - 70 = 60
2 →
3 →
4 →

$$\begin{array}{l}
 TT_1 = 70 - 0 = 70 \\
 TT_2 = 110 - 0 = 110 \\
 TT_3 = 140 - 0 = 140 \\
 TT_4 = 200 - 0 = 200
 \end{array}
 \quad \left| \begin{array}{l}
 ATT = \frac{70 + 110 + 140 + 200}{4} = 130
 \end{array} \right.$$

$$\begin{array}{l}
 WT_1 = 0 - 0 = 0 \\
 WT_2 = 70 - 0 = 70 \\
 WT_3 = 110 - 0 = 110 \\
 WT_4 = 140 - 0 = 140
 \end{array}
 \quad \left| \begin{array}{l}
 AWT = \frac{0 + 70 + 110 + 140}{4} = 80
 \end{array} \right.$$

c) Round Robin (10 ms à 20 ms)



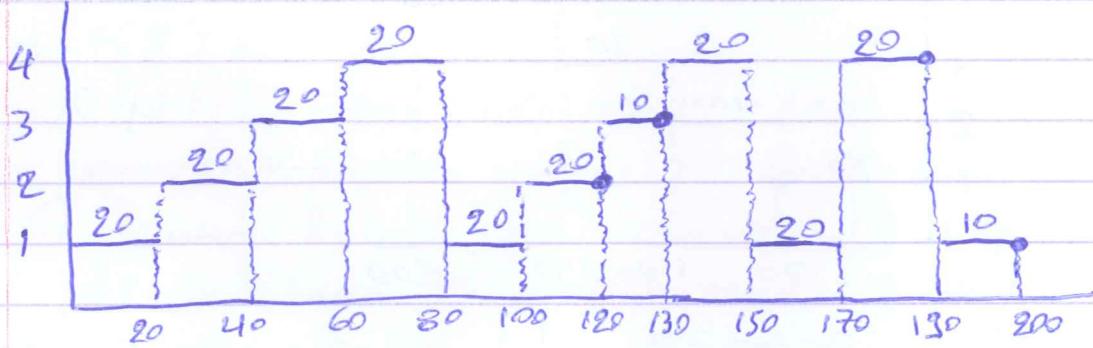
$$\begin{aligned}
 TT_1 &= 200 - 0 = 200 \\
 TT_2 &= 140 - 0 = 140 \\
 TT_3 &= 110 - 0 = 110 \\
 TT_4 &= 190 - 0 = 190
 \end{aligned}$$

$$ATT = \frac{200+140+110+190}{4} = 160$$

$$\begin{aligned}
 WT_1 &= 200 - 70 = 130 \\
 WT_2 &= 140 - 40 = 100 \\
 WT_3 &= 110 - 30 = 80 \\
 WT_4 &= 190 - 60 = 130
 \end{aligned}$$

$$ATT = \frac{130+100+80+130}{4} = 110$$

d) Round Robin (20 ms avto)



$$\begin{aligned}
 TT_1 &= 200 - 0 = 200 \\
 TT_2 &= 120 - 0 = 120 \\
 TT_3 &= 130 - 0 = 130 \\
 TT_4 &= 190 - 0 = 190
 \end{aligned}$$

$$ATT = \frac{200+120+130+190}{4} = 160$$

$$\begin{aligned}
 WT_1 &= 200 - 70 = 130 \\
 WT_2 &= 120 - 40 = 80 \\
 WT_3 &= 130 - 30 = 100 \\
 WT_4 &= 190 - 60 = 130
 \end{aligned}$$

$$AWT = \frac{130+80+100+130}{4} = 110$$

Άστα 2

Ένα σύγχρονα χρονοτροποία για functions down() και up() τόσο για binary semaphores. Οι γενταζόφοροι διαδικασίες για εγγύησης προσών ράπουρ να πάρουν τόσο zis zifēs ο και l. Γράψε για functions downlight() και upgrd() που παρέχουν έτσι χρήση γενικών semaphores, διαδικασίες γενταζόφορους που προσών ράπουρ οντισμού που θα αποτελέσει zifēs. Τα functions που θα γράψες θα χρονοτροπούν για up και down για εγγύησης.

Semaphore mutex=1;
Semaphore binary=1;
int count=0;

```
downgn(){  
    down(binary);  
    count++;  
    up (binary);  
    while(mutex<1){wait}  
    down(mutex);  
}
```

```
upgrd(){  
    down(binary);  
    up (mutex);  
    count--;  
    up (binary);  
    if(count==0)  
        while(mutex<0){  
            up (mutex);  
        }  
}
```

Θέμα 3

"Ενας υπολογιστής έχει 11 ιδίους πόρους και Ν διεργασίες διευδίνουν σε χρήση zous. Κάθε διεργασία λαμβάνει να χρειασθεί τα πολύ 3 πόρους. Για πότες zifēs zou Ν δεν υπάρχει περιζων εκφάντων αδιέξοδου για σύγχρονη

Λύση

Για να την υπάρξει αδιέξοδο θα πρέπει να ισχύει η ευγενεικότερη ανισότητα που είναι το worse case:

$$3 * N \leq n_{\text{pwz}} \text{ (που είναι } 11)$$

$$3N \leq 11$$

Αν λύσω την ανισότητα προκύπτει:

$$N \leq 3.666\ldots$$

(δεν λαμβάνει να έχει αριθμό διεργασιών για την αριθμό των διεργασιών και παραχωρεί ίσια για ~~N~~ $N \leq 3$ δεν υπάρχει αδιέξοδος)

Άρα για zifēs $N > 3$ θα υπάρχει περιζων εκφάντων αδιέξοδου

Θέτα 4

Πολλές φορές ο αλγόριθμος του ανελκυστήρα για τη δροφολόγηση του Braxiova δίσμου αναφέρεται στη βιβλιογραφία ως αλγόριθμος δάρωσης. Αυτό γίνεται μάζι μπάρχει ειδοποιητική διαφορά ανάφεσης στη δροφολόγηση του Braxiova και του πραγματικού ανελκυστήρα. Μακι είναι η διαφορά αυτή; Για να απαντήσετε πως είναι η διαφορά αυτή; Για να αναλυτικότερα μπορούμε σε πραγματικό ανελκυστήρα να χρησιμοποιήσουμε τον αλγόριθμο εξεργαζόμενος θρόφος Πρώτος.

Λύση

Η αύρια διαφορά λεγαρεύει Braxiova δίσμου και του πραγματικού ανελκυστήρα, σχεζίζεται τε τη φύση των αιτημάτων που εγκατατίθενται και την δυνατική των εγκατατίθεντων τους.

- Για του Braxiova δίσμου, ο αλγόριθμος δάρωσης είναι αποστελεχτικούς επειδή ο Braxiovaς κινείται σε γραφική πορεία απογειωνάς από το ίσα αύριο 620 άριτο του δίσμου (και εγκατατίθενται στην αιτημάτων του δίσμου)
- Στην περίπτωση του πραγματικού ανελκυστήρα η μαζαγάνων είναι πιο περιορισμένη δόχεια της δυνατικής του ανθρώπινης κίνησης και των αλλαγών στην αιτημάτων.
- Ο αλγόριθμος ~~εξεργαζόμενος θρόφος Πρώτος~~ ληφθεί να είναι αποστελεχτικούς σε ορισμένες μαζαγάνεις στα τα πειρήσει του κρόνο ανατονής για τους χρήστες που είναι πιο κοντά στον ανελκυστήρα.

Θέμα 6

Ανεβοι η παραπάνω δείρα αντίσεων για σελίδες
και ιδεαζήσεις μνήμης:

5, 3, 18, 5, 7, 3, 5, 7, 13, 14, 15, 3, 5, 14, 15, 12, 12, 19, 20

Ωστριθείτε ότι η φυσική τιμή χωρίς 8 σελίδες και
ότι ζε TLB θα έρει να αλογηνεύει 4 σελίδες.
Οι παραπάνω πίνακες σειχνούν στην αντιστοιχία
ιδεαζώντων διευδύνωσην και διευθύνσεων φυσικής
μνήμης.

Σελίδα Ιδεαζήσεις Μνήμης	Αντιστοιχό ^η Πλαισίο Φυσικής Μνήμης	Σελίδα Ιδεαζήσεις Μνήμης	Αντιστοιχό ^η Πλαισίο Φυσικής Μνήμης	Σελίδα Ιδεαζήσεις Μνήμης	Αντιστοιχό ^η Πλαισίο Φυσικής Μνήμης
0	2	7	6	14	X
1	X	8	3	15	X
2	0	9	X	16	X
3	X	10	X	17	X
4	X	11	4	18	X
5	1	12	5	19	X
6	X	13	X	20	X

Πριν γεμίνησουν οι παραπάνω αντίσεις, η TLB
έχει αποθηκεύσει 8 σελίδες που διαφέρουν από
παραπάνω πίνακα. Το TLB και η φυσική τιμή
χρησιμοποιούν πολιτική FIFO για την αντικατάσταση
των σελίδων. Ωστριθείτε ότι οι σελίδες έχουν την
τελευταία προσέλευση σε παραπάνω πίνακα, γεμίνονται
και διευθύνονται στην παραπάνω πίνακα μετά την
αντικατάσταση σελίδων και την TLB.

Σελίδα Ιδεαζήσεις Μνήμης	Πλαισίο Φυσικής Μνήμης
1	5
2	11
3	2
4	7

**

* πίνακας Page Fault

** πίνακας TLB

- a) Να δείξετε στην τελική παράσταση του TLB για στην δοθεκα σειρά αιτησεων.
- b) Γιατί είναι το hit ratio του TLB.
- c) Αν κάθε ανάγνωση μνήμης ανατεί 100 χρονικές πορείες, πόσος χρόνος ανατείχεται στη συν προσπέλαση στην παραπάνω δελιδιά, υποθέτοντας ότι γίνεται η προσπέλαση με κάθε αναφορά δελιδιά; (Η γεναφορά της δελιδιάς στην φυσική μνήμη δεν παραπομπή χρόνου αφού τότε ευσεβείται άλλη διεργασία).

Παρατητικό: Σε περίπτωση page fault, δεν ρίχνεται στη σελίδα αντιγράφεται από στη διευρεύοντα μνήμη σαν φυσική και εισάγεται σε αντίστοιχη εγγραφή στο TLB.

