Πετράκης Γιάννης Α.Μ. 46

Άσκηση 1

Ο χρόνος δεν πρέπει ποτέ να γυρνάει πίσω. Αν γυρίσουμε το ρολόι στη σωστή τιμή μπορεί να προκαλέσει σοβαρά προβλήματα, όπως το να έχει ένα καταληκτικό αρχείο, το οποίο είναι προϊόν μεταγλώττισης που έγινε αμέσως μετά την αλλαγή στο ρολόι, χρονική ένδειξη μικρότερη από το αρχείο πρωτογενούς κώδικα, το οποίο τροποποιήθηκε αμέσως πριν από την αλλαγή στο ρολόι.
Ας υποθέσουμε ότι ο χρονομετρητής είναι ρυθμισμένος να προκαλεί 100 διακοπές το δευτερόλεπτο. Κανονικά κάθε διακοπή θα προσέθετε 10 msec στην ώρα. Για να γίνει ρύθμιση του ρολογιού μέσα σε 8 sec θα πρέπει η ρουτίνα εξυπηρέτησης διακοπής να προσθέτει 5 msec αντί για 10 για τα επόμενα 8 sec. Έτσι μετά από 8 sec το ρολόι θα έχει τη σωστή τιμή.

Άσκηση 2

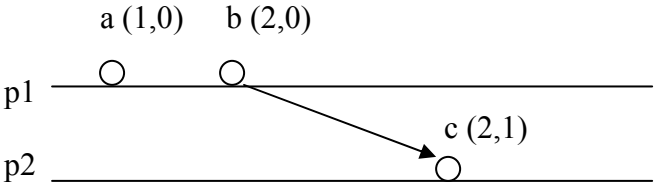
Θα έπρεπε να χρησιμοποιήσει την τρίτη τιμή γιατί το Round-trip time είναι μικρότερο και επομένως η απόκλιση λόγω του χρόνου μεταβίβασης των μηνυμάτων θα είναι πιθανότατα μικρότερη.
Η τιμή που θα πρέπει να πάρει το ρολόι είναι: 10:54:28.352
Στην περίπτωση που ο χρόνος αποστολής και παραλαβής ενός μηνύματος είναι τουλάχιστον 8msec η απάντηση δεν αλλάζει. Σε αυτή την περίπτωση το Round-trip time θα είναι τουλάχιστον 16msec και επομένως το πολύ 20 -16 = 4 msec θα είναι η απόκλιση(είναι η χειρότερη περίπτωση). Αυτό μπορεί να συμβεί αν ο χρόνος αποστολής του μηνύματος στον server είναι 8msec και ο χρόνος από τη στιγμή που ο server έστειλε την απάντηση μέχρι να την πάρει ο client είναι 12 msec.

Άσκηση 3

Έστω *l* το μήκος της αλυσίδας των μηνυμάτων.
Για *l*=0 (δηλαδή δεν συνδέονται με μήνυμα οι e και e') ισχύει $e \rightarrow e' \Rightarrow L(e)<L(e')$
Έστω ότι ισχύει για *l*=k, δηλαδή $e \rightarrow e_2 \rightarrow \ldots \rightarrow e_k \rightarrow e' \Rightarrow L(e) \leq L(e')$ (1)
Θα δείξουμε ότι ισχύει για αλυσίδα μήκους k+1.
Από τη στιγμή που ισχύει για k θα έχουμε $e \rightarrow e_2 \rightarrow \ldots \rightarrow e_k \rightarrow e_{k+1} \Rightarrow L(e) \leq L(e_{k+1})$
Στη συνέχεια το γεγονός e_{k+1} συνδέεται με μήνυμα με το γεγονός e'.
Αφού όμως $e_{k+1} \rightarrow e'$ θα ισχύει ότι $L(e_{k+1}) < L(e')$. Άρα ισχύει η σχέση (1).

Άσκηση 4

(a) Η σχέση αυτή δεν ισχύει πάντα. Ας θεωρήσουμε το ακόλουθο παράδειγμα με δύο διεργασίες:



Παρατηρούμε ότι το $V_2[2]=1<V_2[1]=2$. Επομένως δεν ισχύει.

(b) Δεν ισχύει το (α) …

Άσκηση 5

Είναι απαραίτητη η επιβεβαίωση σε κάθε μήνυμα. Το μήνυμα γίνεται multicast σε όλες της διεργασίες. Κάθε μια από αυτές, όταν το παραλάβει κάνει multicast την επιβεβαίωση σε όλες τις άλλες. Μια διεργασία παραδίδει στην εφαρμογή το μήνυμα το οποίο βρίσκεται στην κορυφή της ουράς (έχει το μικρότερο timestamp) και το οποίο έχει γίνει acknowledged από όλες τις άλλες διεργασίες. Αυτό σημαίνει (το ότι δηλαδή έχει γίνει acknowledged από όλες) ότι και όλες οι άλλες διεργασίες έχουν το μήνυμα στην κορυφή της ουράς και επομένως όλες οι λειτουργίες θα γίνουν με την ίδια σειρά σε όλες τις διεργασίες.

Άσκηση 6

Ένας τρόπος να γίνει αυτό είναι ο εξής:
Ο κόμβος που ξεκινάει τη διαδικασία σύμφωνα με τον αλγόριθμο μπαίνει πρώτος στη λίστα του μηνύματος. Επομένως είναι εύκολο να βρεθεί ποιος κόμβος ξεκίνησε τη διαδικασία εκλογής. Αν δυο (ή περισσότεροι) κόμβοι ξεκινήσουν παράλληλα μια διαδικασία εκλογής τότε μόλις κάποιος από αυτούς λάβει κάποιο μήνυμα από διαδικασία εκλογής που ξεκίνησε από άλλο κόμβο συγκρίνει το id του με το id του άλλου κόμβου. Αν ο άλλος κόμβος έχει μικρότερο id τότε δεν μεταβιβάζει το μήνυμα στον επόμενο. Έτσι τελικά μόνο το μήνυμα εκλογής που ξεκίνησε από τον κόμβο με το μεγαλύτερο id θα συνεχίσει.

Άσκηση 7

Το maximum throughput πετυχαίνεται όταν ο αριθμός των μηνυμάτων που πρέπει να σταλούν για να μπει μια διεργασία στην κρίσιμη περιοχή είναι ο ελάχιστος. Αυτό συμβαίνει στον κεντρικοποιημένο αλγόριθμο όπου ο αριθμός των μηνυμάτων είναι δύο. Έτσι αν ο μέσος χρόνος αποστολής και παραλαβής ενός μηνύματος είναι t seckαι

ο χρόνος που σπαταλάει κάθε διεργασία στο κρίσιμο τμήμα είναι κατά μέσο όρο d sec τότε το μέγιστο throughput είναι: 1 διεργασία / (2t+d) secs

Άσκηση 8

Στις κατανεμημένες δοσοληψίες δεν υπάρχει ένα μόνο scheduler για όλα τα δεδομένα. Υπάρχει ένας scheduler σε κάθε κόμβο ο οποίος χειρίζεται τα δεδομένα του κόμβου αυτού. Επομένως ο ορισμός (σελίδα 283 του βιβλίου) μπορεί να γίνει:
Στο 2PL οι schedulers των διαφορετικών κόμβων στους οποίους υπάρχουν τα δεδομένα της δοσοληψίας πρώτα κλειδώνουν τα τοπικά τους δεδομένα τα οποία θα χρειαστεί η δοσοληψία κατά τη διάρκεια της growing phase και μετά τα αποδεσμεύουν κατά τη διάρκεια της shrinking phase.