

Παράλληλος και Κατανεμημένος Υπολογισμός

Εξετάσεις Ιουνίου 2020

Θέμα 1

Έστω ένας μονοδιάστατος πίνακας a μεγέθους N που περιέχει τυχαίους ακέραιους. Γράψτε παράλληλο πολυνηματικό κώδικα ή ψευδοκώδικα που να υπολογίζει (α) το άθροισμα των ακεραίων που είναι θετικοί (ή μηδέν) και (β) το άθροισμα των ακεραίων που είναι αρνητικοί. Διαθέτετε P επεξεργαστές. Προσέξτε ότι τα δύο αθροίσματα μπορούν να υπολογιστούν παράλληλα, ανεξάρτητα το ένα από το άλλο.

```
posSum = 0; negSum = 0;
for (i=0; i<N; i++){
    if (a[i] >= 0) posSum = posSum + a[i];
    else negSum = negSum + a[i];
}
```

Σχεδιάστε τρεις λύσεις:

- 1.1. Με απλή απεικόνιση και άθροιση των επιμέρους αποτελεσμάτων από το κυρίως πρόγραμμα.
- 1.2. Με απεικόνιση και αναγωγή αλλά με λανθασμένη χρήση του αμοιβαίου αποκλεισμού, η οποία προκαλεί υπερβολική καθυστέρηση.
- 1.3. Με απεικόνιση και αναγωγή και με ορθή χρήση του αμοιβαίου αποκλεισμού.

Προσοχή: Σε όλες τις λύσεις απαιτείται σχολιασμός στα σημαντικά σημεία του κώδικα που να εξηγεί την αναμενόμενη συμπεριφορά του. Λύση χωρίς σχολιασμό δεν λαμβάνεται υπ' όψη.

Παράλληλος και Κατανεμημένος Υπολογισμός

Εξετάσεις Ιουνίου 2020

Θέμα 2

Γράψτε ταυτόχρονο πολυνηματικό κώδικα ή ψευδοκώδικα για το πρόβλημα N παραγωγών - 1 καταναλωτή και Buffer με `άπειρο` size. Σχεδιάστε δύο λύσεις:

2.1. Με χρήση κλειδώματος και μεταβλητών συνθήκης.

2.2. Με σηματοφόρους.

Προσοχή: Ο κώδικας της γενικής λύσης δεν λαμβάνεται υπ'όψη. Σε όλες τις λύσεις απαιτείται σχολιασμός στα σημαντικά σημεία του κώδικα που να εξηγεί γιατί προστέθηκαν ή αφαιρέθηκαν οι μηχανισμοί ταυτοχρονισμού. Λύση χωρίς σχολιασμό δεν λαμβάνεται υπ'όψη.

Δίνεται ενδεικτικός κώδικας της γενικής λύσης (δεν είναι πλήρης, δεν έχει μηχανισμούς ταυτοχρονισμού):

```
Buffer (int size) {
    int[size] contents;
    boolean bufferEmpty = true; boolean bufferFull = false;
    int front = 0, back = size - 1; counter = 0;

    put(int data) {
        if (!bufferFull) {
            back = (back + 1) % size;
            contents[back] = data;
            counter++;
            bufferEmpty = false;
            if (counter == size) bufferFull = true;
        }
    }

    int get() {
        int data = 0;
        if (!bufferEmpty) {
            data = contents[front];
            front = (front + 1) % size;
            counter--;
            bufferFull = false;
            if (counter == 0) bufferEmpty = true;
        }
        return data;
    }
}
```

Παράλληλος και Κατανεμημένος Υπολογισμός

Εξετάσεις Ιουνίου 2020

Θέμα 3

Γράψτε κατανεμημένο κώδικα ή ψευδοκώδικα για τον παράλληλο υπολογισμό του π με χρήση υποδοχών (sockets). Ο διακομιστής-συντονιστής είναι πολυνηματικός και υπάρχουν P πελάτες-εργαζόμενοι. Οι πελάτες-εργαζόμενοι ίναι διεργασίες που εκτελούνται σε διαφορετικούς υπολογιστές, συνδέονται χωρίς καθυστερήσεις, εργάζονται παράλληλα και γνωρίζουν το P . Για τη κατανομή του φόρτου εργασίας numSteps χρησιμοποιείτε κυκλική κατανομή με μέγεθος block 1.

Προσοχή: Δώστε ιδιαίτερη έμφαση (α) στη δομή του πολυνηματικού διακομιστή (β) στα βήματα επικοινωνίας πελάτη-διακομιστή και (γ) στο πρωτόκολλο της εφαρμογής τόσο στο πελάτη όσο και στο διακομιστή. Κώδικας χωρίς σχολιασμό δεν λαμβάνεται υπ' όψη.

Κώδικας υπολογισμού:

```
double sum = 0.0;
double step = 1.0 / (double)numSteps;
for (long i=0; i<numSteps; ++i) {
    double x = ((double)i+0.5)*step;
    sum += 4.0/(1.0+x*x);
}
double pi = sum * step;
```