Λύση

(Για τη σφωντή)

- 1ο Βήμα: Ελέγχουτε αν η σελίδα που γινείται αναγνωρίζεται ως πλαίσιο, αν ναι τότε: (page fault) $PF = \checkmark$, αλλιώς $PF = \text{εστίαρχο}$ και αναγνωρίζεται ανάλογα με στην πολιτική.
- 2ο Βήμα: Ελέγχουτε αν το γεγος σελίδα-πλαίσιο στο TLB. Αν ναι $\rightarrow TLB = \checkmark$, αλλιώς $TLB = *$ (εφαλτά) και ανακαθιστάτε ανάλογα.

Παραπομπές: (για λύση)

- a) Αν $PF \rightarrow *$ τότε $TLB \rightarrow *$

b)

Σελίδα	Πλαίσιο
Ιδεανής	Φυσικής
Μνήμης	Μνήμης
0	-
1	6
2	-
3	1
4	0

Παίρνω στη σελίδα με σημείων π.χ. (5, 3, 0, 2)

Ξεκινώντας με 5 (σενοπάρχει $\rightarrow PF(*)$)

- Άρα το βήμα στο TLB. (και παίρνω το πρώτο πλαίσιο)
- Τέλος κάνω την πίνακα PF_{TLB} φυσικής μνήμης (το τελεόρρευτο (0))

TLB	
10	0
20	1
0	2
	3

a)	PF	5	3	13	5	7	3	5	7	13	14	15	3	5	14	15	12	12
	TLB	✓	*	✓	✓	✓	✓	✓	✓	*	*	*	✓	*	✓	✓	✓	✓
			*	*	*	✓	✓	✓	*	*	*	*	*	*	✓	*	*	

	↓	19	20
PF	*	*	
TLB	*	*	

0	2	11	4
1	X	12	8
2	Ø	13	X
3	X ⁰	14	X ²
4	X	15	X ³
5	X ⁴	16	X
6	X	17	X
7	Ø	18	7
8	Ø	19	X ⁵
9	X	20	X ⁶
10	X		

TLB

8 3 14 12 X Ø 25
 14 18 18 19 X Ø 25
 4 23 5 2 8 3 20 4 Ø 19 6
 2 8 3 5 Ø 19 6 4

Η μετασχηματισμένη TLB (επίλυση):

Σε Τίδα	Παλαιόριο Φυσικός Μνήμης
12	5
19	5
20	6
5	4

b) Ανοί για την πρώτη περιάδα της hit ratio της TLB είναι:

$$8 \text{ hit} / (8 \text{ hit} + 12 \text{ miss}) = 8 / 20 = 0,4 = 40\%$$

Apa Hit ratio: $8 / 19 = 0,421 = 42,1\%$

c) Εάν η ίδια TLB hit \Rightarrow 1 Ανάγκη

Εάν η ίδια TLB miss \Rightarrow 2 Ανάγκες (Επειδή θα διαβάσει)
 και \Rightarrow PMT

$$8 \text{ hit} * 1 = 8 \quad \left. \right\} \text{Apa} \quad 8 + 22 = 30 * 100 = 3.000$$

$$11 \text{ miss} * 2 = 22 \quad \left. \right\} \text{Χρονικός Μονάδες}$$

Ιούνιος 2018

Θίκα 2

Να γράψετε παραδημοκρατημένη επίλυση για την πρόβλημα των παραγωγού-καταναλωτή, διότι οποιοι υπάρχουν 2 καταναλωτές και 1 γράφημα εγγίζει κανόνες:

- 1) Ο παραγωγός πρέπει να γράφει σεν τυπί αφότου γίνει ανάγνωση και από τους δύο καταναλωτές ή η έγγιρηση στην πρώτη έγγραφη.
- 2) Κανένας καταναλωτής δεν πρέπει να διαβάσει ανά στην τυπί αυτόν οποιος δεν έχει γραφεί για πρώτη φορά.
- 3) Κανένας καταναλωτής δεν πρέπει να διαβάσει τα ίδια περιεχόμενα του buffer δύο φορές γενέκει.
- 4) Οι καταναλωτές πηρούν να διαβάζουν ταυτόχρονα

Process Producer	Process Consumer 1	Process consumer 2
:	:	:
repeat	repeat	repeat
:	:	:
add_item	consume_item	consume_item
:	:	:
until Forever	until Forever	until Forever

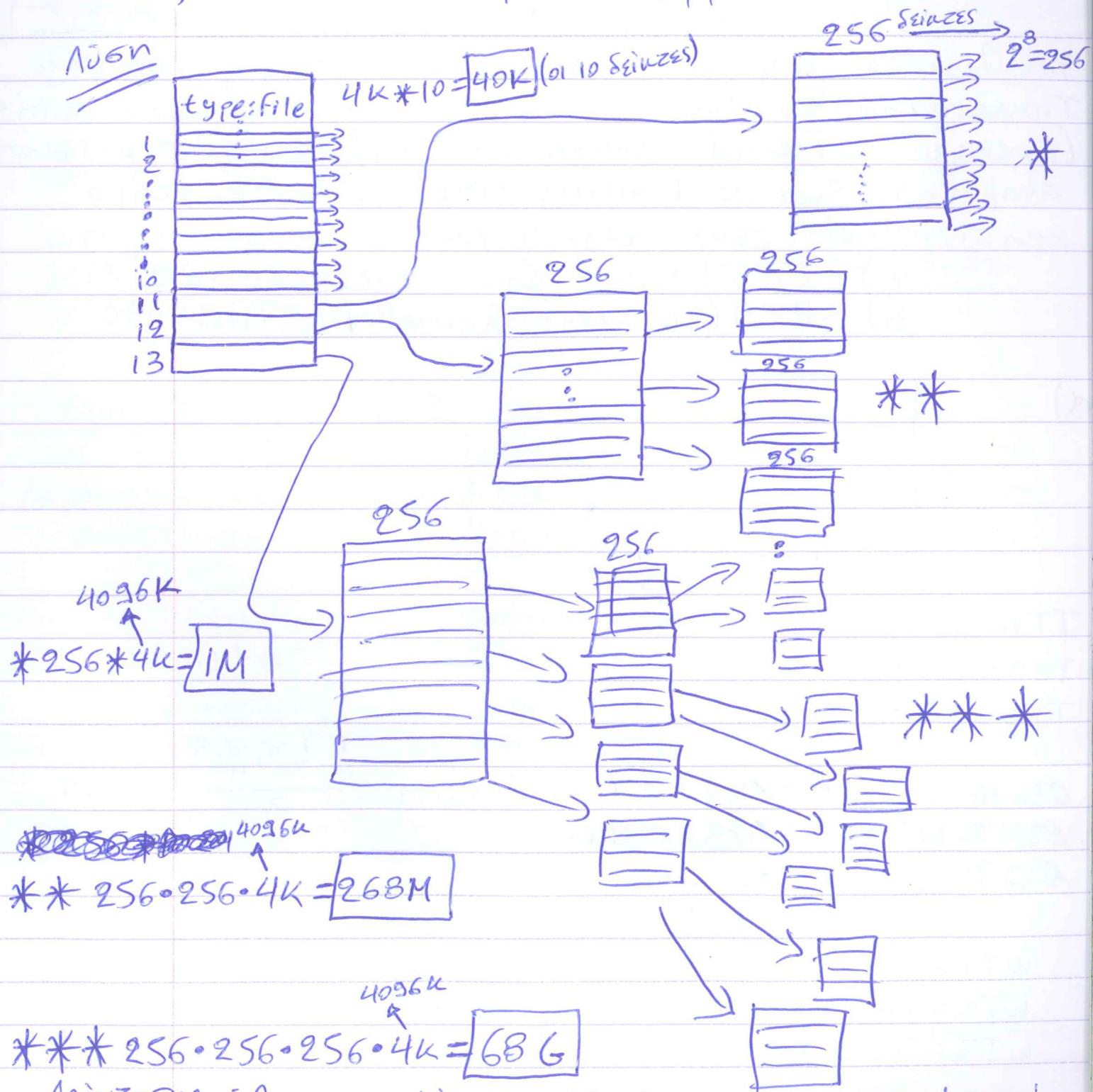
NiGN

Semaphore • $P_1 = 2$, $C_1 = 0$, $C_2 = 0$

Process Producer	Process Consumer 1	Process consumer 2
:	:	:
repeat	repeat	repeat
down(P_1)	down(C_1)	down(C_2)
add_item	down(C_1)	consume_item
up(C_1)	consume_item	up(P_2)
:	up(C_2)	up(P_2)
until Forever	until Forever	until Forever

Ωρια 4

Ποια είναι τα περιγραφέα αρχείου που μπορεί να ανοθυσεύεται σε ένα δίσκο ή να γίνεται
δεικτός ή είναι περιγραφέα 8 bytes και μάζε link είναι
4K; Σε μάζε i-node υπάρχουν 10 αρεσοι δεικτές
είναι ανταί επεργοι δεικτών, είναι διαδικτικοι δεικτών δεικτών, και είναι ζειντά επεργοι δεικτών.



Anάγνωση: Από τα περιγραφέα αρχείου που μπορεί να ανοθυσεύεται είναι: $40u + 1M + 268M + 68G$

Ιούνιος 2020

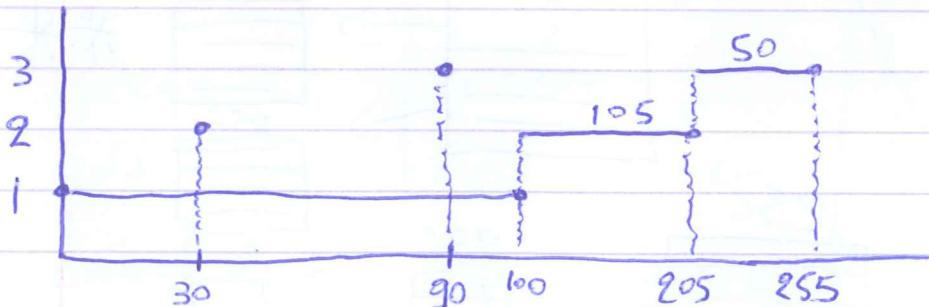
Θέμα 1

Σε ένα σύστημα που δεν χρέω ο είναι αδρανής, έχουν ταξιδέψει 3 jobs για εγκέντρη τε και παραγάγουν χαρακτηριστικά:

Job	Xρόνος Ηφίγυς	Xρόνος Εγκέντρων
1	0 SEC	100 SEC
2	30 SEC	105 SEC
3	90 SEC	50 SEC

Υπολογίζεται τος μέσος χρόνος επιβροφής (Average Turnaround Time), τος αντιτέθεντος χρόνος επιβροφής (Average Weighted Turnaround Time) και τος μέσος χρόνος αναπονής (Average Waiting Time) αν τα σύγχρονα χρηματοποιείταις αδιαρρέως.

