Experiment Number: 5

Problem Statement: Implement multithreading for Matrix Operations using Pthreads.

NAME:Manoj Dhanraj Mule ROLLNO: 71

CLASS: IT-B BATCH: B3

DATE OF PERFORMANCE: 23/9/2024

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#define MAX 10

int matA[MAX][MAX], matB[MAX][MAX], matSum[MAX][MAX], matProduct[MAX][MAX];

int row1, col1, row2, col2;

typedef struct

{

int row;

int col;

} MatrixIndex;

void \*add\_matrices(void \*arg)

{

MatrixIndex \*index = (MatrixIndex \*)arg;

int i = index->row;

int j = index->col;

matSum[i][j] = matA[i][j] + matB[i][j];

pthread\_exit(0);

}

void \*multiply\_matrices(void \*arg)

{

MatrixIndex \*index = (MatrixIndex \*)arg;

int i = index->row;

int j = index->col;

matProduct[i][j] = 0;

for (int k = 0; k < col1; k++)

{

matProduct[i][j] += matA[i][k] \* matB[k][j];

}

pthread\_exit(0);

}

int main()

{

printf("Enter rows and columns for Matrix A: ");

scanf("%d %d", &row1, &col1);

printf("Enter rows and columns for Matrix B: ");

scanf("%d %d", &row2, &col2);

if (col1 != row2)

{

printf("Matrix multiplication not possible. Columns of A must be equal to rows of B.\n");

return -1;

}

printf("Enter elements of Matrix A:\n");

for (int i = 0; i < row1; i++)

{

for (int j = 0; j < col1; j++)

{

scanf("%d", &matA[i][j]);

}

}

printf("Enter elements of Matrix B:\n");

for (int i = 0; i < row2; i++)

{

for (int j = 0; j < col2; j++)

{

scanf("%d", &matB[i][j]);

}

}

pthread\_t threads[MAX][MAX];

MatrixIndex index[MAX][MAX];

printf("Performing matrix addition...\n");

for (int i = 0; i < row1; i++)

{

for (int j = 0; j < col1; j++)

{

index[i][j].row = i;

index[i][j].col = j;

pthread\_create(&threads[i][j], NULL, add\_matrices, &index[i][j]);

}

}

for (int i = 0; i < row1; i++)

{

for (int j = 0; j < col1; j++)

{

pthread\_join(threads[i][j], NULL);

}

}

printf("Matrix addition result:\n");

for (int i = 0; i < row1; i++)

{

for (int j = 0; j < col1; j++)

{

printf("%d ", matSum[i][j]);

}

printf("\n");

}

printf("Performing matrix multiplication...\n");

for (int i = 0; i < row1; i++)

{

for (int j = 0; j < col2; j++)

{

index[i][j].row = i;

index[i][j].col = j;

pthread\_create(&threads[i][j], NULL, multiply\_matrices, &index[i][j]);

}

}

for (int i = 0; i < row1; i++)

{

for (int j = 0; j < col2; j++)

{

pthread\_join(threads[i][j], NULL);

}

}

printf("Matrix multiplication result:\n");

for (int i = 0; i < row1; i++)

{

for (int j = 0; j < col2; j++)

{

printf("%d ", matProduct[i][j]);

}

printf("\n");

}

return 0;

}

**Output:**

