Lab Assignment No: 04

NAME: Vivek Vikram Pundkar ROLLNO: 77

CLASS: C BRANCH: ENTC BATCH: 3

DATE OF PERFORMANCE: 23-09-2021

Multiplication Matrix

STEPS:

- Create a file with extension .c using **gedit matrix.c**
- A new window will pop up and write necessary code.
- Then create a executable file using **gcc -pthread -o 'name of executable file' matrix.c**
- To execute use **./'file name'**

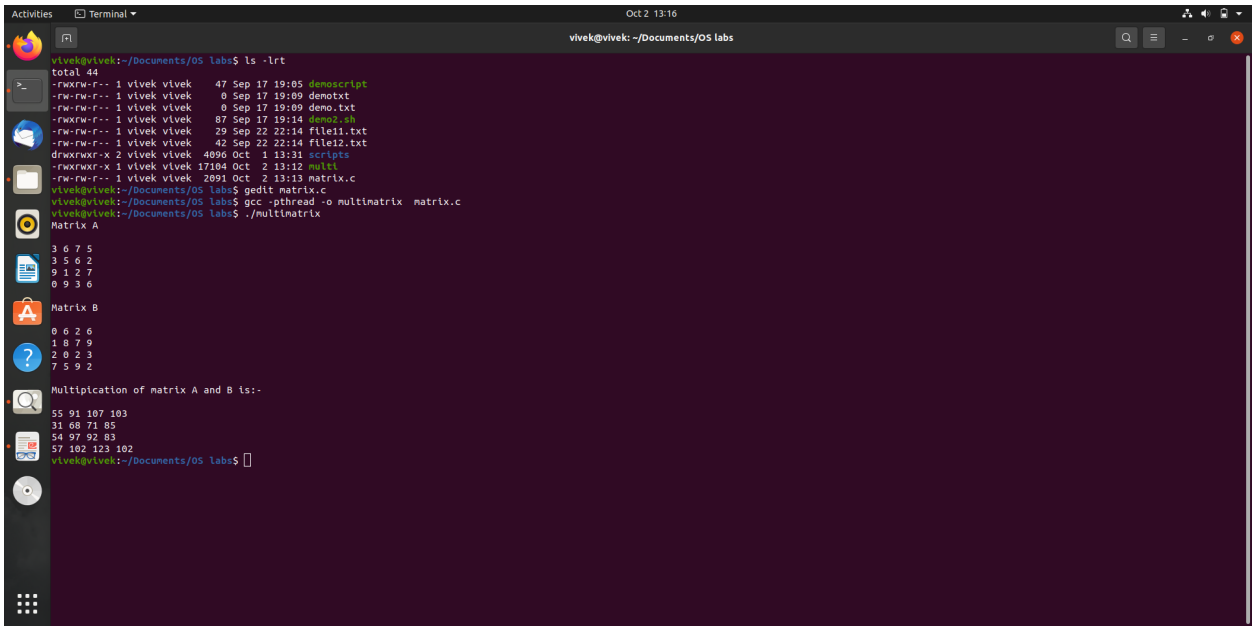
CODE:

```
// C Program to multiply two matrix using pthreads
#include<stdio.h>
#include<pthread.h>
#include<unistd.h>
#include<stdlib.h>
#define MAX 4  // size of matrix


//Each thread computes single element in the resultant matrix
void *mult  (void* arg)
{
   int *data = (int *)arg;
   int k = 0, i = 0;

   int x = data[0];
   for (i = 1; i <= x; i++)
         k += data[i]*data[i+x];

   int *p = (int*)malloc(sizeof(int));
         *p = k;

//Used to terminate a thread and the return value is passed as a pointer
   pthread_exit(p);
}
```

```
//Driver code
int main()
{

    int matA[MAX][MAX];
    int matB[MAX][MAX];


    int r1=MAX,c1=MAX,r2=MAX,c2=MAX,i,j,k;


    // Generating random values in matA
    for (i = 0; i < r1; i++)
                    for (j = 0; j < c1; j++)
                            matA[i][j] = rand() % 10;

            // Generating random values in matB
    for (i = 0; i < r1; i++)
                    for (j = 0; j < c1; j++)
                            matB[i][j] = rand() % 10;

    // Displaying matA
    printf("Matrix A \n\n");
    for (i = 0; i < r1; i++){
            for(j = 0; j < c1; j++)
                    printf("%d ",matA[i][j]);
            printf("\n");
    }

    // Displaying matB
    printf("\nMatrix B \n\n");
    for (i = 0; i < r2; i++){
            for(j = 0; j < c2; j++)
                    printf("%d ",matB[i][j]);
            printf("\n");
    }


    int max = r1*c2;


    //declaring array of threads of size r1*c2
    pthread_t *threads;
    threads = (pthread_t*)malloc(max*sizeof(pthread_t)); // pthread_t data type memory 4 bytes
```

```
    int count = 0;
    int* data = NULL;
    for (i = 0; i < r1; i++)
            for (j = 0; j < c2; j++)
                    {

                    //storing row and column elements in data
                    data = (int *)malloc((20)*sizeof(int));
                    data[0] = c1;

                    for (k = 0; k < c1; k++)
                            data[k+1] = matA[i][k];

                    for (k = 0; k < r2; k++)
                            data[k+c1+1] = matB[k][j];

                    //creating threads
                            pthread_create(&threads[count++], NULL,
                                                    mult, (void*)(data));

                                    }

    printf("\nMultipication of matrix A and B is:- \n\n");
    for (i = 0; i < max; i++)
    {
    void *k;

    //Joining all threads and collecting return value
    pthread_join  (threads[i], &k);


            int *p = (int *)k;
    printf("%d ",*p);
    if ((i + 1) % c2 == 0)
            printf("\n");
    }



return 0;
}
```

OUTPUT:


Activities
Terminal
Oct 2 13:16
vivek@vivek: ~/Documents/OS labs
vivek@vivek:~/Documents/OS labs$ ls -lrt
total 44
-rwxrw-r-- 1 vivek vivek 47 Sep 17 19:05 demoscript
-rw-rw-r-- 1 vivek vivek 0 Sep 17 19:09 demotxt
-rw-rw-r-- 1 vivek vivek 0 Sep 17 19:09 demo.txt
-rwxrw-r-- 1 vivek vivek 87 Sep 17 19:14 demo2.sh
-rw-rw-r-- 1 vivek vivek 29 Sep 22 22:14 file11.txt
-rw-rw-r-- 1 vivek vivek 42 Sep 22 22:14 file12.txt
drwxrwxr-x 2 vivek vivek 4096 Oct 1 13:31 scripts
-rwxrwxr-x 1 vivek vivek 17104 Oct 2 13:12 multi
-rw-rw-r-- 1 vivek vivek 2091 Oct 2 13:13 matrix.c
vivek@vivek:~/Documents/OS labs$ gedit matrix.c
vivek@vivek:~/Documents/OS labs$ gcc -pthread -o multimatrix matrix.c
vivek@vivek:~/Documents/OS labs$ ./multimatrix
Matrix A
3 6 7 5
3 5 6 2
9 1 2 7
0 9 3 6
Matrix B
0 6 2 6
1 8 7 9
2 0 2 3
7 5 9 2
Multipication of matrix A and B is:-
55 91 107 103
31 68 71 85
54 97 92 83
57 102 123 102
vivek@vivek:~/Documents/OS labs$