- a) FCFS (First Come First Served)
- b) SJF (Shortest Remaining Time)



$$TT_1 = 100 - 0 = 100$$

$$TT_2 = 205 - 30 = 175$$

$$TT_3 = 255 - 90 = 165$$

$$ATT = \frac{100 + 175 + 165}{3} = 146,66$$

~~$$WTT_1 = 100 + 100 = 200$$~~

~~$$WTT_2 = 175 + 105 = 280$$~~

~~$$WTT_3 = 165 + 50 = 215$$~~

$$AWTT = \frac{0 + \frac{1.66}{3} + \frac{3.3}{3}}{3} = 1.98$$

$$WT_1 = 100 - 0 = 100$$

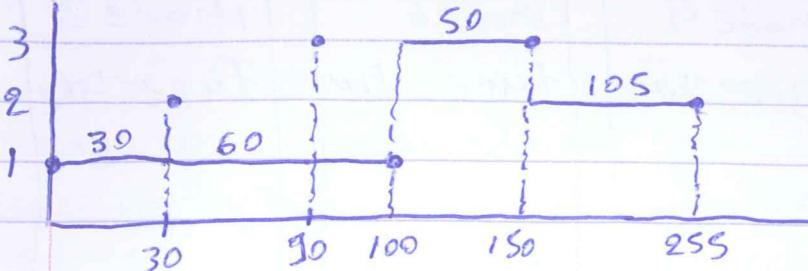
$$WT_2 = 100 - 30 = 70$$

$$WT_3 = 205 - 90 = 115$$

$$AWT = \frac{100 + 70 + 115}{3} = 61.66$$

$$100 - 30 = 70 - 60 = 10$$

B)



$$TT1 = 100 - 0 = 100$$

$$TT2 = 225 - 100 = 125$$

$$TT3 = 255 - 150 = 105$$

$$WTT1 = 100 / 100 = 1$$

$$WTT2 = 225 / 125 = 1.8$$

$$WTT3 = 105 / 105 = 1$$

$$WT1 = 100 - 100 = 0$$

$$WT2 = 225 - 100 = 125$$

$$WT3 = 105 - 50 = 55$$

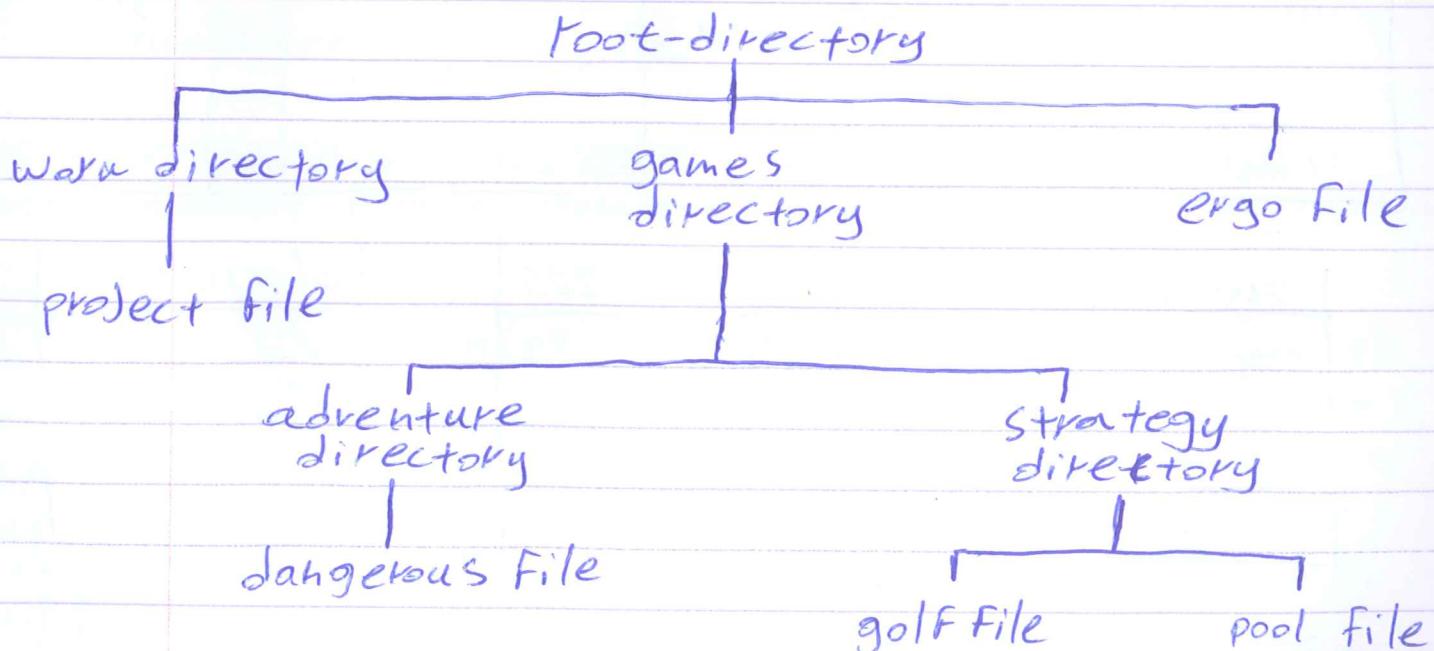
$$ATT = \frac{100 + 225 + 105}{3} = 128.3$$

$$AWTT = \frac{1 + 1.8 + 1}{3} = 1.44$$

$$AWT = \frac{0 + 125 + 55}{3} = 43.33$$

Örök 2

Na δύεται τα i-nodes και data blocks για να
μαρτυρήσουν σένας (μαθητής εγείς τα routes που γίνονται
τα i-nodes και data blocks)



inode	file	inode 4	inode 6	inode 3
1	..			
1	..	type:dir	type:dir	type:file
4	worm			
6	games			
3	ergo	500	600	700

block 500	block 600	block 700	inode 10
4 ..	6 ..		
1 ..	1 ..		
10 project	12 adventure	(Σεδοφέρα απτείου)	type:file
	14 strategy		800

block 800	inode 12	inode 14	block 900
(Σεδοφέρα απτείου)	type:dir	type:dir	
	900	1000	

inode 16	block 1100	block 1000	inode 18
type:file	(Σεδοφέρα απτείου)	14 .. 6 .. 18 golf 20 pool	type:file
1100			1200

inode 20	block 1200	block 1300
type:file	(Σεδοφέρα απτείου)	(Σεδοφέρα απτείου)
1300		

Θέμα 3

Έτσι ότι δίνεται πα σπουδαση του προβλήματος του παραγωγού μαζαναδυτή τε δύο παραγωγούς και δύο μαζαναδυτές που επενδύνται αρχικά πα ταυτόχρονα. Το γεγοντα έχει τους παραγωγούς περιορισμένους

- 1) Ηδε διεργασία έχει ήταν αριθμός κτικα που περιορίζεται σε 1. Εάν έχει ήταν διαθέσιμο από έτσι ήταν αποτέλεσμα.
- 2) Οι μαζαναδυτές δεν μπορούν να επενδύσουν (loop) σε αριθμό κτικα που πριν γράψουν (ηδε φορά) και οι δύο παραγωγοί.
- 3) Κανένας από τους παραγωγούς δεν μπορεί να επενδύσει (loop) σε αριθμό κτικα που πριν διαβάσουν και οι δύο μαζαναδυτές. Εγαίρεται αποτέλεσμα το πρώτο γράψαντα των παραγωγών.
- 4) Μα τι δύο παραγωγοί μπορούν να γράψουν ταυτόχρονα και οι δύο μαζαναδυτές μπορούν να διαβάσουν ταυτόχρονα

Χρησιμοποιείτε semaphores για να επιτύχετε ατοιχαίο ανοιχτείστο και ευχρηστικό παραγωγού.

~~Διάλογος~~

Semaphore p1 = 2, p2 = 2, c1 = 0, c2 = 0

Process Producer1	Process Producer2	Process Consumer1	Process Consumer2
loop	loop	loop	loop
... down(p1)	... down(p2)	... down(c1)	... down(c2)
down(p1)	down(p2)	down(c1)	down(c2)
procedure	procedure	procedure	procedure
up(c1)	up(c2)	up(p1)	up(p2)
up(c2)	up(c1)	up(p2)	up(p1)
...
until forever	until forever	until forever	until forever

Θετα 4

Δινέται η παραπάνω μαζίσασαν ενδιαφέρουσας

Δοθέντα		Μέσιγρα		Διαθέσιμα	
		A	B	C	D
p0	0	0	1	2	0
p1	1	0	0	0	1
p2	1	3	2	4	7
p3	0	6	0	2	3
p4	0	0	1	4	2
					5
					6
					9
					0

Ανανιγίζεται έτσι παραπάνω τις αλγόριθμοι αποφυγής αδιεξόδου:

- a) Ποτα είναι για περιεχόμενα του πινακα "Ανοικούσες Αιχνές"
- b) Είναι για διασημά για αδιεξόδο;
- c) Αν εισέλθει αιχνή από p1 ή (0,3,2,0) θα πρέπει να εξαπορευτεί από την αιχνή;

i)

Ανοικούσες Αιχνές		(Μέσιγρες - Δοθένες)			
		Περιεχόμενα του πινακα Ανοικούσες			
A	B	C	D		
0	0	0	0		
0	7	8	2		
1	0	0	2		
0	0	8	0		
4	2	4	9		

ii)

Δοθέντα		Ανοικούσες		Διαθέσιμα	
		A	B	C	D
p0	0	0	1	2	0
p1	1	0	0	0	1
p2	1	3	2	4	7
p3	0	6	0	2	3
p4	0	0	1	4	2

Είδεται οι Ανοικούσες ≤ Διαθέσιμες

Με τον παριόρων μια διεργασία στην οποία
πόρους εποιείται. Οι ανωμένες μενούν
ταΐες.

