



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

Ακ. έτος 2018-2019, 9ο Εξάμηνο ΗΜ&ΜΥ

Ν. Κοζύρης, Κ. Δόκα

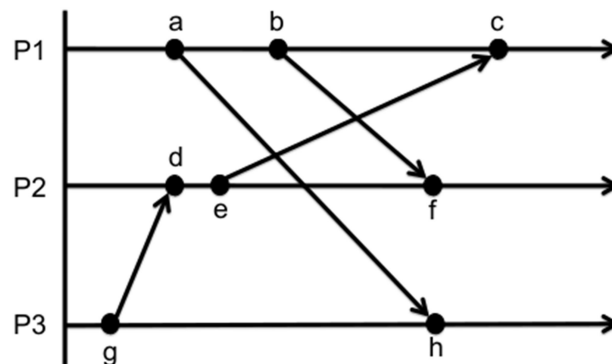
Εξέταση Ιανουαρίου 2019

Χρόνος εξέτασης: 2.5 ώρες

Θέμα 1 (25%)

A. Θεωρήστε τις διεργασίες P1, P2 και P3 οι οποίες παράγουν τα παρακάτω γεγονότα {a..h}. Τα βέλη αντιπροσωπεύουν την ανταλλαγή μηνυμάτων ανάμεσα στις διεργασίες.

Χρησιμοποιήστε διανυσματικά ρολόγια και καταγράψτε τις χρονοσφραγίδες για όλα τα γεγονότα και καταγράψτε όλα τα πιθανά ζεύγη από ταυτόχρονα γεγονότα. Θεωρήστε όλα τα ρολόγια αρχικοποιημένα στο (0, 0, 0). (10%)



B. Θεωρήστε ένα κατανεμημένο σύστημα με 4 διεργασίες που χρησιμοποιεί χρονοσφραγίδες Lamport για λογική ταξινόμηση. Θεωρήστε ότι $L_i(m_j, \text{send})$ είναι η χρονοσφραγίδα Lamport της αποστολής του μηνύματος j από τη διεργασία i, ενώ $L_i(m_j, \text{recv})$ η χρονοσφραγίδα Lamport της λήψης του μηνύματος j από τη διεργασία i. Τα λογικά ρολόγια όλων των διεργασιών έχουν αρχικοποιηθεί στο 0. Βασιζόμενοι στις παρακάτω χρονοσφραγίδες, απαντήστε στις ερωτήσεις που ακολουθούν:

$L_1 (m_1, \text{ send}) = 1$
 $L_1 (m_2, \text{ send}) = 2$
 $L_2 (m_1, \text{ recv}) = 2$
 $L_3 (m_2, \text{ recv}) = 3$
 $L_2 (m_3, \text{ send}) = 3$
 $L_3 (m_4, \text{ send}) = 4$
 $L_3 (m_3, \text{ recv}) = 5$
 $L_4 (m_4, \text{ recv}) = 5$
 $L_4 (m_5, \text{ send}) = 6$
 $L_1 (m_5, \text{ recv}) = x$
 $L_4 (m_6, \text{ send}) = 7$
 $L_1 (m_7, \text{ send}) = 8$
 $L_3 (m_7, \text{ recv}) = 9$
 $L_3 (m_6, \text{ recv}) = 10$

i) Ποια είναι η τιμή του x ; Αν δεν μπορεί να καθοριστεί, απαντήστε “άγνωστη”. (5%)

ii) Για τα παρακάτω γεγονότα, κυκλώστε αυτό που συνέβη πριν από το άλλο και εξηγήστε συνοπτικά. Αν δεν μπορούμε να αποφανθούμε, κυκλώστε “άγνωστο”. (10%)

- $L_1 (m_1, \text{ send})$ $L_4 (m_5, \text{ send})$ άγνωστο
- $L_2 (m_3, \text{ send})$ $L_3 (m_4, \text{ send})$ άγνωστο
- $L_3 (m_4, \text{ send})$ $L_3 (m_3, \text{ recv})$ άγνωστο
- $L_2 (m_3, \text{ send})$ $L_4 (m_6, \text{ send})$ άγνωστο
- $L_4 (m_6, \text{ send})$ $L_1 (m_7, \text{ send})$ άγνωστο

Θέμα 2 (30%)

Έχετε ένα dataset με tweets της μορφής

```
tweetId                      author                      status
```

όπου tweetId το αναγνωριστικό της εγγραφής, author ο χρήστης που δημοσίευσε το tweet και status το περιεχόμενό του. Στο twitter, κάποια tweets είναι αναδημοσιεύσεις παλαιότερων μηνυμάτων (retweets). Τα retweets ξεκινούν από RT που ακολουθείται από το tweet που αναδημοσιεύεται. Για παράδειγμα:

```

263748                      el_professor                      oh bella ciao, bella ciao, bella ciao ciao ciao
324234                      berlin                      RT oh bella ciao, bella ciao, bella ciao ciao ciao
645645                      tokyo                      love is a good reason for everything to fall apart
567654                      el_professor                      sometimes what's wrong is the only choice
345674                      denver                      RT sometimes what's wrong is the only choice
345674                      rio                      RT love is a good reason for everything to fall apart
754564                      tokyo                      RT sometimes what's wrong is the only choice
235322                      nairobi                      RT sometimes what's wrong is the only choice
859234                      el00052                      taking Distributed Systems final, lol

