

Κατανεμημένα Συστήματα

Παναγιώτα Φατούρου

3^ο Σετ Ασκήσεων

Προθεσμία παράδοσης: Τρίτη, 9/6, ώρα 13:00-14:00
(στο βοηθό του μαθήματος, γραφείο A25)

Άσκηση 1

- α. Περιγράψτε την εκτέλεση του EIGStop για 4 διεργασίες και δύο αποτυχίες, δεδομένου ότι οι διεργασίες έχουν αρχικές τιμές 1,0,0 και 0, αντίστοιχα. Υποθέστε ότι οι διεργασίες 1 και 2 αποτυγχάνουν με τη διεργασία 1 να καταρρέει στον 1^ο γύρο έχοντας στείλει το μήνυμά της μόνο στη διεργασία 2, ενώ η διεργασία 2 καταρρέει στο 2^ο γύρο αφού έχει στείλει το μήνυμά της στις διεργασίες 1 και 3 αλλά όχι στη διεργασία 4.
- β. Θεωρήστε τον αλγόριθμο EIGStop σε σύστημα όπου μπορούν να συμβούν f αποτυχίες. Υποθέστε ότι ο αλγόριθμος αντί να εκτελείται για $f+1$ γύρους, εκτελείται μόνο για f γύρους. Αφού ο αλγόριθμος δεν εκτελείται για αρκετούς γύρους, το δένδρο δεν διανθίζεται πλήρως και ο αλγόριθμος αποφασίζει την έξοδό του βάσει των τιμών του τελευταίου διανθισμένου επιπέδου στο δένδρο κατά τον τερματισμό (ακολουθώντας τον ίδιο κανόνα με τον αρχικό αλγόριθμο). Περιγράψτε εκτέλεση στην οποία καταστρατηγείται η ορθότητα του αλγορίθμου.

Άσκηση 2

Το πρόβλημα k -set consensus ορίζεται ως εξής. Κάθε επεξεργαστής ξεκινά με κάποια αυθαίρετη ακέραια τιμή x_i και πρέπει να επιστρέψει μια ακέραια τιμή y_i έτσι ώστε:

Validity: $y_i \in \{x_0, \dots, x_{n-1}\}$

k-Agreement: ο αριθμός των διαφορετικών τιμών που επιστρέφουν οι διεργασίες είναι το πολύ k .

Αποδείξτε ότι ο αλγόριθμος που περιγράφεται παρακάτω επιλύει το πρόβλημα k -set consensus, δεδομένου ότι μπορούν να συμβούν f crash failures στο σύστημα, όπου $f < n$. Ποια είναι η χρονική πολυπλοκότητα του αλγορίθμου αυτού;

Initially $V_i = \{x_i\}$ // code for process p_i

Round r , $0 < r \leq f/k + 1$:

send V to all processors;

receive S_j from p_j , $0 \leq j \leq n-1$, $j \neq i$;

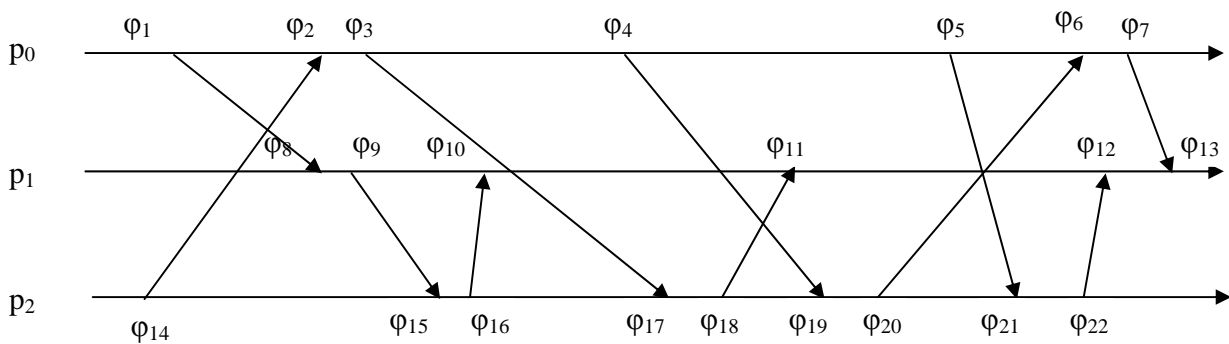
$V := V \cup \bigcup_{i=0}^{n-1} S_j$

if ($r == f/k+1$) then $y := \min(V)$;

Άσκηση 3

Σας δίνεται η εκτέλεση που φαίνεται στο Σχήμα 1 ενός ασύγχρονου συστήματος στο οποίο οι επεξεργαστές επικοινωνούν με ανταλλαγή μηνυμάτων.

- i. Περιγράψτε ποια γεγονότα του επεξεργαστή p_0 σχετίζονται με ποια γεγονότα του επεξεργαστή p_1 μέσω της σχέσης happen-before.
- ii. Χρησιμοποιήστε τον απλό αλγόριθμο απόδοσης λογικών χρονοσφραγίδων (logical timestamps) που συζητήθηκε στο μάθημα για να αποδώσετε λογικές χρονοσφραγίδες στα γεγονότα της εκτέλεσης του Σχήματος 1.
- iii. Χρησιμοποιήστε τον αλγόριθμο απόδοσης διανυσματικών χρονοσφραγίδων (vector timestamps) που μελετήθηκε στο μάθημα για να αποδώσετε διανυσματικές χρονοσφραγίδες στα γεγονότα της εκτέλεσης του Σχήματος 1.



Σχήμα 1