

Κατανεμημένος Υπολογισμός – Μεταπτυχιακό Μάθημα Παναγιώτα Φατούρου

2ο Σετ Ασκήσεων

Προθεσμία Παράδοσης: Δευτέρα 4/5, ώρα 16:00, στο μάθημα

Άσκηση 1

1. Παρουσιάστε (event-driven) ψευδοκώδικα για τον Αλγόριθμο Εκλογής Αρχηγού που έχει πολυπλοκότητα επικοινωνίας $O(n^2)$ και αποδείξτε φορμαλιστικά την ορθότητα και την πολυπλοκότητά του. Στον ψευδοκώδικα θα πρέπει να φαίνονται και οι ενέργειες που πραγματοποιεί μια διεργασία όταν λαμβάνει το μήνυμα <terminate>.
2. Υποθέστε ότι ο Αλγόριθμος 5 (an $O(n \log n)$ asynchronous leader election algorithm) τροποποιείται ώστε να χρησιμοποιεί m^k -γειτονιές αντί για 2^k -γειτονιές, όπου $m > 2$. Αναλύστε την πολυπλοκότητα επικοινωνίας και τη χρονική πολυπλοκότητα του τροποποιημένου αλγορίθμου. Συγκρίνετε τις νέες πολυπλοκότητες με τις αντίστοιχες του αυθεντικού αλγορίθμου.

Άσκηση 2

Έστω ότι δύο διεργασίες εκτελούν τον κώδικα που φαίνεται στο παρακάτω σχήμα:

```
#define N 5
```

```
shared int tally = 0;          /* διαμοιραζόμενη μεταβλητή */
```

Διεργασία A	Διεργασία B
<pre>int tmpA; /* τοπική μεταβλητή */ for (i=0; i < N; i++) { tmpA = tally; tmpA = tmpA + 1; tally = tmpA; }</pre>	<pre>int tmpB; /* τοπική μεταβλητή */ for (i=0; i < N; i++) { tmpB = tally; tmpB = tmpB + 1; tally = tmpB; }</pre>

- α. Περιγράψτε σενάριο εκτέλεσης των δύο διεργασιών στο οποίο η τελική τιμή της tally είναι 2.
- β. Ποια είναι η μεγαλύτερη και ποια η μικρότερη τιμή που μπορεί να έχει η tally;
- γ. Πως διαφοροποιείται η απάντησή σας στο ερώτημα β. αν το N έχει τιμή 50 αντί για 5;
- δ. Πως διαφοροποιείται η απάντησή σας στο ερώτημα β αν $N = 5$ και ο κώδικας του σχήματος εκτελείται από περισσότερες από δύο διεργασίες (π.χ., από m διεργασίες);

Άσκηση 3

Εξετάστε αν η λύση στο πρόβλημα του αμοιβαίου αποκλεισμού για δύο διεργασίες, που προτείνεται στη συνέχεια, ικανοποιεί τις συνθήκες του αμοιβαίου αποκλεισμού και της προόδου. Η ρουτίνα `GetPid()` επιστρέφει τον ίδιο πάντα ακέραιο όταν καλείται από τη διεργασία 0 και τον ίδιο πάντα ακέραιο, διαφορετικό από αυτόν που επιστρέφει στη διεργασία 0 όταν καλείται από τη διεργασία 1. Ο κώδικας που εκτελεί η κάθε διεργασία είναι ο ακόλουθος:

```
shared int busy = 0;  
shared int trying = -1;
```

```
start:  trying = GetPid();  
        if (busy == 1) goto start;  
        busy = 1;  
        if (trying != GetPid()) then  
            begin  
                busy = 0;  
                goto start;  
            end
```

```
<κρίσιμο τμήμα>;
```

```
busy = 0;
```

```
<μη-κρίσιμο τμήμα>;
```