

**Computer Science and Engineering & Information
Technology (2nd Year B.Tech.)
IIIT Kalyani, West Bengal**

Operating System Lab (CS 411): (Spring: 2019-2020)

Assignment - 6

Marks: 10

Assignment Out: 28th February, 2020

Write a C++ program to compute $A\vec{u}$ to produce \vec{v} , where A is a $m \times n$ matrix, \vec{u} is an n -dimensional vector and \vec{v} is an m -dimensional vector. Following is a guideline to write the program.

1. Reads the number of rows (`row`) and the number of columns (`col`) of a matrix.
2. Defines a 2-D array `mat[row][col]` and two 1-D arrays `inVect[col]` and `outVect[row]` of type `double`.
3. Read the matrix in *row-major order* from `stdin` using the function `void readMat(int row, int col, double m[])`.
Note that `m[]` is of type `double *`. So the $[i][j]$ th element of the array can be accessed as `m[i*col+j]`.
4. The function `void printMat(int row, int col, double m[])` prints the matrix in `stdout`.
5. `void readVect(int dim, double v[])` and `void printVect(int dim, double v[])` reads and prints a vector of dimension `dim`.
6. The function `void matVectMult(int row, int col, double m[], double vIn[], double vOut[])` computes $A\vec{u}$ to \vec{v} , where A is in `m[]`, \vec{u} and \vec{v} are in `vIn[]` and `vOut[]` respectively.
7. The computation of $A\vec{u}$ will be done by *two threads*. The thread t_1 finds the product of row 0 to $m/2 - 1$ and the thread t_2 finds the product of row $m/2$ to $m - 1$. So t_1 applies row $A[0]$ to row $A[m/2 - 1]$ on \vec{u} to produce $\vec{v}[0 \cdots (m/2 - 1)]$; and t_2 applies row $A[m/2]$ to row $A[m - 1]$ on \vec{u} to produce $\vec{v}[m/2 \cdots (m - 1)]$.
8. Both the threads are using the same thread function with different parameters. You may use the following structure to pass a parameter.

```
struct threadParam{
    int row;
    int col;
    double *mat;
    double *vectIn;
    double *vectOut;
};
```

The last parameter of `pthread_create()` is of type `(struct threadParam *)`. Type cast it to `(void *)`.

9. Redirect the data from an input file.

Input:

```
4
3
1 2 3
4 5 6
7 8 9
-1 -2 -3
10 20 30
```

Output:

```
$ ./a.out < data
Enter the number of rows: Enter the number of columns:
Enter matrix data in row order:
Input Matrix is:
1 2 3
4 5 6
7 8 9
-1 -2 -3
Enter 3 dimension vector:
Input vector is:
10 20 30
Transformed vector:
140 320 500 -140
```

Input:

```
1
3
1 2 3
10 20 30
```

```
$ ./a.out < data1
Enter the number of rows: Enter the number of columns:
Enter matrix data in row order:
Input Matrix is:
1 2 3
Enter 3 dimension vector:
Input vector is:
10 20 30
Transformed vector:
140
```