**12.3 Write a program for Matrix Multiplication in OpenMP.**

**Objectives:**

1. To learn about openMP for better use multicore system.

2. Implement the program OpenMP threads and print prime number task, odd number and Fibonacci series using three thread on core. Comment on performance CPU.

**Theory:**

OpenMP

• OpenMP is an Application Program Interface (API) that may be used to explicitly direct multi-threaded, shared memory parallelism.

• OpenMP is a directive-based method to invoke parallel computations on

share-memory multiprocessors.

• The omp for directive instructs the compiler to distribute loop iterations within the team of threads that encounters this work-sharing construct.

• OpenMP consists of a set of compiler #pragmas that control how the program works. The pragmas are designed so that even if the compiler does not support them, the program will still yield correct behavior, but without any parallelism. Here are two simple example programs demonstrating OpenMP.

**Program:**

#include<stdio.h>

void main()

{

int m1[3][3], m2[3][3], m3[3][3], sum=0, i, j, k;

printf("Enter first matrix element : ");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

scanf("%d",&m1[i][j]);

}

}

printf("Enter second matrix element : ");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

scanf("%d",&m2[i][j]);

}

}

#pragma omp parallel for

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

sum=0;

for(k=0;k<3;k++)

{

sum=sum+m1[i][k] \* m2[k][j];

}

m3[i][j]=sum;

}

}

printf("\nMultiplication of two Matrices : \n");

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

printf("%d\t",m3[i][j]);

}

printf("\n");

}

}

**Output:**

Enter first matrix element:

4 6 8

2 1 4

6 9 3

Enter second matrix element:

1 5 8

4 3 2

7 6 5

Multiplication of two Matrices :

84 86 84

34 37 38

63 75 81

**Conclusion:**

1 Hence we learnt that the Sequential programs can be enhanced with OpenMP directives, leaving the original program essentially intact.

**References:**

[1] http://openmptutorial.com/tutorials/openmp-introduction/