

1^η Εργαστηριακή Ενότητα

Προετοιμασία συστήματος: Βασική εγκατάσταση εργαλείων και βιβλιοθηκών (Ubuntu/WSL)

Εκτελέστε τις ακόλουθες σε ένα τερματικό για να εγκαταστήσετε τις απαραίτητες βιβλιοθήκες και εργαλεία για να ετοιμάσετε το προγραμματιστικό περιβάλλον:

```
sudo apt-get update  
sudo apt-get install gcc gdb make libnuma1 libnuma-dev htop
```

Προγραμματισμός με διεργασίες

Άσκηση 1: Δημιουργήστε ένα αρχείο με τον παρακάτω κώδικα και όνομα 'exer01.c'. Τι πρόκειται να εμφανίσει ο παρακάτω κώδικας;

```
#include <stdio.h>  
#include <unistd.h>  
  
int main(int argc, char** argv){  
    int x = 5;  
    printf("A. pid=%d, value of x is %d\n", getpid(), x);  
    pid_t chid=fork();  
  
    if (chid==0) {  
        x++;  
        printf("B. pid=%d, value of x is %d\n", getpid(), x);  
    } else {  
        x--;  
        printf("C. pid=%d, value of x is %d\n", getpid(), x);  
    }  
    printf("D. pid=%d, value of x is %d\n", getpid(), x);  
    return 0;  
}
```

Μεταγλωττίστε τον κώδικα με την εντολή: `gcc -Wall -o exer01.run exer01.c.`

Προγραμματισμός με νήματα

Άσκηση 2: Δημιουργήστε ένα αρχείο με τον παρακάτω κώδικα και όνομα 'exer02.c'. Τι πρόκειται να εμφανίσει ο παρακάτω κώδικας;

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

volatile unsigned int num_threads;

void *thread_work(void *arg) {
    long mytid = (long)arg;
    printf("Total threads: %d - Current Thread %ld\n", num_threads, mytid);
    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;
    }

    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 exer02.run exer02.c -lpthread`

Άσκηση 3: Δημιουργείστε ένα 'exer03.c' με κώδικα στο οποίο θα δημιουργείται ένα κοινόχρηστος πίνακας 1000 ακεραίων 'shared_array'. Ο χρήστης θα έχει τη δυνατότητα να δίνει ως όρισμα τον αριθμό των νημάτων που θέλει να δημιουργήσει, και κάθε νήμα θα αρχικοποιεί ένα μέρος του πίνακα 'shared_array'. Το i-οστό στοιχείο του πίνακα θα πρέπει να λαμβάνει ως αρχική τιμή την τιμή i^2 . Βασιστείτε στον κώδικα της Άσκησης 2.