	Δοστένοι				Ανοκένουσες				Διαθέσιτοι			
	A	B	C	D	A	B	C	D	A	B	C	D
• P0	0	0	0	0	0	0	0	0	0	0	0	0
• P1	0	1	0	0	0	7	3	2	1	5	3	2
• P2	1	3	2	4	1	0	0	2	1	3	2	4
P3	0	6	0	2	0	0	8	0	+	+	+	2
P4	0	0	1	4	4	2	4	2	+	3	2	4

	Δοστένοι				Ανοκένουσες				Διαθέσιτοι			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	0	0	2	8	5	6
P1	1	0	0	0								
P2	0	0	0	0								
P3	0	6	0	2								
P4	0	0	1	4								

Anάντην ΔΕΝ ταριχεύονται όπες οι διεργασίες
αλλά λένε ν Ρ0 και P2 γιαντό και
το σύστημα είναι σε αδιέξοδο

Γ) Αφαίρω στην αίσιην P1 (0, 8, 2, 0) από τον πορό
από την αντιστοιχη θέση του P1 (Δοστένοι) και *

	Δοστένοι				Ανοκένουσες				Διαθέσιτοι			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	0	0	1	5	2	0
P1	1	8	2	0	0	7	3	2				
P2	1	3	2	4	1	0	0	2				
P3	0	6	0	2	0	0	8	0				
P4	0	0	1	4	4	2	4	2				

* Ελέγχω αν έχω διαθέσιτους πόρους για την επόμενη ανοκένουση.
Διαθέσιτοι (1, 5, 2, 0) \leq (0, 7, 3, 2)

Δεν μπορούμε να εγκατεβούμε στην P1 γιατί
δεν έχουμε αριεζούσι διαθέσιτους πόρους.

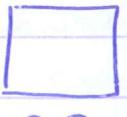
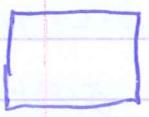
Άρτα 5
Δίνεται ο παρακάτω κώδικας:

Repeat

Do something
until Done

Η διεργασία "Do something" θέτει διαδοχική προβολή των 10 γελιδιών τάσας κυκλών σε πάθεια για 6 πλαισια. Υποθέτεται ότι η μηδενική διεργασία διαρρέει 6 πλαισια των λόγρων κυκλών, και οι κατια ανά τις 10 γελιδιές δεν είναι μηδενική αρχικά.

Προσπελαγεί ανά την Do something





P1

P2

P3

P10

a) Ποια είναι η σχέση ανάτετα στον αριθμό των σφαλτών γελιδιών και στον αριθμό των μηδενικών της διεργασιών "Do something" ήταν χρησιμοποιείται ο αδύοτος LRU;

✓6η

Ενεργεί από γελιδιές της διεργασίας (10) δεν χωρούν μηδενική (6 πλαισια) και δίνεται δραστική προβολής, ο αριθμός των σφαλτών είναι:

$$\Sigma \Sigma = 10 * h$$

$h \rightarrow$ αριθμός μηδενικών της διεργασιών

Επεξετροπός 2020

Ωρια 1

Διένεση η παρακάτω δείρα αιχμών για σελίδες της ιδεακής μνήμης:

14, 15, 21, 19, 5, 21
 5, 10, 12, 5, 16, 10, 5, 16, 13, 14, 5, 15, 19, 5, ~~10, 10, 10, 10, 10, 10, 10, 10~~

~~10, 10, 10, 10, 10, 10, 10, 10~~

Θεωρήσε οι η φυσική τιμή χωρίς επεξετροπή και οι τιμές της TLB πλορεί να αποθηκεύεται 4 σελίδες. Ο πινακάς χαραγμάτων σελίδων (αποθηκευόμενος σεντράλη) είναι:

Σελίδα Ιδεακής Μνήμης	Αντιστροφό Πλαίσιο Αφύσικης Μνήμης
0	-
1	-
2	6
3	-
4	-
5	-
6	-
7	7
8	-
9	-
10	1

Σελίδα Ιδεακής Μνήμης	Αντιστροφό Πλαίσιο Αφύσικης Μνήμης
11	4
12	5
13	-
14	-
15	-
16	0
17	-
18	-
19	-
20	2
21	3

Πριν γεννιγούν οι παρακάτω αιχμές, ο TLB έχει ζα περιεχότερα να δινορτά παρακάτω. Θεωρήσε οτι το TLB και η φυσική τιμή χρησιμοποιούν πολιτική FIFO για την αντικατάσταση των σελίδων και οι σελίδες έχουν την τιμή της δείρας, γενινώντας από τη μηρόσερη διεύθυνση της μνήμης και του TLB.

Σελίδα Ιδεακής Μνήμης	Πλαίσιο Αφύσικης Μνήμης
16	0
10	1
20	2
21	3

- 1) Να σειρέσε την σειρή μαζίσματος του TLB για να δοθείται σύρπα αντίγεων.
- 2) Πότο είναι το hit ratio του TLB;
- 3) Πότες αναφοράς γιαν πρίν θα γίνουν για την κατάχυτων των παρατάνω σελίδων, αγνοώντας την αναφορά σελίδας μαζί το page fault.
- 4) Να εναντιτάξετε το ερώτημα (3) αν δεν υπάρχει TLB.

Ιεράρχημα: Σε περιπτώσεις page fault, δεμπρίσεται οι παραπάνω σελίδες αντιγράφεται από τη διεύθυνση ουσα πρίν την φύσιση της είσβεσης στο TLB. Ενιδέον, το TLB θα προσειπει να έχει σελίδες που δεν είναι γιαν πρίν

Ιεράρχημα

1)	0	-	11	A
	1	-	12	B
	2	B	13	C ³
	3	-	14	D ⁴
	4	-	15	E ⁵
	5	F ⁰	16	F ¹
	6	-	17	-
	7	F	18	-
	8	-	19	F ⁷
	9	-	20	G ⁸
	10	X ²	21	H ⁶

16	B ¹⁰	Φ	Φ ²
17	C ¹¹	Φ	Φ ⁶
18	D ¹³	X	Φ ⁷
19	E ¹⁵	Φ	Φ ⁸
20	F ¹⁵	X	Φ ⁹

5	10	12	5	16	10	5	16	13	14	5	15	10	5	14	15	21	19	5	21
PF*	✓	V	✓	*	*	V	✓	*	*	V	*	V	V	V	V	*	*	✓	✓
TLB*	V	*	✓	*	*	V	V	*	*	*	*	V	V	V	V	*	*	*	✓

Ιεράρχημα
Η σειρή μαζίσματος του
TLB είναι : \Rightarrow

Paged#	Frame#
10	2
21	6
19	7
5	0

2) Hit ratio: $8/20 = 0,4 = 40\%$

3) κάθε hit (TLB) \Rightarrow 1 Αναγρων
κάθε miss (TLB) \Rightarrow 2 Αναγρωσις

$$(8 * 1) + (12 * 2) = 32 \text{ αναφορές}$$

4) Αν δεν υπήρχε TLB κάθε αναγρων θα έδει
2 αναφορές

Apa $(20 * 2) = 40 \text{ αναφορές}$

Άστα 3

Έσω οζι γε ένα σύγκριτη υπάρχουν τέρο δύο διεργασίες. Ας ναλ Δι που γράφουν τέσσερα υβάρτα 100 χρονιες ποράδες και χρηματοποιούν στην λόγη του PETERSON, όπως φαίνεται παραπάνω. Έσω οζι γενικά είναι αρχικά ν Δι να οι 100 χρονιες ποράδες αρχιούν πέκπι να τεθει ~~t=0~~ turn = process, (αφού αδηλώνει ν turn). Σια εννυ Δο, οι 100 ποράδες αρχιούν για να ήνει έσω δρόγχο while εντολής εισαγωγής 620 κρίσης γράφτα. Σε ποια χρονια γράφτη θα μνου γε KT οι δύο διεργασίες (αν θα μνου); Εινινίζει ανά το χρόνο ~~t=0~~ t=0 και δεμπίζει τον χρόνο του KT καθώς και τον χρόνο πετρωτήν εντολών ως αποτέλεσμα.

```
#define FALSE 0
#define TRUE 1
#define N 2 // number of process
int turn; // whose turn is it
int interested[N]; // all values initially 0 (FAILE)
void enter_region(int process){
    int other;
    other = 1 - process;
    interested[process] = TRUE;
    turn = process;
    while (turn == process && interest[other] == TRUE) // null state
}
void leave_region(int process){ // process: who is leaving //
    interested[process] = FALSE;
}
```

ΛΟΓΙΣΤΗΚΑΝ Η σειρά εναλλαγής είναι Δι - Δ2 - Δ1(KT) - Δ2(KT). Έσω δρώντα υβάρτα γίνεται turn = 1, έσω σύζερο turn = 0. Απα ν Δι γναίρει δέσμη KT γε χρόνο 200 και ν Δο γε χρόνο 300 ποράδες.

Σήμα 4

Δινέσαι ένα σύστημα Linux με 4 διεργασίες A, B, Γ, Δ
και αριθμούς προτεραιότητων 105, 107, 117, 125 αντίστοιχα.
Οι διεργασίες κατέφθασαν σε χρόνο $t=0$ με αλφαριθμητική σειρά. Να δώσεται τον χρόνο επεξεργασίας κάθε διεργασίας
και κάθε φανάρι ανά αντίστοιχη διεργασία. Επίσης (run time)
= 1000ms. Ο χρόνος υπολογισμού των κώδικων
κατούει στην πολιτική του $O(1)$ χρονοδρομούσης
και οι διεργασίες είναι CPU bound (σεντ κώδικας 110).

