

## Λογισμικό και Προγραμματισμός Συστημάτων Υψηλής Επίδοσης Χειμερινό Εξάμηνο 2010-2011

### 1<sup>η</sup> Άσκηση

Ημερομηνία Παράδοσης: Παρασκευή 25/11/2011, 15:00

#### Παραλληλοποίηση επαναλήψεων βρόχου

Στόχος είναι η παράλληλη εκτέλεση των επαναλήψεων ενός παραλληλοποιήσιμου βρόχου. Οι επαναλήψεις ενός τέτοιου βρόχου μπορούν να διαμοιραστούν σε ομάδες και να ανατεθούν για εκτέλεση σε νήματα. Η διαμοίραση αυτή μπορεί να είναι είτε λεπτή (fine-grain) είτε αδρή (coarse-grain) ανάλογα με το ποσοστό παραλληλισμού που επιθυμούμε να εκμεταλλευτούμε.

Αντίστοιχα η ανάθεση των ομάδων στα νήματα βασίζεται σε διάφορες πολιτικές χρονοδρομολόγησης επιπέδου χρήστη. Ζητούμενο της άσκησης είναι να υλοποιήσετε μια συνάρτηση που θα απελευθερώνει τον προγραμματιστή από τις λεπτομέρειες της κατάτμησης του βρόχου και θα περιλαμβάνει ως επιλογή πολιτικές δρομολόγησης που περιγράφονται στη συνέχεια. Η συνάρτηση αυτή θα είναι η:

- **int pthread\_execute\_loop (void (\*func)(void \*arg, int low, int high), void \*arg, int policy, int chunk, int nthreads, int low, int high)**

Όσον αφορά στα ορίσματα της, η συνάρτηση func παρέχεται από τον προγραμματιστή και περιέχει τον προς παραλληλοποίηση βρόχο (και μόνο αυτό) παραμετροποιημένο ως προς τα όριά του. Οι παράμετροι low και high της func είναι το κάτω και το πάνω όριο των επαναλήψεων του loop που θα εκτελεστούν από κάθε νήμα, ενώ η παράμετρος arg είναι στη διάθεση του προγραμματιστή και μπορεί να χρησιμοποιηθεί ανάλογα με τις απαιτήσεις του κώδικά του. Παίρνει την τιμή της από την παράμετρο arg της pthread\_execute\_loop. Η παράμετρος policy χρησιμοποιείται για την επιλογή της πολιτικής χρονοδρομολόγησης που θα εφαρμοσθεί. Οι επιτρεπτές τιμές της είναι οι ακόλουθες: STATIC και DYNAMIC. Σε περίπτωση που χρησιμοποιηθεί τιμή πέρα από τις επιτρεπόμενες επιστρέφεται EINVAL.

Η chunk ορίζει (ανάλογα με την πολιτική) το ακριβές ή το ελάχιστο μέγεθος του chunk. Εάν ο προγραμματιστής δεν επιθυμεί να ορίσει chunk, η μεταβλητή τίθεται στην τιμή -1. Με την nthreads δηλώνεται ο αριθμός των νημάτων που θα χρησιμοποιηθούν, ενώ low και high είναι αντίστοιχα το κάτω και πάνω όριο του αρχικού βρόχου. Στην περίπτωση που ο αριθμός των νημάτων που θα χρησιμοποιηθούν είναι < 1 επιστρέφεται EINVAL. Σε περίπτωση επιτυχούς εκτέλεσης η συνάρτηση επιστρέφει 0.

Αναλυτικότερα, οι πολιτικές δρομολόγησης των επαναλήψεων έχουν ως εξής:

- **Static:** Έστω N οι επαναλήψεις και P τα νήματα. Οι επαναλήψεις χωρίζονται σε  $[P/N]$  ομάδες. Κάθε ομάδα αποτελείται από διαδοχικές επαναλήψεις και ανατίθεται σε ένα νήμα (η ομάδα 0 ανατίθεται στο νήμα 0, η ομάδα 1 στο νήμα 1 ..., η ομάδα N-1 στο νήμα N-1). Στην υλοποίηση που σας ζητείται, η πολιτική δεν χρησιμοποιεί κάποιο chunk.
- **Dynamic:** Κάθε φορά που ένα από τα νήματα βρίσκεται σε απραξία δεσμεύει και εκτελεί τις chunk επόμενες επαναλήψεις. Στην περίπτωση που το chunk δεν έχει οριστεί, θεωρείται ίσο με 1.

