Week 3; High Performance Computing

Swoyam Pokharel

November 5

Contents

1 · Write a multi-threaded C program to print out all the prime numbers between 1 to 1000, using exactly 3 threads.

```c
#include <pthread.h>
#include <stdio.h>
#include <stdbool.h>
#include <math.h>

void* is_prime(void* num) {
    int casted_num = *(int *)(num);
    if (casted_num<2){
        printf("%d is not prime\n", casted_num);
        return NULL;
    }

    int lim = (int)sqrt(casted_num);
    for (int i = 2; i <= lim; i++) {
        if (casted_num % i == 0) {
            return NULL;
        }
    }
    printf("The Number: %d, is prime\n", casted_num);
    return NULL;
}


int main(){
    pthread_t t1,t2,t3;
    int n1=4, n2=13, n3=22;
    pthread_create(&t1, NULL, is_prime, &n1);
    pthread_create(&t2, NULL, is_prime, &n2);
    pthread_create(&t3, NULL, is_prime, &n3);
    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    pthread_join(t3, NULL);
    return 0;
}
```

2 · Update the above function to ask the user for the input, the numbers to check for prime and the number of threads t odo so

```c
#include <pthread.h>
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <math.h>

void* is_prime(void* num) {
    int n = *(int*)num;

    if (n < 2) {
        printf("%d is not prime\n", n);
        return NULL;
    }

    int lim = (int)sqrt(n);
    for (int i = 2; i <= lim; i++) {
        if (n % i == 0) {
            printf("%d is not prime\n", n);
            return NULL;
        }
    }

    printf("%d is prime\n", n);
    return NULL;
}

int main() {
    int count;

    printf("How many numbers do you want to check for primes? ");
    scanf("%d", &count);

    if (count <= 0) {
        printf("Invalid count.\n");
        return 1;
    }

    pthread_t* threads = malloc(count * sizeof(pthread_t));
    int* numbers = malloc(count * sizeof(int));

    if (!threads || !numbers) {
        printf("Memory allocation failed.\n");
        return 1;
    }

    for (int i = 0; i < count; i++) {
        printf("Enter number %d: ", i + 1);
        scanf("%d", &numbers[i]);
    }
    for (int i = 0; i < count; i++) {
        pthread_create(&threads[i], NULL, is_prime, &numbers[i]);
    }

    for (int i = 0; i < count; i++) {
        pthread_join(threads[i], NULL);
    }

    free(threads);
    free(numbers);

    return 0;
}
```