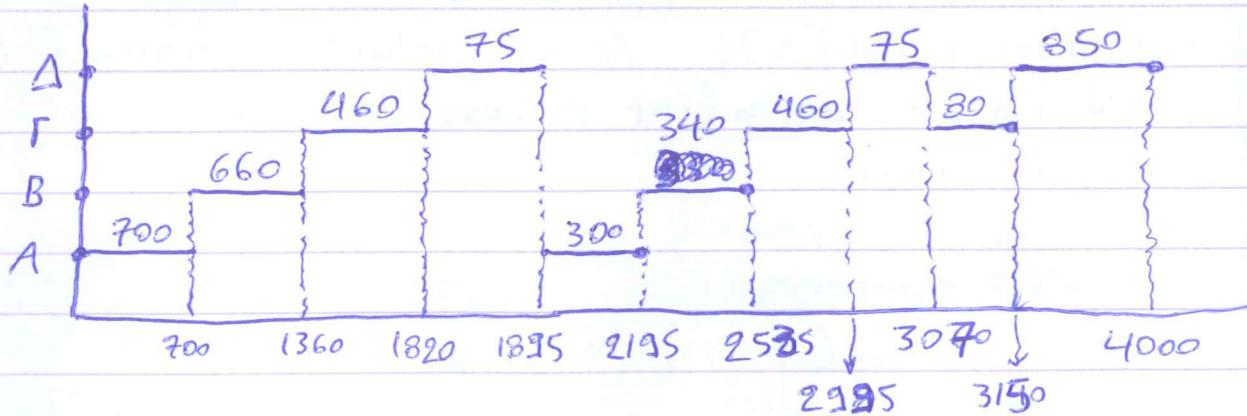
JGN

$$105 : A = (149 - 105) * 20 = 35 * 20 = 700 \text{ ms}$$

$$107 : B = (149 - 107) * 20 = 33 * 20 = 660 \text{ ms}$$

$$117 : \Gamma = (149 - 117) * 20 = 32 * 20 = 640 \text{ ms}$$

$$125 : \Delta = (149 - 125) * 5 = 24 * 5 = 120 \text{ ms}$$



Από

A → 2195ms χρόνος επεξεργασίας

B → 2535ms χρόνος επεξεργασίας

Γ → 3150ms χρόνος επεξεργασίας

Δ → 4000ms χρόνος επεξεργασίας

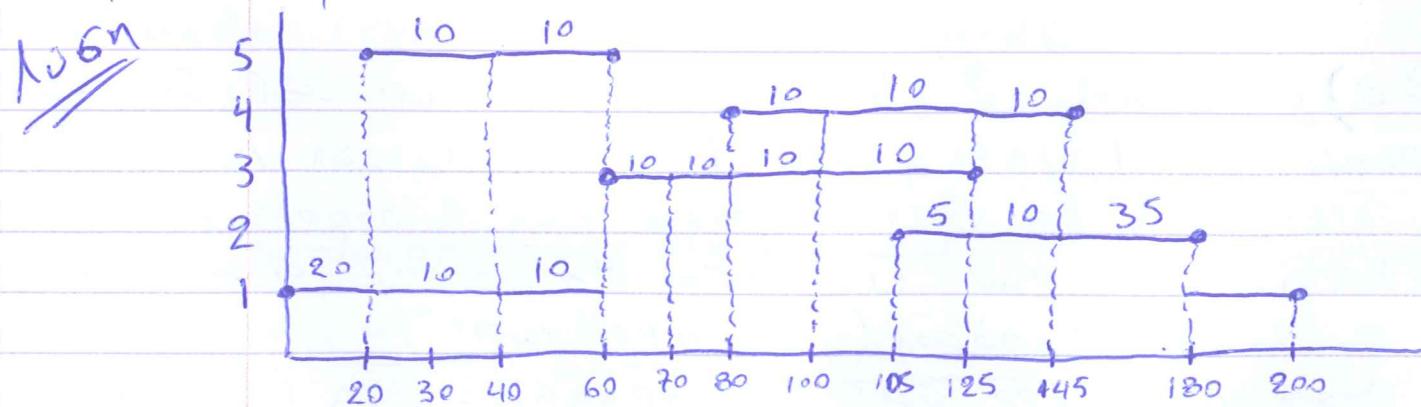
Άσκηση 5

Ένα σύστημα χρησιτούσει έναν αλγόριθμο για προσεταιρίσεις (priorities) για την χρονοδιάγραμμα, σε αντίστοιχη σειρά ή ως ένα περιφρόνης (RR). Αναδιπλό, τόνοι οι διεργασίες που έχουν την υψηλότερη προτεραιότητα και είναι η πρώτη που πάγιζεται κάθε φορά την χρήση της CPU. Το σύστημα στα χρόνια 0 είναι νευρό (idle). Ερχονται 5 έργασις για εκτέλεση για τα παρακάτω χαρακτηριστικά:

Διεργασία	Priority	Xρόνος Αφίξης	Xρόνος Εκτέλεσης
1	1	0 SEC	60 SEC
2	3	105 SEC	50 SEC
3	3	60 SEC	40 SEC
4	3	80 SEC	30 SEC
5	1	20 SEC	20 SEC

Ανάγεται το χρονοδιάγραμμα εκτελεσης των διεργασιών καθώς και τους Average Turnaround Time, Average Weighted Turnaround Time,

Επιειδή Για τις προτεραιότητες, θεράπευτης νούτρου περιλαμβάνει περιτύπων προτεραιότητα.



$$TT1 = 200 - 0 = 200$$

$$TT2 = 180 - 105 = 75$$

$$TT3 = 125 - 60 = 65$$

$$TT4 = 145 - 80 = 65$$

$$TT5 = 60 - 20 = 40$$

$$WTT1 = 200 / 60 = 3.3$$

$$WTT2 = 75 / 50 = 1.5$$

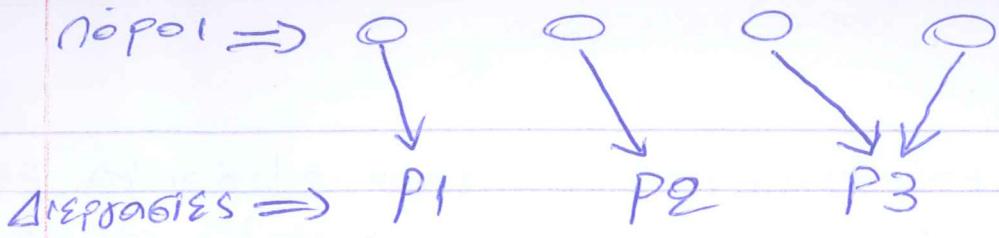
$$WTT3 = 65 / 40 = 1.6$$

$$WTT4 = 65 / 30 = 2.1$$

$$WTT5 = 40 / 20 = 2$$

$$ATT = \frac{200 + 75 + 65 + 65 + 40}{5} = 89$$

$$AWTT = \frac{3.3 + 1.5 + 1.6 + 2.1 + 2}{5} = 2.1$$



έρα 6

Ευρήσεις ήταν σύσητα για 4 ίδιους nōpoous που λοιπάζουν
ε 3 διέργασιες. Κάθε διέργασια ανατζει όχι
περισσότερους από 2 nōpoous. Οταν για διέργασια γίνεται
νόρος, το διευρύνει σύσητα δίνει οποιονδήποτε
ιαθέσιμο nōpo. Αν κανένας nōpo δεν είναι διαθέσιμος
διέργασια οριτέψει.

Είναι δυνατόν να έχουμε αδιέξοδο με σύσητα αυτού;
ναι, δώσεις μια γειρά αιχμέων που μαρατήσει τη
διέξοδο. Αν όχι, ποιες από τις 4 συνδικες αναγνωρίζεις
α αδιέξοδο δεν υπάρχουν με σύσητα αυτού;
Αναντίσεις Ridi στην ερώτηση (a) υποδιώκεται οτι
δε διέργασια χρειάζεται όχι περισσότερους από 3
nōpoous.

6a) Όχι, γιατί αν έχουμε 4 nōpoous παι πάθει διέργασια
πάρει από 1, πας τέλει 1 nōpo. Με το που μαρτί^{τη}
ια διέργασια θα έχει σίγουρα έως 2 nōpoous για να
ισχεδιστεί. Μετά την ευγένεια θα επιτρέψει τους nōpoous
αλ θα τινούν 2 nōpois (διαθέσιμοι χωρίς να χρησιμοποιούνται)
αλ ~~ποτέ~~ από 1 nōpo δεις 2 διέργασιες, αρά θα ευγένεια
δεις πατ δεν υπάρχει αδιέξοδο.

Ναι, γιατί αν έχουμε 4 nōpoous παι πάθει διέργασια πάρει
τη 1 nōpo τόσες γίνει 1 διαθέσιμος nōpo παι πατ δεν
παρει να ολοι πρωθει πατια διέργασια.

ΠΑΡΑΔΕΙΓΜΑ : P1, P2, P3, P1, P2
ΑΔΙΕΞΟΔΟΥ :

Επεξιφρος 2022

Ρέψα 1

Ένα υπολογιστικό δισκότα διαθέτει 4Mbytes τυχη για zis διεργασίες. Το λειζουργικό δισκότα διαχειρίζεται την τυχη ώς ηλεγία 64bytes καθέδου. Ένας αναυας ηλεγίων, που αποθηκεύεται σε αυτά τα 4Mbytes τυχης, περιέχει την παραπομπή κάθε ηλεγίου της τυχης. Θέλουμε να υπολογίσουμε πόσο θέλαμε να πρέπει να είναι κάθε ηλεγίο.

Αναυας ηλεγίων	Διαθέσιμος χώρος για διεργασίες
-------------------	---------------------------------

4MByte Μνήμη

Οι 6χειστικές αποφάσισαν ότι τα πέντε των ηλεγίων προπει να είναι τις 2K, τις 4K, τις 8K bytes. Ποια αποφάσισαν τις επιλογές τους για ελαχιστοποίηση του χώρου που δεν χρησιμοποιείται ανά την διεργασία ενεργίας της οποίας είναι παρατηρήσιμη ανά την αναυα ηλεγίων τις 2K bytes. Η αποφάσιση αυτή οφείλεται στην παρατηρήση ότι τα 2K bytes έχουν μετατρέψει την τυχη σε καρακτήρες (fragmentation).