```

Χρησιμοποιήστε MapReduce για να βρείτε:

(α) Το αριθμό των retweets για κάθε status που έχει αναδημοσιευτεί τουλάχιστον μια φορά. (10%) Στο παράδειγμά μας το αποτέλεσμα θα ήταν:

```
oh bella ciao, bella ciao, bella ciao ciao ciao      1
love is a good reason for everything to fall apart    1
sometimes what's wrong is the only choice            3
```

(β) Για κάθε author, το status με τα περισσότερα retweets. (20%) Στο παράδειγμά μας το αποτέλεσμα θα ήταν:

```
el_professor      sometimes what's wrong is the only choice
tokyo              love is a good reason for everything to fall apart
```

Εξηγήστε όλες τις λειτουργίες λεκτικά και δώστε ψευδοκώδικα. Ο ψευδοκώδικας θα πρέπει να περιλαμβάνει για κάθε εργασία MapReduce τις συναρτήσεις map και reduce στην παρακάτω μορφή:

```
MAP(key1, value1):                                REDUCE(key2, list(value2)) :
//επεξεργασία για κάθε                          //επεξεργασία για κάθε
//<key1,value1>                                  //<key2, list(value2)>
emit(key2,value2)                                emit(key3,value3)
```

Θέμα 3 (20%)

Αποφασίζετε να γράψετε τον δικό σας αλγόριθμο για κατανεμημένο αμοιβαίο αποκλεισμό γιατί κουραστήκατε να προσπαθείτε να καταλάβετε αλγόριθμους άλλων. Ο αλγόριθμος αυτός υποθέτει ότι α) τα δικτυακά κανάλια είναι αξιόπιστα και παραδίδουν τα μηνύματα με FIFO διάταξη και β) οι διεργασίες δεν αποτυγχάνουν. Παρακάτω δίνεται η περιγραφή του αλγόριθμου που τρέχει σε κάθε διεργασία p_i .

```
#1: On initialization
    state := RELEASED;
    cnt := 0;
    for each process  $p_j$  ( $j \neq i$ ) in the group
        rcvj := 0;

#2: To enter the critical section
    state := WANTED; cnt++;
    for each process  $p_j$  ( $j \neq i$ ) in the group,
        Send a request to  $p_j$  with rcvj attached to it;
    wait until (number of replies received = (N-1));
    state := HELD;

#3: On receipt of a request from  $p_j$  at  $p_i$  ( $i \neq j$ )
    tmp_rcv := rcvi value attached in the request from  $p_j$ 
    if (state = HELD)
        queue the request from  $p_j$  without replying;
    else if (state = WANTED and tmp_rcv < cnt)
        queue the request from  $p_j$  without replying;
    else
        reply immediately to  $p_j$ ;
        rcvj++;

#4: To exit the critical section
    state := RELEASED;
```

```
for each queued request,  
    //Say the request is from pj  
    Send a reply to pj;  
    rcvj++;
```

α) Εξασφαλίζει ο παραπάνω αλγόριθμος liveness; Αν η απάντηση είναι ναι, αποδείξτε το. Αν όχι, δώστε ένα παράδειγμα όπου δεν εξασφαλίζεται liveness.(10%)

β) Εξασφαλίζει ο παραπάνω αλγόριθμος safety; Αν η απάντηση είναι ναι, αποδείξτε το. Αν όχι, δώστε ένα παράδειγμα όπου δεν εξασφαλίζεται safety.(10%)

Θέμα 4 (25%)

Θεωρήστε τα παρακάτω χρονοπρογράμματα. $R_i(X)$ σημαίνει ότι η δοσοληψία T_i διαβάζει το δεδομένο X , ενώ $W_i(X)$ σημαίνει ότι η δοσοληψία T_i γράφει το δεδομένο X .

S1: $R_1(A), W_2(A), W_1(B), R_2(C), W_2(B), R_1(C), R_2(C), W_2(C)$

S2: $R_2(B), W_2(A), R_1(A), R_3(A), W_1(B), W_2(B), W_3(B)$

Για καθένα από αυτά απαντήστε στις ακόλουθες ερωτήσεις και εξηγήστε:

- i) Είναι το χρονοπρόγραμμα σειριακό;
- ii) Είναι το χρονοπρόγραμμα σειριοποιήσιμο;
- iii) Είναι το πρόγραμμα σειριοποιήσιμο συγκρούσεων;
- iv) Έχει το χρονοπρόγραμμα dirty read;
- v) Έχει το χρονοπρόγραμμα lost update;