

Πανεπιστήμιο Ιωαννίνων  
Τμήμα Πληροφορικής & Τηλεπικοινωνιών  
Μάθημα: Κατανεμημένα και Παράλληλα Συστήματα  
Διδάσκων: Νικόλαος Καλλιμάνης

## 2<sup>η</sup> Εργαστηριακή Ενότητα

### Προγραμματισμός με νήματα (Posix Threads)

Άσκηση 4: Δημιουργήστε ένα αρχείο με όνομα `exer04.c` και χρησιμοποιείτε τον παρακάτω κώδικα. Τι πρόκειται να εμφανίσει ο παρακάτω κώδικας;

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>

#define ARRAY_SIZE 1000

volatile int shared_array[ARRAY_SIZE];
volatile unsigned int num_threads;

void *thread_work(void *arg) {
    long mytid = (long)arg;
    long sum = 0;

    int chunk_size = ARRAY_SIZE / num_threads;
    int start = mytid * chunk_size;
    int end = (mytid + 1) * chunk_size;
    if (mytid == num_threads - 1) {
        // Adjust the end of the last thread to cover any remaining elements
        end = ARRAY_SIZE;
    }
    for (int i = start; i < end; ++i)
        sum += shared_array[i];
    printf("Thread: %ld -- Reads array: start: %d - end: %d - sum: %ld\n", mytid, start, end - 1, sum);
    return NULL;
}

int main(int argc, char* argv[]) {
    if (argc != 2) {
        printf("Usage: %s <num_threads>\n", argv[0]);
        return EXIT_FAILURE;
    }
    num_threads = atoi(argv[1]);
    if (num_threads <= 0) {
        printf("Number of threads must be a positive integer.\n");
        return EXIT_FAILURE;
    }

    for (long i = 0; i < ARRAY_SIZE; i++)
        shared_array[i] = i*i;
    pthread_t *threads = malloc(num_threads * sizeof(pthread_t));
    // Create threads
    for (long t = 0; t < num_threads; t++)
        pthread_create(&threads[t], NULL, thread_work, (void *)t);
    printf("This is the main thread\n");
    for (long t = 0; t < num_threads; t++)
        pthread_join(threads[t], NULL);
    return 0;
}
```

Μεταγλωττίστε τον κώδικα με την εντολή `gcc -Wall -o exer04.run exer04.c -lpthread`

Άσκηση 5: Αναπτύξτε ένα πρόγραμμα που θα δημιουργεί έναν κοινόχρηστο πίνακα 1000 ακεραίων `shared_array`. Ο χρήστης θα έχει τη δυνατότητα να δίνει ως όρισμα τον αριθμό των νημάτων που θέλει να δημιουργήσει. Το κύριο (γονικό) νήμα θα αρχικοποιεί ένα μέρος του πίνακα `shared_array`, όπου το  $i$ -οστό στοιχείο του πίνακα θα πρέπει να λαμβάνει ως αρχική τιμή την τιμή  $i^2$ . Βασιστείτε στον κώδικα της Άσκησης 4.

α) Να υπολογιστεί ο μέσος όρος των στοιχείων του πίνακα χωρίς να χρησιμοποιήσετε κοινόχρηστες μεταβλητές πλην του πίνακα `shared_array`, και μιας μεταβλητής που περιέχει τον αριθμό των νημάτων που δημιουργούνται.

β) Να υπολογιστεί ο μέσος όρος των στοιχείων του πίνακα χρησιμοποιώντας κοινόχρηστες μεταβλητές.

Άσκηση 6: Αναπτύξτε ένα πρόγραμμα όπου θα πολλαπλασιάζει ένα μονοδιάστατο πίνακα  $A$  μεγέθους  $N$  με ένα διάνυσμα  $B$  μεγέθους  $N$  στοιχείων. Το  $i$ -οστό στοιχείο του πίνακα  $A$  θα πρέπει να λαμβάνει ως αρχική τιμή την τιμή  $i + 1.0$ , ενώ το  $i$ -οστό στοιχείο του διανύσματος  $B$  πρέπει να έχει τιμή ίση με  $1.0/(i+1.0)$ . Το αποτέλεσμα θα αποθηκεύεται σε ένα κοινόχρηστο πίνακα  $C$ . Βασιστείτε στον κώδικα της Άσκησης 4.