Experiment Number: 5

Problem Statement: **Implement multithreading for Matrix Operations using Pthreads.**

NAME: Aadesh Chawla ROLLNO: 12

CLASS: TY-IT-A BATCH: B1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Code:**

#include <iostream>

#include <pthread.h>

using namespace std;

const int core = 4;

pthread\_t\* thread[core];

int n;

int matrix\_A[100][100];

int matrix\_B[100][100];

int sum[100][100];

int sub[100][100];

int mult[100][100];

void\* addition(void\* step) {

    int start = \*((int\*)step);

    int end = start + n / core;

    for (int i = start; i < end; i++) {

        for (int j = 0; j < n; j++) {

            sum[i][j] = matrix\_A[i][j] + matrix\_B[i][j];

        }

    }

    pthread\_exit(NULL);

}

void\* subtraction(void\* step) {

    int start = \*((int\*)step);

    int end = start + n / core;

    for (int i = start; i < end; i++) {

        for (int j = 0; j < n; j++) {

            sub[i][j] = matrix\_A[i][j] - matrix\_B[i][j];

        }

    }

    pthread\_exit(NULL);

}

void\* multiplication(void\* step) {

    int start = \*((int\*)step);

    int end = start + n / core;

    for (int i = start; i < end; i++) {

        for (int j = 0; j < n; j++) {

            mult[i][j] = 0;

            for (int k = 0; k < n; k++) {

                mult[i][j] += matrix\_A[i][k] \* matrix\_B[k][j];

            }

        }

    }

    pthread\_exit(NULL);

}

int main() {

    cout << "Enter n: ";

    cin >> n;

    cout << "Enter elements of matrix A:" << endl;

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {

            cout << "A(" << i << "," << j << "): ";

            cin >> matrix\_A[i][j];

        }

    }

    cout << "Enter elements of matrix B:" << endl;

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {

            cout << "B(" << i << "," << j << "): ";

            cin >> matrix\_B[i][j];

        }

    }

    thread[core];  // Array declaration

    int step = 0;

    for (int i = 0; i < core; i++) {

        pthread\_create(&thread[i], NULL, &addition, (void\*)&step);

        step += n / core;

    }

    for (int i = 0; i < core; i++) {

        pthread\_join(thread[i], NULL);

    }

    cout << "Sum Matrix:" << endl;

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {

            cout << sum[i][j] << " ";

        }

        cout << endl;

    }

    cout << "Subtraction Matrix:" << endl;

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {

            cout << sub[i][j] << " ";

        }

        cout << endl;

    }

    cout << "Multiplication Matrix:" << endl;

    for (int i = 0; i < n; i++) {

        for (int j = 0; j < n; j++) {

            cout << mult[i][j] << " ";

        }

        cout << endl;

    }

    delete[] thread;

    return 0;

}

**Output:**

Enter n: 3

Enter elements of matrix A:

A(0,0): 1

A(0,1): 2

A(0,2): 3

A(1,0): 4

A(1,1): 5

A(1,2): 6

A(2,0): 7

A(2,1): 8

A(2,2): 9

Enter elements of matrix B:

B(0,0): 9

B(0,1): 8

B(0,2): 7

B(1,0): 6

B(1,1): 5

B(1,2): 4

B(2,0): 3

B(2,1): 2

B(2,2): 1

Sum Matrix:

10 10 10

10 10 10

10 10 10

Subtraction Matrix:

-8 -6 -4

-2 0 2

4 6 8

Multiplication Matrix:

30 24 18

84 69 54

138 114 90