- Η αποφάσιση ότι τα 10 διεργασίες δημιουργούνται στην τυχη. Οι πέντε της τυχης που παρατηρήστηκαν είναι 1/2 ηλεγίων για κάθε διεργασία.
- Ο αναυας ηλεγίων ζητείται για εγγραφή για κάθε ηλεγίο και κάθε εγγραφή χρειάζεται 10 bytes.

ΜΩΒ

$$4\text{MB} = 2^{22} \quad , \quad 2\text{K} = 2^{11}$$

In περιπτώσει: (2K ηλεγίο)

$$\frac{2^{22}}{2^{11}} = 2^1 = 2048 \text{ διεργασίες}$$

bytes *

$$10 \cdot \frac{1}{2} = 5 * 2\text{K} = [10\text{K}] \text{ παρατηρήσεις της fragmentation}$$

$$2048 * 10 = [20480 \text{ bytes}] \text{ για την αναυα}$$

Apa Σύνολο: $20480 + 10\text{K} =$ 30720 bytes

10240 byte

~~QUESTION~~

en nepincwony: (4K niajcio)

$$4K = 2^{12}$$

$$\frac{2^{22}}{2^{12}} = 2^9 = \underline{1024 \text{ stepsizes}}$$

$$10 \cdot \frac{1}{2} * 4K = 5 * 4K = \boxed{20K} \text{ dojw fragmentation}$$

$$1024 * 10 = \boxed{10240 \text{ bytes}}$$

Apa cijvoro: $10240 + 20K = \boxed{30720} \text{ bytes}$

en nepincwony: (8K niajcio)

$$8K = 2^{13}$$

$$\frac{2^{13}}{2^{22}} = 2^9 = \underline{512 \text{ stepsizes}}$$

$$10 \cdot \frac{1}{2} * 8K = 5 * 8K = \boxed{40K} \text{ dojw fragmentation}$$

$$512 * 10 = \boxed{5120 \text{ bytes}}$$

Apa cijvoro: $40K + 5120 = \boxed{46080 \text{ bytes}}$

Δίκτα 9

priorities

Δινέζαι η νύχτα Linux και οι διεργασίες A, B, Γ, Δ & Ε αριθμούς προτεραιοτήτων 103, 107, 116, 127 αριστούνται. Οι διεργασίες ωρίμαναν σε χρόνο $t=0$ την αλφαριθμητική σειρά. Να δινέζεται τον χρόνο σερβισισμού καθε διεργασία αν η αθετία από αυτές έχει χρόνο επεξεργασίας (run time) = 1000 ms. Ο χρόνος υπολογισμού των πλάνων αναλυθεί στην πολιτική του O(1) χρονοδρομολογίας και οι διεργασίες είναι CPU bound (δεν υπάρχουν I/O)

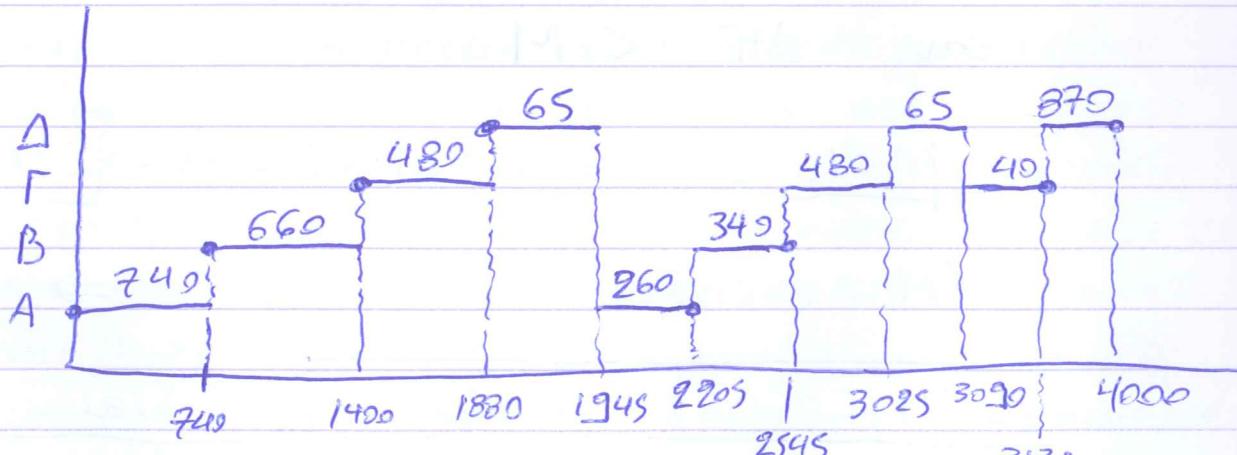
Λύση

$$103: A = (140 - 103) * 20 = 740 \text{ ms}$$

$$107: B = (140 - 107) * 20 = 660 \text{ ms}$$

$$116: \Gamma = (140 - 116) * 20 = 480 \text{ ms}$$

$$127: \Delta = (140 - 127) * 5 = 65 \text{ ms}$$



A → 2205ms

B → 2545ms

Γ → 3090ms

Δ → 4000ms

ΠΡΟΣΟΧΗ ΤΑ MS

Bitmap

έρα 3

Η παραπομπή του ελεύθερου χώρου για δίσκου προσινά
ναι ότι στην χρήση μιας λίγας ελεύθερων πύλων ή ενός χαρακτηρικού.
Οι διευδύνεις του δίσκου ανταντούν Δ bits. Αν έχεις
λίγες έχει χωρητικότητα Μ πύλων από τα οποία Ε
τίνει ελεύθερα, δώσε στην ευθύνη της στην οποία
η λίγα ελεύθερη πύλων χρησιμοποιείται αριθμός χώρου
που θα ξαφνικώνεται. Αν το Δ έχει την $[16 \text{ bits}]$
κατανομή στην ανάνεωση GAS WS ποσοστό είναι 20%
κατανομή του χώρου δίσκου ο οποίος πρέπει να
τίνει ελεύθερος για να λεχύνει την ευθύνη.

~~JMN~~

Bitmap \rightarrow Δ bits

Lίγα \rightarrow $\Delta * E$

Η λίγα χρειάζεται αριθμός χώρου από το Bitmap
όπως $\Delta * E < M$

$$\text{Αν } \underline{\Delta=16} \text{ κατανομή } \Delta * E < M \Rightarrow E/M < \frac{1}{16}$$

$E/M = 0.0625$ ελεύθερων πύλων \Rightarrow

$$\boxed{E/M < 0.0625 \text{ ή } 6.25\%} \text{ του δίσκου είναι κενό.}$$

Άσκηση 4

A. Σε υπολογιστές που διαθέτουν την εντολή TSL (Test and Set Lock) είδησε ότι γνωρεί να γράφει μια pouziva εισόδου σε κριτικό χώρο ως egnis:

enter:

```
TSL REGISTER, LOCK
CMP REGISTER, #0
JNE, enter
RET
```

* (#0... contains
#0... begin writing)

Έχω τύπο ότι ο υπολογιστής πας δεν διαθέτει την εντολή TSL, αλλά διαθέτει την εντολή που αναπτύχθηκε για περιεκότερα ενσύντητη λειτουργία. Μπορεί αυτή η εντολή να χρησιμοποιείται για την pouziva enter. Αν ναι γράψε την pouziva αυτή.

B. Εάν random access αρχείο έχει records καθορισμένη θέσης αντίτοιχης αρχείου χρησιμοποιείται βείριακά ανά διάφορες διεργασίες που γράφουν στο αρχείο αυτό. Καθε διεργασία γράφει στο ίδιοκείο record του αρχείου. Γράψε του ψευδοιώδη την διεργασία αυτών, χρησιμοποιώντας binary semaphores. ΠΡΟΣΟΧΗ: το λειτουργικό σύστημα δεν αντέτει ανά πόρο του την δεινή του αρχείου (file point). Δεν υπάρχουν processes που να διαβάζουν ανά το αρχείο.

Α.

enter:

```
SWAP REGISTER, LOCK
SET LOCK, #1
CMP REGISTER, #0
JNE, enter
RET
```

B
⇒

```

B. Semaphore s1=1;
int file_p=0;
write() {
    while(true) {
        data();
        down(s1);
        file_p++;
        write(file_p);
        up(s1);
    }
}

```

Άριθμα 5 }

Δινέται ένα γογζίκα που απορρείται από 5 λόρους του
ιδρου ζύουν οι οποίοι δημιουργούνται σε 4 διεργασίες. Οι
λόροι γιατί τούνται και αποδίδονται από τις διεργασίες
ένας κάθε φορά (δηλαδι κάθε αρχή από την άνωθεν ένας λόρος
κνοεί η διεργασία να γινθεί του ξεκίνησε). Η απόδειξη
ότι το γογζίκα δεν κνοείται φθάνει σε αδιέξοδο
αν 16χιούνται οι εγγίσεις δύο βυνθίσκες:

- 1) Ο περιγραμμός από τον πόρο που κνοείται γιατί
κάθε διεργασία (περιγραμμές ανάγκες) είναι ανάγκες σε
1 και ως 5.
- 2) Το αδρογότα όλων των περιγραμμών ανάγκες είναι πιο πολύτερες
από ως 9.

Λύση

5 λόροι
4 διεργασίες

Αρά ξέπουλε ότι δεν το γογζίκα
δεν είναι σε αδιέξοδο οι καν
[16χιές] περιγραμμές ανάγκες < λόροι + διεργασίες

\Rightarrow Μέσης ανάγκες < 5+4

\Rightarrow Μέσης ανάγκες < 9 που 16χιές



Αρά το γογζίκα δεν
είναι σε αδιέξοδο

ΦΕΒΡΟΥΑΡΙΟΣ 2023

Πίκα!