#### Παράδειγμα χρήσης της pthread\_execute\_loop

Έστω ότι ο προγραμματιστής θέλει να παραλληλοποιήσει τον παρακάτω βρόχο με πολιτική dynamic, chunk 10 και 16 threads χρησιμοποιώντας την pthread\_execute\_loop:

```
void main() {
    int i;
    double A[2000];
    for (i = 5; i < 1030; i++) {
        A[i] = i + 10;
    }
    print("A[340] = %f", A[340]);
}
```

Ο αντίστοιχος κώδικας θα είναι:

```
#include "loop.h"

void ploop_func(void *arg, int low, int high) {
    double *A = (double *) arg;
    int loop_counter;
    for (loop_counter = low, loop_counter <= high; loop_counter++) {
        A[loop_counter] = loop_counter + 10;
    }
}

void main() {
    double A[2000];
    pthread_execute_loop(ploop_func, (void *)A, DYNAMIC, 10, 16, 5, 1029);
    print("A[340] = %f", A[340]);
}
```

### Παρατηρήσεις

1. Υλοποιήστε τη συνάρτηση **pthread\_execute\_loop** στα αρχεία **loop.c** και **loop.h**
2. Πέρα από τον κώδικα του παραδείγματος, χρησιμοποιήστε τη συνάρτηση για να παραλληλοποιήσετε το βρόχο ενός προγράμματος που υλοποιεί την πράξη  $C = A + B$ , όπου  $A, B$  και  $C$  διανύσματα μεγέθους  $N$ .
3. Προσπαθήστε να υλοποιήσετε τη συνάρτηση έτσι ώστε να είναι δυνατή η ταυτόχρονη κλήση της από διαφορετικά νήματα.

### Παραδοτέα

Οι ασκήσεις θα πραγματοποιηθούν σε ομάδες 1-2 ατόμων. Οι ομάδες αυτές θα ισχύσουν και τα επόμενα σύνολα ασκήσεων.

Η παράδοση της άσκησης πρέπει να γίνει ηλεκτρονικά με αποστολή στη διεύθυνση ηλεκτρονικού ταχυδρομείου του διδάσκοντα (**xdoukas (at) ceid.upatras.gr**) μέχρι την **Παρασκευή 25/11/2011** και ώρα **3μμ. (15:00)**. Μετά την συγκεκριμένη διορία, δεν θα γίνεται δεκτή καμία άσκηση. Το θέμα του μηνύματος θα είναι **ADVPAR-EX1 (AM:X1,X2)**, όπου  $X1, X2$  οι αριθμοί μητρώου του κάθε μέλους της ομάδας σας. Συνημμένα θα υπάρχει συμπιεσμένο αρχείο το οποίο θα περιέχει:

1. Πηγαίο κώδικα, γραμμένο σε C, για κάθε περίπτωση, με τα απαραίτητα **σχόλια**.
2. **Αναλυτική αναφορά** για την εργασία σας όπου θα συμπεριλαμβάνεται το ονοματεπώνυμό σας, ο αριθμός μητρώου σας και η διεύθυνση ηλεκτρονικού ταχυδρομείου που χρησιμοποιείτε. Θα πρέπει να αναφέρετε τα βασικά χαρακτηριστικά του υπολογιστικού συστήματος στο οποίο εκτελέσατε τα προγράμματά σας. Στην αναφορά θα περιγράφετε και τεκμηριώνετε την υλοποίησή και τις σχεδιαστικές αποφάσεις σας και θα παρουσιάσετε τα αποτελέσματά σας.

Για τυχόν απορίες ή διευκρινίσεις μπορείτε να χρησιμοποιήσετε τις ώρες διδασκαλίας του μαθήματος ή να επικοινωνήσετε με τον διδάσκοντα μέσω e-mail. Αν απαιτείται, θα προστεθούν διευκρινίσεις, βοηθητικές πληροφορίες και απαντήσεις στην ιστοσελίδα του μαθήματος.

**Καλή επιτυχία!**

**Παναγιώτης Χατζηδούκας (Π.Δ. 407/80)**