

Κατανεμημένα Συστήματα
Καστίδου Γεωργία Α.Μ. 49
6η Ομάδα Ασκήσεων

Άσκηση 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 8 secs have elapsed.

Λύση:

Δεν είναι επιθυμητό να ρυθμιστεί το ρολόι πίσω διότι π.χ. μπορεί να έχουμε κάνει make κάποια αρχεία στις 10:27:53.0, ρυθμίζοντας το πίσω στις 10:27:50.0 αν μετά από 2 second κάνουμε κάποια αλλαγή σε ένα .c από τα παραπάνω αρχεία το αρχείο θα έχει ημερομηνία τροποποίησης 10:27:52.0 η οποία όμως αν και έγινε μετά την εκτέλεση του make έχει παλαιότερη ημερομηνία.

Και γενικά αρχεία που τροποποιήθηκαν ή δημιουργήθηκαν στο χρονικό διάστημα 10:27:50.0 έως 10:27:54.0 αν ρυθμιστεί η ώρα πίσω στη σωστή τιμή μπορεί να φαίνονται ότι δημιουργήθηκαν αργότερα από ότι αρχεία που δημιουργήθηκαν μετά τη ρύθμιση της ώρας δημιουργώντας προβλήματα ασυνέπειας.

Η ιδέα για τη ρύθμιση του ρολογιού βασίζεται την αύξηση του χρόνου αντί για ένα δευτερόλεπτο σε 0.5. δηλαδή, οπότε μετά από 8 «κανονικά» δευτερόλεπτα το ρολόι θα έχει αυξηθεί κατά 4 δευτερόλεπτα συν όμως τα 4 δευτερόλεπτα που ήταν πια μπροστά το ρολόι τελικά θα δείχνει 10:28:02.0. Το σημαντικό είναι πως με τον παραπάνω αλγόριθμο ο ένδειξη του ρολογιού συνεχώς αυξάνεται με αποτέλεσμα να μην παρατηρούνται προβλήματα ασυνέπειας όπως με τον αλγόριθμο που προτείνεται στην εκφώνηση της άσκησης.

Άσκηση 2: A client attempts to synchronize with a time server. 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?

Λύση:

Ο χρόνος που πρέπει να χρησιμοποιηθεί για να ρυθμιστεί το ρολόι του server είναι ο τελευταίος και αυτό γιατί για να είναι συγχρονισμένο το ρολόι του πελάτη με το ρολόι του εξυπηρετή θα πρέπει ο πελάτης να πάρει την τελευταία ένδειξη (πιο πρόσφατη) του ρολογιού του εξυπηρετή δεδομένου του γεγονότος ότι το ρολόι του εξυπηρετή δεν γυρίζει πίσω (το γεγονός αυτό είναι δεδομένο).

Ο χρόνος με τον οποίο πρέπει να ρυθμιστεί το ρολόι είναι 10:54.28.352 και αυτό προκύπτει από:

Χρονική Ένδειξη Πελάτη=Χρονική Ένδειξη Ρολογιού εξυπηρετή + Round-Trip/2 = 10:54:28.342 “+” 0.010 = 10:54.28.352

Εάν γνωρίζουμε πως ο χρόνος για την αποστολή και λήψη ενός μηνύματος στο παραπάνω σύστημα είναι 8ms δεν θα αλλάξει η απάντηση διότι ο χρόνος αυτός υπολογίζεται ήδη μέσα στο χρόνο που απαιτείται για να γίνει ένα roundtrip.

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

Λύση:

Θα προσπαθήσουμε να επιλύσουμε το πρόβλημα με την μέθοδο της αναγωγής.

Έστω ότι υπάρχει ένα μήνυμα που συνδέει τα γεγονότα e και e' και $e \rightarrow e'$. Αυτό συνεπάγεται ότι η e στέλνει κάποιο μήνυμα στην e' . Σύμφωνα με τον αλγόριθμο του Lamport η e θα στείλει και το χρονόσημο της $L(e)$ μαζί με το μήνυμα στη συνέχεια η e' όταν λάβει το μήνυμα θα θέσει το χρονόσημο της $L(e') = \max\{L(e)+1, L(e')\}$ οπότε $L(e) < L(e')$.

Έστω τώρα ότι υπάρχουν n μηνύματα που συνδέουν τα γεγονότα e και e' και $e \rightarrow e'$. Τότε $L(e) < L(e')$.

Θα δείξουμε ότι αυτό ισχύει και αν συνδέουν $n+1$. Στο n -οστό γεγονός όπως δείξαμε θα ισχύει $L(e) < L(e')$.

Στο $N+1$ γεγονός η e' λαμβάνει ξανά μήνυμα από την e , στην συνέχεια η e' θέτει το timestamp της ίσο με $L(e') = \max\{L(e)+1, L(e')\}$ οπότε $L(e') < L(e)$.

Άσκηση 6: In Fig 5-12 we have two ELECTION message circulating simultaneously. While it does no harm to have two of them, it would be more elegant if one could be killed off. Devise an algorithm for doing this without affecting the operation of the basic election algorithm.

Λύση:

Ένας τρόπος για να «σκοτώσουμε» τη μία αίτηση για coordinator είναι όταν ο κόμβος ο οποίος έστειλε τη δεύτερη αίτηση δεχτεί αίτηση για coordinator από κάποιον άλλον κόμβο με μεγαλύτερη ετικέτα να κάνει broadcast ένα μήνυμα με το οποίο να σημαίνει ότι παραιτείται από την αίτηση για coordinator και πως όποιος κόμβος εκείνη τη στιγμή επεξεργάζεται εκείνη την αίτηση να την απορρίψει και να μην την αναμεταδώσει καθώς επίσης και από όποιον κόμβο έχει ήδη περάσει η αίτηση ο

κόμβος αυτός να διαγράψει ότι πληροφορία είχε κρατήσει σχετικά με την υποδιαγραφή αίτηση.