**ASSIGNMENT NO. 3**

**THREAD MANAGEMENT USING PTHREAD LIBRARY**

#include<stdio.h>

#include<pthread.h>

#include<sys/types.h>

#include<stdlib.h>

void enter(int[10][10],int,int);

void display(int[10][10],int,int);

void\* multiplication(void \*);

void enter(int a[10][10],int r,int c)

{

int i,j;

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

{

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

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

}//end of for

}//end of enter()

void display(int a[10][10],int r,int c)

{

int i,j;

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

{

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

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

printf("\n");

}//end of for

}//end of display()

typedef struct mElement

{

int a, b;

}mElement;

void \*multiplication(void \*data)

{

int sum;

mElement \*e2;

e2 = (mElement \*)data;

//e2.a=(mElement)data.a;

//e2.b=(mElement)data.b;

sum = e2->a \* e2->b;

pthread\_exit((void\*)sum);

}

int main()

{

int m1[10][10], m2[10][10], m3[10][10]={0}, r1, c1, r2, c2;

int i, j, k=0, l=0, temp, value\_ptr;

pthread\_t mulId[20];

mElement e1;

printf("\n\*\*\*Program For Matrix Multiplication Using Threads\*\*\n\nEnter Data For Matrix 1:\nNumber Of Rows: ");

scanf("%d",&r1);

printf("Number Of Columns: ");

scanf("%d",&c1);

printf("Elements:\n");

enter(m1, r1, c1);

printf("\nEnter Data For Matrix 2:\nNumber Of Rows: ");

scanf("%d",&r2);

printf("Number Of Columns: ");

scanf("%d",&c2);

printf("Elements:\n");

enter(m2, r2, c2);

printf("\n\*\*Matrices\*\*\nMatrix 1:\n");

display(m1,r1,c1);

printf("\nMatrix 2:\n");

display(m2,r2,c2);

if(c1==r2)

{

for(i=0; i<r1; i++)

{

for(j=0;j<c2;j++)

{

for(l=0;l<c1;l++)

{

e1.a=m1[i][l];

e1.b=m2[l][j];

pthread\_create(&mulId[k], NULL, multiplication ,(void\*)&e1);

k++;

}//End of for of l

}//End Of For of j

}//End Of For of i

k=0;

for(i=0; i<r1; i++)

{

for(j=0;j<c2;j++)

{

temp=0;

for(l=0;l<c1;l++)

{

pthread\_join(mulId[k],(void \*)&value\_ptr);

temp= temp+value\_ptr;

k++;

}//End of for of l

m3[i][j]=temp;

}//End Of For of j

}//End Of For of i

printf("\n\*\*After Multiplication\*\*\n\nMatrix 3:\n");

display(m3, r1, c2);

}//end of if

else

printf("\n\*\*Multiplication is not Possible!!\*\*\n\n");

}

**OUTPUT**

[it@localhost ~]$ gcc MatrixMultiplication.c -lpthread

[it@localhost ~]$ ./a.out

\*\*\*Program For Matrix Multiplication Using Threads\*\*

Enter Data For Matrix 1:

Number Of Rows: 3

Number Of Columns: 2

Elements:

1

2

3

4

5

6

Enter Data For Matrix 2:

Number Of Rows: 2

Number Of Columns: 2

Elements:

1

2

3

4

\*\*Matrices\*\*

Matrix 1:

1 2

3 4

5 6

Matrix 2:

1 2

3 4

\*\*After Multiplication\*\*

Matrix 3:

7 10

15 22

23 34

[it@localhost ~]$ gcc MatrixMultiplication.c -lpthread

[it@localhost ~]$ ./a.out

\*\*\*Program For Matrix Multiplication Using Threads\*\*

Enter Data For Matrix 1:

Number Of Rows: 3

Number Of Columns: 2

Elements:

1

2

3

4

5

6

Enter Data For Matrix 2:

Number Of Rows: 3

Number Of Columns: 2

Elements:

1

2

3

4

5

6

\*\*Matrices\*\*

Matrix 1:

1 2

3 4

5 6

Matrix 2:

1 2

3 4

5 6

\*\*Multiplication is not Possible!!\*\*

[it@localhost ~]$