Πετράκης Γιαννης Α.Μ. 46

Άσκηση 1

Θα δείξουμε πρώτα ότι $V_j[i] \leq V_i[i]$ (1)
Αυτό όμως είναι προφανές αφού για να ενημερωθεί η διεργασία *j* για τον αριθμό των γεγονότων που έχουν γίνει στη διεργασία *i* πρέπει η τελευταία να της στείλει μήνυμα οπότε η *j* κάνει το $V_j[i]$ ίσο με το $V_i[i]$. Δηλαδή μόνο με αυτό τον τρόπο μπορεί να αλλάξει η τιμή $V_j[i]$. Άρα ισχύει ότι $V_j[i] \leq V_i[i]$.
Θα δείξουμε τώρα ότι αν τα γεγονότα e και e' είναι ταυτόχρονα τότε ούτε
$V(e) \leq V(e')$ ούτε $V(e) \geq V(e')$ (2)
Έστω ότι τα γεγονότα αυτά ανήκουν στις διεργασίες p_i και p_j αντίστοιχα. Αφού είναι ταυτόχρονα σημαίνει ότι δεν έχει γίνει καμία άμεση ή έμμεση αποστολή μηνύματος από το ένα στο άλλο. Άρα σύμφωνα με την (1) ισχύουν οι σχέσεις:
$V_j[i] < V_i[i]$ και $V_i[j] < V_j[j]$ (δεν βάζουμε την ισότητα αφού δεν έχει γίνει καμιά επικοινωνία μεταξύ των δύο διεργασιών ώστε να ενημερωθούν οι τιμές των $V_i[j]$ και $V_j[i]$). Άρα ισχύει η (2).
Επίσης είναι προφανές σύμφωνα με τα παραπάνω ότι αν $V(e) < V(e')$ τότε $e \rightarrow e'$, αφού αν τα e και e' είναι ταυτόχρονα τότε ούτε $V(e) \leq V(e')$ ούτε $V(e) \geq V(e')$.

Άσκηση 2

Πρόβλημα 9

Αυτή η έξοδος προκύπτει μόνο όταν εκτελούνται πρώρα οι εντολές print και μετά οι αναθέσεις. Αυτό μπορεί να συμβεί μόνο στη FIFO και στην Casual consistency, αφού σε αυτές τις δύο περιπτώσεις υπάρχει ο περιορισμός να είναι ορατές με τη σειρά που γίνονται μόνο οι εντολές write κάθε διεργασίας. Επομένως δεν παραβιάζει τον κανόνα να εκτελείται πρώτα το print (εντολή ανάγνωσης) και μετά η ανάθεση (εντολή γραψίματος)

Πρόβλημα 10

Τα πρώτα δύο bits, 00, σημαίνουν ότι και οι δύο εντολές της P1 εκτελούνται πριν αρχίσει η εκτέλεση των P2 και P3. Τα επόμενα bits, 11,σημαίνουν ότι πρέπει η εντολή print(x,z) της P2 να εκτελεστεί μετά την πρώτη εντολή (z=1) της P3. Άρα η πρώτη εντολή (y=1) της P2 πρέπει να εκτελεστεί πριν την print(x,z) αφού είναι εντολές της ίδιας διεργασίας και πρέπει να είναι ορατές με τη σειρά που έγιναν. Τα δύο τελευταία bit, 10, σημαίνουν ότι η εντολή print(x,y) της P3 πρέπει να εκτελεστεί πριν από την y=1 της P2. Επομένως είναι αποδεκτή η έξοδος. Με βάση τα παραπάνω μια έγκυρη ακολουθία εκτέλεσης είναι:

```
x=1;
print(y,z);
z=1;
print(x,y) ;
y=1 ;
```

print(x,z);

Πρόβλημα 12

1)
x=1;
If(y==0) kill(P2);
y=1;
If(x==0)kill(P1);

2)
x=1;
y=1;
If(y==0) kill(P2);
If(x==0)kill(P1);

3)
x=1;
y=1;
If(x==0)kill(P1);
If(y==0) kill(P2);

4)
y=1;
If(x==0)kill(P1);
x=1;
If(y==0) kill(P2);

5)
y=1;
x=1;
If(x==0)kill(P1);
If(y==0) kill(P2);

6)
y=1;
x=1;
If(y==0) kill(P2);
If(x==0)kill(P1);

Πρόβλημα 27

Όταν μια εντολή της write queue στέλνεται σε κάποιο από τα αντίγραφα της αποθήκης δεδομένων ζητείται επιβεβαίωση από τον παραλήπτη ότι πήρε το μήνυμα και ενημέρωσε το αντίγραφό του. Σε κάθε εντολή της write queue αποθηκεύονται οι κόμβοι από τους οποίους έχει έρθει επιβεβαίωση ότι έχει εκτελεστή η συγκεκριμένη

εντολή. Όταν έρθει επιβεβαίωση από όλους τους κόμβους σημαίνει ότι όλα τα αντίγραφα έχουν ενημερωθεί (έχουν εκτελέσει την εντολή αυτή) και επομένως μπορεί να διαγραφεί από την ουρά.

Άσκηση 3

Ας θεωρήσουμε το ακόλουθο σχήμα το οποίο παρουσιάζει sequential consistency:

P1: W(x) a
P2: W(x) b
P3: R(x) b R(x) a W(x) c
P4: R(x) b R(x) a

Monotonic Reads: Η sequential consistency δεν συνεπάγεται σε monotonic reads αφού όπως φαίνεται από το παραπάνω σχήμα η διεργασία 3 διαβάζει πρώτα την τιμή b του στοιχείου x και μετά την τιμή a (η οποία είναι παλαιότερη τιμή)

Monotonic Writes: Ισχύει αφού στο sequential consistency η σειρά με την οποία εμφανίζονται τα writes στην ίδια διεργασία είναι αυτή με την οποία έγιναν στην πραγματικότητα, γιατί όταν μια εντολή write εκτελείται σε μια διεργασία το αποτέλεσμα του γίνεται αμέσως γνωστό σε αυτήν. Άρα πάντα το write θα γίνονται αφού έχουν ολοκληρωθεί τα προηγούμενα writes και είναι γνωστά τα αποτελέσματά τους.

Read Your Writes: Από τη στιγμή που το αποτέλεσμα μιας εντολής write γίνεται αμέσως γνωστό σε μια διεργασία, αυτό σημαίνει ότι κάθε επακόλουθο read από την ίδια διεργασία θα διαβάσει την τελευταία τιμή που γράφτηκε. Άρα η sequential consistency συνεπάγεται σε *Read Your Writes.*

Writes Follow Reads: Δεν ισχύει γιατί από το παραπάνω σχήμα φαίνεται ότι η διεργασία 3 γράφει στο δεδομένο x του οποίου οποίου η τελευταία τιμή που διαβάστηκε δεν είναι η ποιο πρόσφατη αφού προηγουμένως έχει διαβαστεί η τιμή b.