ΜΕΣΡΙΓΕΙΣ ΓΕ ΕΥΧΕΔΑΡΙΤΗΡΟ ΕΥΘΥΝΗ ΈΧΟΥΝ ΔΕΙΓΕΙ, ΉΤΑΝ ΤΑ
ΤΕΩΝ ΔΙΕΡΓΑΣΙΑ ΣΥΣΤΕΛΕΙΖΑΙ ΓΙΑ ΧΡΟΪΟ Τ ΠΡΙΝ ΑΝΑΣΤΑΣΕΙ ΓΙΑ
ΣΙΓΟΣΟΛ/ΈΓΟΔΟ. Η ΕΥΑΓΓΑΓΗ ΔΙΕΡΓΑΣΙΑΣ ΑΝΑΙΖΕΙ ΧΡΟΪΟ
Ε, Ο οποίος θεωρείται χαρένος χρόνος επειδή η ευαγγά^{γη}
λογίζεται ως επιθαρυβων. Για χρονοπρογραφήσιτο εγκυρώσεων
ΕΙΝΑΙ ΝΕΡΙΖΟΝΙΣ ΚΣ ΉΒΑΝΤΟ ΧΡΟΪΟΥ Κ, ΔΙΩΓΖΕ ΖΟΥ ΖΩΝΤΑ
ΑΝΟΔΟΣΙΚΩΝΤΑΣ ΖΩΝ CPU, ΓΙΑ ΉΔΗΣ ή ΜΑ ΑΝΤΙ ΖΩΣ
ΑΝΟΔΟΥΣ ΠΡΟΖΑΓΕΙΣ:

- | | |
|----------------|------------------|
| a) $K = 0$ | δ) $K = E$ |
| b) $K > T$ | ε) $K \approx 0$ |
| c) $E < K < T$ | |

~~Νο 6η~~ d) Αφού ζω ήβαντο είναι τεράστιο μήδε διεργασία ήα διανοει
όταν γινήσει I/O, διηδύνεται από ζω χροΪο T. Εποχήνως:

$$U = \frac{T}{T+E}$$

b) Αφού ζω ήβαντο είναι πραλύζερο ζου χροΪου που θα γρέζει
χωρίς διανοει μία διεργασία, μήδε διεργασία ήα διανοει ζων
~~γινήσει~~ γινήσει I/O, διηδύνεται από χροΪο T. Εποχήνως:

$$U = \frac{T}{T+E}$$

c) Αφού ζω ήβαντο είναι πραλύζερο από T, μήδε διεργασία ήα ευθυδηρί^{σει}
ήα ήβαντο πριν διανοει. Εποχήνως: $U = \frac{U}{U+E}$

d) Το ήβαντο υποζίθεται ότι είναι πραλύζερο ζου T, αλλάς ήα εικάε
ζων περινων CB1. Εχουτε δοινόν ζων περινων (g) ήε ίσα ζω
K και E. Εποχήνως: $U = 50\%$

e) Αφού ζω ήβαντο είναι 6χεδών i/o ήε ζω 0, έχουτε ζων
περινων (g) ήε πάρα πολὺ πιο ζω K. Εποχήνως:

$$U \approx 0$$

Είδη 2

Μεταξύ των παραπάνω προγραμμάτων αφορούν αυτόν τον καλόντει τις ανατριχίες πους "κατίν σών"; Αν υπάρχει ενέργεια πήδησα για την αρχή της προγράμματος. Αν ούτι εγκληματική.

program/module mutex4

...

var p1using, p2using : boolean

process p1

begin

while (true) do

begin

p1using=true;

while (p1using) do

begin

p1using=false;

p1using=true;

end; {while}

critical_section;

p1using=false;

other-p1-processing

end; {p1}

process p2

begin

while (true) do

begin

p2using=true

while (p2using) do

begin

p2using=false

p2using=true

end; {while}

critical_section;

p2using=false;

other-p2-processing

end; {p2}

{parent process}

begin {mutex4}

p1using=False;

p2using=False;

initiate p1, p2

end; {mutex4}

*initiate enthalves γενικά
τις αυτούρdes διεργασίες

Nu⁶ⁿ



Ajgn Ο αποταμιας αποκλειστος ειναι πρι αδιόχησα του
ελέγχου ευδημονιστού, η οποια διασφαλίζει ότι μόνος
διεργασίες διαδέχονται σεν εισέρχονται ταυτόχρονα για κρίσιμα
χρήστα τους

Ο παραπάνω αλγόριθμος ενσημενώνει πρι
τις διεργασίες για να διαχειριστεί ταυτόχρονα οι δύο διεργασίες
αλλά αν την ίδια την ίδια ταυτόχρονα για εκείδεσην οι δύο
διεργασίες θα γίνουν οι περισσότερες plusing και minusing
true και false ή θα επελεγεται ταυτόχρονα σε αυτή την
λεπτίσηση. Ωστόσο, δεν εγασφαλίζεται η διάρκη των
αποταμιας αποκλειστος. Μια λύση θα προσθέσει να
ειναι περι Ajgn ένα άλλη για.

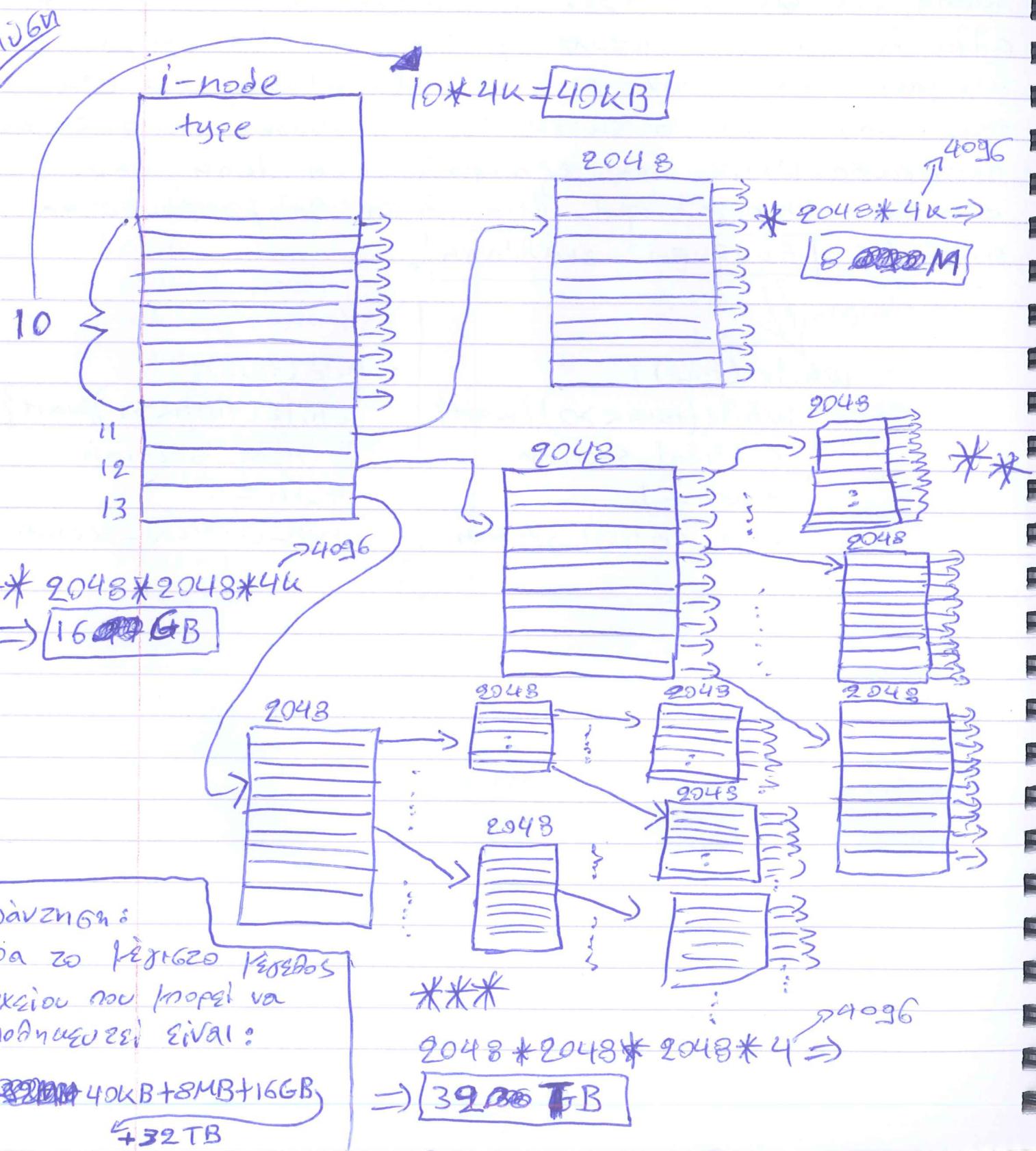
while(true){
 while(turn<>0){wait}
 critical_section
 turn=1
 non_critical_section
}

while(true){
 while(turn<>1){wait}
 critical_section
 turn=0
 non_critical_section
}

1,000,000.000.000

Dita 3

Εξίσω από την αρχική μέθοδο για την παραγωγή αρχείων, υπάρχουν δύο άλλες στοιχεία που πρέπει να διατηρηθούν για την επιτυχή λειτουργία της εφαρμογής. Το πρώτο είναι η μέγιστη μέγεθος αρχείου που μπορεί να διατηρηθεί στην μνήμη της συσκευής, η οποία ορίζεται στην εγκατάσταση της εφαρμογής και έχει την τιμή 4KB. Το δεύτερο στοιχείο που πρέπει να διατηρηθεί είναι η μέγιστη μέγεθος αρχείου που μπορεί να διατηρηθεί στην δισκογραφία, η οποία ορίζεται στην εγκατάσταση της εφαρμογής και έχει την τιμή 2048.



