

LAB:NO:07

Objective: To understand multithreading

Task:

Write the same code for matrix multiplication and divide the code into ten threads. Also note the timestamp at the start and end of the program. Give your conclusion.

Matrix Multiplication using Traditional Method:

```
import java.lang.*;
public class MatrixMultiplicationExample{
    public static void main(String args[]){
        System.out.print("Current Time in milliseconds = ");
        System.out.println(System.currentTimeMillis());
        double time1=System.currentTimeMillis();
        //creating two matrices
        int a[][]={{1,1,1},{2,2,2},{3,3,3}};
        int b[][]={{1,1,1},{2,2,2},{3,3,3}};

        //creating another matrix to store the multiplication of two matrices
        int c[][]=new int[3][3]; //3 rows and 3 columns

        //multiplying and printing multiplication of 2 matrices
        for(int i=0;i<3;i++){
            for(int j=0;j<3;j++){
                c[i][j]=0;
                for(int k=0;k<3;k++){
                    {
                        c[i][j]+=a[i][k]*b[k][j];
                    }
                }
                //end of k loop
                System.out.print(c[i][j]+" "); //printing matrix element
            }
            //end of j loop
            System.out.println();//new line
        }
        double time2=System.currentTimeMillis();
        double totalTime=time2-time1;
        System.out.println("totalTime"+totalTime);
    }
}
```

Output:

```
E:\java program>javac MatrixMultiplicationExample.java

E:\java program>java MatrixMultiplicationExample
Current Time in milliseconds = 1558633843324
6 6 6
12 12 12
18 18 18
totalTime0.0
```

Matrix Multiplication using Thread:

```

public class ParentTh {

    public static final int NUM_OF_THREADS = 9;

    public static void main(String args[])
    {
        int row;
        int col;
        int A[][] = {{1,1,1},{2,2,2},{3,3,3}};
        int B[][] = {{1,1,1},{2,2,2},{3,3,3}};
        int C[][] = new int[3][3];
        int threadcount = 0;
        System.out.print("Current Time in milliseconds = ");
        System.out.println(System.currentTimeMillis());
        double time1=System.currentTimeMillis();
        Thread[] thrd = new Thread[NUM_OF_THREADS];

        try
        {
            for(row = 0 ; row < 3; row++)
            {
                for (col = 0 ; col < 3; col++ )
                {
                    // creating thread for multiplications
                    thrd[threadcount] = new Thread(new WorkerTh(row, col, A, B, C));
                    thrd[threadcount].start(); //thread start

                    thrd[threadcount].join(); // joining threads
                    threadcount++;
                }
            }
        }

        catch (InterruptedException ie){}

        // printing matrix A
        System.out.println(" A Matrix : ");
        for(row = 0 ; row < 3; row++)
        {
            for (col = 0 ; col < 2; col++ )
            {
                System.out.print(" "+A[row][col]);
            }
            System.out.println();
        }

        // printing matrix B
        System.out.println(" B Matrix : ");
        for(row = 0 ; row < 2; row++)
        {
            for (col = 0 ; col < 3; col++ )
            {
                System.out.print(" "+B[row][col]);
            }
            System.out.println();
        }

        // printing resulting matrix C after multiplication
        System.out.println(" Resulting C Matrix : ");
        for(row = 0 ; row < 3; row++)
        {
            for (col = 0 ; col < 3; col++ )
            {
                System.out.print(" "+C[row][col]);
            }
            System.out.println();
        }
        double time2=System.currentTimeMillis();
        double totalTime=time2-time1;
    }
}

```

```

        System.out.println("totalTime"+totalTime);

    }

}

class WorkerTh implements Runnable
{
    private int row;
    private int col;
    private int A[][];
    private int B[][];
    private int C[][];

    public WorkerTh(int row, int col, int A[][], int B[][], int C[][])
    {
        this.row = row;
        this.col = col;
        this.A = A;
        this.B = B;
        this.C = C;
    }

    @Override
    public void run()
    {

        for(int k = 0; k < B.length; k++)
        {
            C[row][col] += A[row][k] * B[k][col];
        }

    }

}
}

```

Output:

```

E:\java program>java ParentTh
Current Time in milliseconds = 1558633887907
A Matrix :
1 1
2 2
3 3
B Matrix :
1 1 1
2 2 2
Resulting C Matrix :
6 6 6
12 12 12
18 18 18
totalTime16.0

```