

Καταναμημένα συστήματα

Παπαφώτη Μαρία Α.Μ: 45

19/5/03

1. A clock is reading 10:27:54.0 (hr:min:sec) when it is discovered to be 4 sec fast. Explain why it is undesirable to set it back to the right time at that point and show (numerically) how it should be adjusted so as to be correct after 8secs have elapsed

Δεν είναι καλή τακτική να γυρίζουμε πίσω το ρολόι ενός συστήματος μόλις συνειδητοποιήσουμε ότι πηγαίνει πιο μπροστά από το κανονικό κι αυτό γιατί αυτή η αλλαγή είναι πιθανό να αλλάξει τη σειρά των γεγονότων στο σύστημα. Έτσι αν ένα γεγονός συνέβη πριν ακριβώς την αλλαγή του ρολογιού κι ένα συμβεί αμέσως μετά το δεύτερο γεγονός θα φαίνεται στο σύστημα να έχει συμβεί πρώτο, γεγονός που μπορεί να επηρεάσει τη λειτουργία του συστήματος αν η σειρά των προαναφερθέντων γεγονότων παίζει κάποιο ρόλο. Για να έχω σωστή ώρα στο σύστημα μετά από 8 sec θα πρέπει να αυξάνω το ρολόι του, με το μισό ρυθμό από ότι κανονικά, με άλλα λόγια αν κάθε x interrupts πρόσθεταν ένα second στο ρολόι τώρα θα πρέπει να προσθέτουν μισό.

2. A client attempts to synchronize with a timeserver. It records the round trip times and timestamps returned by the server in the table below.

Round-trip (ms)	Time (hr:min:sec)
22	10:54:23.674
25	10:54:25.450
20	10:54:28.342

Which of these times should it use to set its clock? To what time should it set it? Estimate the accuracy of the setting with respect to the server's clock. If it is known that the time between sending and receiving a message in this system is at least 8ms, do your answers change?

Θα χρησιμοποιούνταν ο μεγαλύτερος από τους χρόνους (10:54:28.342) για να τεθεί τιμή στο ρολόι. Η νέα τιμή του ρολογιού θα ήταν η τιμή του τελευταίου ρολογιού όπως είπαμε συν το μισό του μέσης διάρκειας του round trip, δηλαδή θα ήταν: $10:54:28.(342+(20+25+22)/3/2)= 10:54:28.(342+11,16) = 10:54:28.353$. Η ακρίβεια είναι της τάξης των δευτερολέπτων αφού το ρολόι θα τεθεί σε τιμή κατά 16εκατοστά του δευτερολέπτου περίπου μικρότερη από τον timeserver. Στη συγκεκριμένη περίπτωση δε θα μας επηρέαζε γιατί ο μέσος χρόνος round trip είναι ίσως με 22,3ms το οποίο είναι πολύ μεγαλύτερο από το 8ms.

3. By observing a chain of zero or more messages connecting events e and e' and using induction, show that $e \rightarrow e' \Rightarrow L(e) \leq L(e')$

Γνωρίζουμε ότι:

1. $\forall \alpha \rightarrow \beta \Rightarrow L(\alpha) \leq L(\beta)$
2. $\forall \alpha$ γεγονός αποστολής μηνύματος και β γεγονός λήψης μηνύματος τότε $L(\alpha) \leq L(\beta)$
3. Για όλα τα ξεχωριστά γεγονότα α και β $L(\alpha) \neq L(\beta)$

Για μια ακολουθία οσονδήποτε γεγονότων με 0 μηνύματα είμαστε πάντα στην ίδια διεργασία και ισχύει από το (1)

Για μια ακολουθία οσονδήποτε γεγονότων με 1 μήνυμα μόνο (2 από τα γεγονότα είναι: γεγονός αποστολής μηνύματος και γεγονός λήψης μηνύματος) ισχύει από το (2)

Υποθέτω ότι ισχύει για ακολουθίες οσονδήποτε γεγονότων με λιγότερα από n μηνύματα

Θα δείξω ότι ισχύει για ακολουθία με οσονδήποτε γεγονότων με n μηνύματα:

Μπορώ να σπάσω την ακολουθία αυτή των γεγονότων στο σημείο όπου έχω τη n -ιστή αποστολή μηνύματος οπότε θα έχω δυο υποακολουθίες όπου $e \rightarrow \alpha \rightarrow e'$ για τις οποίες ισχύει $L(e) \leq L(\alpha)$ και $L(\beta) \leq L(e')$ από υπόθεση. Από τον τρόπο που σπάσαμε την ακολουθία με βάση το (2) έχουμε $\alpha \rightarrow \beta$ και επειδή η σχέση «προηγείται» ορίζεται να είναι μεταβατική έχουμε $e \rightarrow \beta$ και $e \rightarrow e'$.

5. Problem 4 from the textbook

Είναι απαραίτητο να έχουμε μηνύματα επιβεβαίωσης σε κάθε μήνυμα αν θέλουμε το multicasting τα ίδια μηνύματα να μπου σε σειρά σε όλους τους κόμβους. Διαφορετικά αν μας αρκεί τα μηνύματα που θα φτάσουν στον κόμβο να είναι σε σειρά τότε δεν χρειαζόμαστε μηνύματα επιβεβαίωσης.

6. Problem 7 from the textbook

Όταν ένας κόμβος ξεκινάει μια διαδικασία εκλογής θα μπορούσε εκτός από το μήνυμα που στέλνει για τις εκλογές να κάνει broadcast και ένα άλλο μήνυμα σε όλους τους άλλους έτσι ώστε έχει ξεκινήσει κι άλλη ψηφοφορία να την διακόψει. Το κακό με αυτό τον αλγόριθμο είναι ότι μπορεί να διακοπεί μια ψηφοφορία λίγο πριν την ολοκλήρωση της απλώς διότι άρχισε μια άλλη και έτσι να πάνε χαμένα πολλά μηνύματα. Στην περίπτωση όμως που οι ψηφοφορίες αρχίσουν σχεδόν μαζί τότε έχουμε κέρδος από τον αλγόριθμο αυτό. Μια δεύτερη λύση είναι όταν ένας κόμβος προωθήσει ένα μήνυμα εκλογής να παίρνει τη πρωτοβουλία να προωθήσει ένα από τα δύο μηνύματα.

8. Extend the definition of two-phase locking to apply to distributed-transactions. Explain how this is ensured by distributed transactions using two-phase locking locally.

Στο κλείδωμα δύο φάσεων ο δρομολογητής πρώτα παίρνει όλα τα κλειδιά των δεδομένων που χρειάζονται σε μια πρώτη φάση που ονομάζεται φάση ανάπτυξης και ύστερα σε μια δεύτερη φάση την φάση της συρρίκνωσης τα αφήνει. Όταν έχουμε να κάνουμε με κατανεμημένες δοσοληψίες αρκεί κάθε δοσοληψία να κλειδώσει τοπικά το δεδομένο στη συνέχεια η συνέπεια των δεδομένων ολόκληρου του συστήματος θα εξασφαλιστεί από άλλους μηχανισμούς.