

Καταναμημένα Συστήματα
Καστίδου Γεωργία Α.Μ. 49
Μαρία Παπαφώτη Α.Μ. 45
7η Ομάδα Ασκήσεων

1. Show that if events e and e' are concurrent then neither $V(e) < V(e')$ nor $V(e') < V(e)$. Hence that if $V(e) < V(e')$ then $e \rightarrow e'$

α) Αρκεί να δείξουμε πως αν $V(e) \leq V(e')$ ή $V(e') \leq V(e)$ ισχύει τότε δεν είναι ταυτόχρονα τα e, e'

Έστω ότι ισχύει $V(e) \leq V(e') \Leftrightarrow V_e(e) \leq V_{e'}(e) \quad (1) \quad \&\& \quad V_{e'}(e') \leq V_e(e') \quad (2)$

x	e	e'
$V_e(x)$	a	b

x	e	e'
$V_{e'}(x)$	c	d

(1) $\Leftrightarrow a \leq c \quad (3)$

(2) $\Leftrightarrow b \leq d \quad (5)$

Όμως αν $a < c$ δεν γίνεται διότι ένα event πρώτα αυξάνει το στην διεργασία όπου λαμβάνει χώρα και μετά μεταδίδεται στις άλλες διεργασίες, με άλλα λόγια $a \geq c$ (4).

Άρα από (3), (4) $a = c$. Όμως $a = c$ σημαίνει πως το event e έγινε πριν την e' και συνεπώς δεν είναι ταυτόχρονα. Λόγω του γεγονότος αυτού $b < d$. Άρα τελικά ισχύει ότι $V(e) < V(e')$.

Αντίστοιχα πράττουμε και στην περίπτωση που δεχτούμε ότι ίσχυε η $V(e') < V(e)$

2. Problems 9,10,12,27

9. In Fig 6-7 is 000000 a legal output for a distributed shared memory that is only FIFO consistent?

Ναι διότι στην περίπτωση που έχουμε FIFO consistency οι εγγραφές από διαφορετικές διεργασίες θεωρούνται ταυτόχρονες, συνεπώς θα μπορούσαμε να έχουμε την σειρά εκτέλεσης όπως φαίνεται σε καθεμιά από τις 3 διεργασίες (σχήμα 1).

x=1	y=1	z=1
print(y,z)	print(x,z)	print(x,y)
y=1	x=1	y=1
print(x,z)	print(y,z)	print(x,z)
z=1	z=1	x=1
print(x,y)	print(x,y)	print(y,z)
Prints: 00	Prints: 00	Prints: 00
(a)	(b)	(c)

Σχήμα 1:

10. In Fig 6-8 is 001110 a legal output for a sequentially consistent memory? Explain your answer.

Θεωρώντας ότι το **00 11 10** είναι signature. Το **00 11 10** είναι νόμιμη signature του παραδείγματος που παρουσιάζεται στην εικόνα 6.8 του βιβλίου διότι αν δούμε αναλυτικά τη σημασία των bits (σχήμα 2) βλέπουμε πως κανένας από τους «κανόνες» δεν έρχεται σε αντιδιαστολή με κάποιον άλλο «κανόνα».

Οι κανόνες είναι που προκύπτουν από την έξοδο 001110 είναι:
1° (*) bit 0 ⇔ P ₁ ξεκινάει την εκτέλεση πριν την P ₂
2° bit 0 ⇔ P ₁ ξεκινάει την εκτέλεση πριν την P ₃
3° bit 1 ⇔ P ₂ ξεκινάει την εκτέλεση μετά την εκκίνηση της εκτέλεσης της P ₁
4° bit 1 ⇔ P ₂ ξεκινάει την εκτέλεση μετά την εκκίνηση της εκτέλεσης της P ₃
5° bit 1 ⇔ P ₃ ξεκινάει την εκτέλεση μετά την εκκίνηση της εκτέλεσης της P ₁
6° bit 0 ⇔ P ₃ ξεκινάει την εκτέλεση πριν την P ₂

Σχήμα 2: Κανόνες εκτέλεσης διεργασιών

Επίσης η πληροφορία που παίρνουμε είναι πως η σειρά με την οποία εκκινούν την εκτέλεση τους οι διεργασίες είναι P₁, P₃, P₂.

- Η αρίθμηση των bits γίνεται από αριστερά προς τα δεξιά.

12. In Fig. 6-13, a sequentially consistent memory allows six possible statement interleavings. List them all.

Process P1	Process P2
X=1;	Y=1;
If(Y= =0) kill P2	If(X= =0) kill P1

Οι δυνατές εκτελέσεις που επιτρέπει το μοντέλο της ακολουθιακής συνέπειας είναι:

P1: X=1;

P1: If(Y= =0) kill P2

P1: X=1;

P2: Y=1;

P1: If (Y= =0) kill P2

P2: If (X= =0) kill P1

P1: X=1;

P2: Y=1;

P2: If (X= =0) kill P1

P1: If (Y= =0) kill P2

P2: Y=1;

P2: If (X= =0) kill P1

P1: Y=1;

P2: X=1;

P1: If (Y= =0) kill P2

P2: If (X= =0) kill P1

P1: Y=1;

P2: X=1;

P2: If (X= =0) kill P1

P1: If (Y= =0) kill P2

27. Consider causally-consistent lazy replication. When exactly can an operation be removed from a write queue?

Ο τρόπος με τον οποίο μπορεί να διαγραφεί ένα write γεγονός από την ουρά είναι κάθε διεργασία που εκτελεί ένα write να στέλνει ένα acknowledgement σε όλες τις υπόλοιπες και στον εαυτό της. Όταν κάθε διεργασία έχει πάρει acknowledgement από όλες τότε μπορεί να διαγράψει το write από την ουρά.

3. Does sequential consistency implies monotonic reads, monotonic writes, read your writes, writes follows reads?