Δίκτα 4

Σε ίνα σύγχρονα πολυ χρησιμοποιούμενα ανοικτά αστικά διεγόδου ναι επιδιόρθωσης (λαβάψη) υπάρχουν δύο (1) printer και δύο (2) drives. Σε κάποια χρονική σειρά των λειτουργιών των λαβάψην τρεις διεργασίες για εκτέλεση Οι τρεις αυτές διεργασίες για την ~~εκτέλεση~~ περιφερειακής της για παραγάνει την σειρά:

1. Process 1 printer

2. Process 2 disk

3. Process 3 disk

4. Process 2 disk

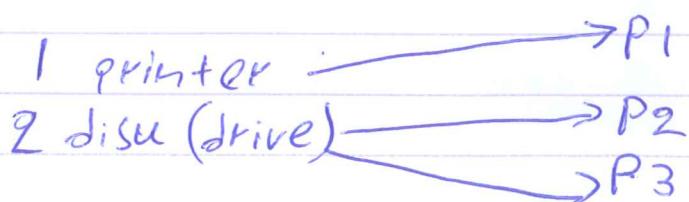
5. Process 3 printer

6. Process 1 disk

a) Εγενέρασε κάθε αιώνα ναι αναφέρεται αυτήν την περίοδον σε κάποιο σημείο νοτίζεται σειρά υπάρχει αστικός.

b) Αν το σύγχρονα χρησιμοποιούμενα spooling για τον printer, σεια ακέρως επέτεν αιώνων (διλαδικού αιώνου - Νο 7) θα προκαλέσει αδιέξοδο

ΑΙΩΝ



Εφόσον, έχουμε δύο printer → P1 διεργασία, disk στην P2 διεργασία και disk → P3 διεργασία οι πόροι που έχουν φεύγει είναι: printer (1/1) και disk (2/2) από έχουν δεσμευτεί στην process 2 στο επόμενο λεπτό συντάσσει διοικητικής σημασίας, η process 3 για printer και φεύγει δεκτή επιτίμηση, η process 1 για disk και φεύγει δεκτή επιτίμηση.

σεν προσούν να μαδύψουν
γιατί οι οποιοι ήσαν γνωστοί ~~κατά~~ στις διεργασίες
ενεργή γινεται έτσις μιαδος και η ημέρα διεργασία για
τόπους αντί όχι από την αλλη.

?) Η αίτηση Νο7 θα είναι η εξής:

7. Process 1 δισυ (Η διεργασία για την πρόβλημα
αντί των 10 δισυ, αλλά ο
δισυς αποχωρίζεται από τη
διεργασία γιατί δεν είναι
και δεν έχει ανέλευθερωδή^{ανάτα}. Και προκαλείται
αδιέξοδος.)

Δίκτα 5

Ένα δειγματικό σύγχρονα χρησιτούσεις virtual memory για demand paging. Αναδριθείτε σελίδα δεν έρχεται σενναν πριν αν δεν γνωθεί. Έτσι, αρχικά δεν φορτίνεται κατία σελίδα σενναν πριν και έχουμε page fault αυτό και τε σενναν αρώνα διεύθυνσης για την ηπιέζεται. Το σύγχρονα χρησιτούσεις ζων αλγόριθμο FIFO για την αντικατάσταση των σελίδων. Άλλες τρεις παραπάνω page trace (σειρά αντικατεβάσεων σελίδων):

1, 2, 3, 4, 1, 5, 1, 3, 2, 4, 3, 4, 5, 1

Διερεύεται αριθμός σφαλμάτων σελίδων (μεριώντας και τα αρχικά) καθώς και τα σελίδαι περιεχόμενα της πριν από αυτό της παρακάτω περιοχής αν:

- Διερεύεται σενναν διεργασία 3 ηλιστικά νέπια πριν.
- Διερεύεται σενναν διεργασία 4 ηλιστικά νέπια πριν.
- Διερεύεται σενναν διεργασία 3 ηλιστικά νέπια πριν και χρησιτούσειται ο αλγόριθμος LRU.

a) (3 ηλιστικά) ↓

1	2	3	4	1	5	1	3	2	4	3	4	5	1
①	1	1	④	4	4	4	③	3	3	3	3	⑤	5
②	2	2	①	1	1	1	②	2	2	2	2	2	①
③	3	3	⑤	5	5	5	④	4	4	4	4	4	
*	*	*	*	*	*	*	*	*	*	*	*	*	*

Έχουμε 11 σφάλματα και 3 hit.

(4 ηλιστικά) ↓

1)	2	3	4	1	5	1	3	2	4	3	4	5	1
①	1	1	1	1	⑤	5	5	5	5	5	④	4	4
②	2	2	2	2	①	1	1	1	1	1	③	5	
③	3	3	3	3	②	2	2	2	2	2	①		
④	4	4	4	4	4	4	4	③	3	3	3		
*	*	*	*	v	*	*	v	*	*	*	*	*	*

Έχουμε 11 σφάλματα και 3 hit.

«Хаки» 11 690 ₽ за м² 3 листа

Агуун 6н (рэ Мурэн)

Εε πλα τυπου 256MB, να δείχνεται ότι ζρόπο ευχώρων
τυπους ήταν τη δύσκολη συντήρηση φίλων και η συντήρηση
εναργεών συντήρησης και αποθήκευσης τυπους, για τα οποία ευθύνεται:

AP154 AIEPOARIAS A 4MB ✓
AP154 AIEPOARIAS B 8MB ✓

Άφιξη Διεργασίας B 28 MB ✓

A sign ΔΙΕΡΓΑΣΙΑΣ Γ, 96 MB ✓

Tēdos Διεργασίας f ✓

Αφίγη Διεργασίας Δ 128MB ✓

Todos Acopagias A ✓

Αφίγη Διεργασίας Ε 58MB ✓,

Αφίξη Διεργαστής 2 48MB

Οδυσσείων Αρχαρίων (Β.Α.Ζ.Ε) τε και σερβ. ✓

* HZ māivēl gē avapovī hēxpi
va un p̄zēl xii pos.

256			
A	48	16	32
A	48	16	B
A	48	16	B
A	48	16	B
A	48	16	B
A	48	16	B
4	48	16	B
32		B	
32		B	
64		E	
2		E	
2		E	128
128			
64		E	128
64		64	128
256			

P1, P2

C1, C2, C3
↑

Aγοράνη τε 2 producer, 3 consumer

Να σπάτετε τια ενίσχυση του προβλήματος παραγωγής - παραλαβής, για ονοματούχους 2 παραγωγούς και 3 παραλαβές και η γειπά ευχέρειας είναι: P1, P2 και C1, C2, C3 παραγωγής παραλαβών. Μόλις παραλαβώνται όλες οι παραγωγές, αρχίζει η παραγωγή της P1.

Άριθμοι

Semaphore: P1=3, P2=0, C1=0, C2=0, C3=0, mutex=1

P1	P2	C1	C2	C3
down(p1)	down(p2)	down(c1)	down(c2)	down(c3)
down(p1)	add_item	down(mutex)	down(mutex)	down(mutex)
down(p1)	up(c1)	consume_item	consume_item	consume_item
add_item	up(c2)	up(mutex)	up(mutex)	up(mutex)
up(p2)	up(c3)	up(p1)	up(p1)	up(p1)

* Αν έπειτα να παραλαβώνται όλες οι 3 consumer

Semaphore: P1=3, P2=0, C1=0, C2=0, C3=0, mutex=1

P1	P2	C1	C2	C3
down(p1)	down(p2)	down(c1)	down(c2)	down(c3)
down	add_item	down(c1)	down(c2)	down(c3)
down	up(c1)	down(mutex)	down(mutex)	down(mutex)
add_item	up(c2)	consume_item	consume_item	consume_item
up(p2)	up(c3)	up(mutex)	up(mutex)	up(mutex)
up(p2)		up(p1)	up(p1)	up(p1)

* Το κρίσιμο Τύπω οργάνωνται τε το mutex

όπως ΜΑΡΑΤΑΝΩ ↑

C_1, C_2 ,
↑

1, 3 CONCURENCE

για δυαδικής παραγοφής της φυτής περιέχει τα
κεράνη μενού (ελεύθερων) γραμμάτων και
προβλήματος
ουν 2 παραγμ., 5KB, 18KB, 16KB, 6KB, 7KB, 10KB, 13KB)

είδες είναι
σήσ ουαραδώ
o PI.

2) BKB 3) 9KB
προσομοιώνεται στη αλγόριθμοι :

B) Best-fit γ) Worst-fit δ) Next-fit

$=0, C_2=0, C_3=0$

C_2	BF	WF	NF
down(c_2)	down		
down(mutex)	down		
consume_item	con	12	12
up(mutex)	u	6	6
up(pi)	u	6	
		9	9
	12	9	

C_2
down (c_2)
down (c_2)
down(mutex)
consume_item
up(mutex)
up(pi)

ΖΑΖΕΥΣΕΙ ΤΕ
ΠΑΡΑΤΑΝΟ

First-fit: Ραιρνεται στην πρώτη διάθεση που έχει απορία και στην οποία η πρώτη διάθεση που έχει απορία προσέρχεται.

Best-fit: Ραιρνεται στην πρώτη διάθεση που έχει απορία και βασικώς στην πρώτη διάθεση που έχει απορία προσέρχεται.

Worst-fit: Ραιρνεται στην πρώτη διάθεση που έχει απορία και βασικώς στην πρώτη διάθεση που έχει απορία προσέρχεται.

Next-fit: Ραιρνεται στην πρώτη διάθεση που έχει απορία και στην πρώτη διάθεση που έχει απορία προσέρχεται. Η επόμενη διάθεση προσέρχεται να φέρει απορία και η επόμενη διάθεση προσέρχεται να φέρει απορία.

Ενα σύγχρονα διατάξιμα μαζικούς υπολογιστές περιέχει τα παρακάτω τεχνητά κενά (ελεύθερων) γραμμάτων και

βελή: (8KB, 5KB, 18KB, 16KB, 6KB, 7KB, 10KB, 13KB)

Αλγόριθμοι:

1) First-fit 2) Best-fit 3) Worst-fit

• και χρησιμοποιούνται οι αλγόριθμοι:

a) First-fit b) Best-fit c) Worst-fit d) NextFit

Νύση

	FF	BF	WF	NF
8	6			
5				
18	12		12	12
16	9		6	6
6		6		
7				
10		9	9	
13		12	9	

First-Fit: Ραιρνει τις αίσχησις πα-πα γε μάθε αλγόριθμο και σαν ν πρώτη που θα κυρίσσει πραγματεύεται.

Best-Fit: Ραιρνει τις αίσχησις πα-πα και βάσω την μάθε αίσχηση ευει που κυριάτελη καλύτερα.

Worst-fit: Ραιρνει τις αίσχησις πα-πα και βάσω την μάθε αίσχηση γραφειού που κυριάτελη καλύτερα.

Next-Fit: Ραιρνει την μάθε αίσχηση και σ' αυτού κυριάτελη την βάση. Η επόκεινη αίσχηση γενινάει και φάκνει θέση που θα έχει αυτό εκεί που την προηγουμένων. (Αν φάκνει σε σέλιδα γενινάει από την αρχή)