Abinash Satapathy

Reg. No.: 16BCE0081

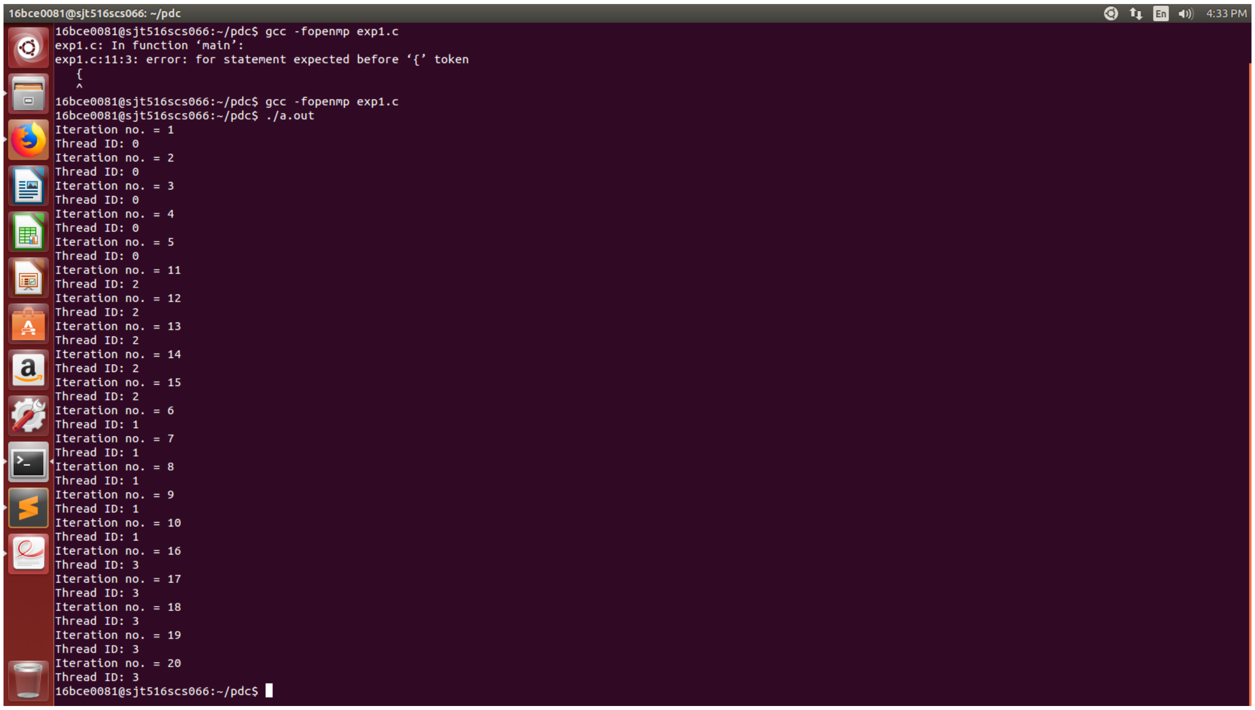
Slot: L45 + L46

Subject: Parallel and Distributed Computing (CSE4001) Lab

Experiment: 2

1. Program to check and print which iteration is done by which thread ID

```
#include <stdio.h>
#include <omp.h>
#include <time.h>

int main(){
        int i = 0;
        // clock_t start_clock = clock();
        #pragma omp parallel
        {
                #pragma omp for
                for(i=0;i<20;i++){
                        printf("Iteration no. = %d\n", i+1);
                        printf("Thread ID: %d\n", omp_get_thread_num());
                }
        }

        return 0;
}
```

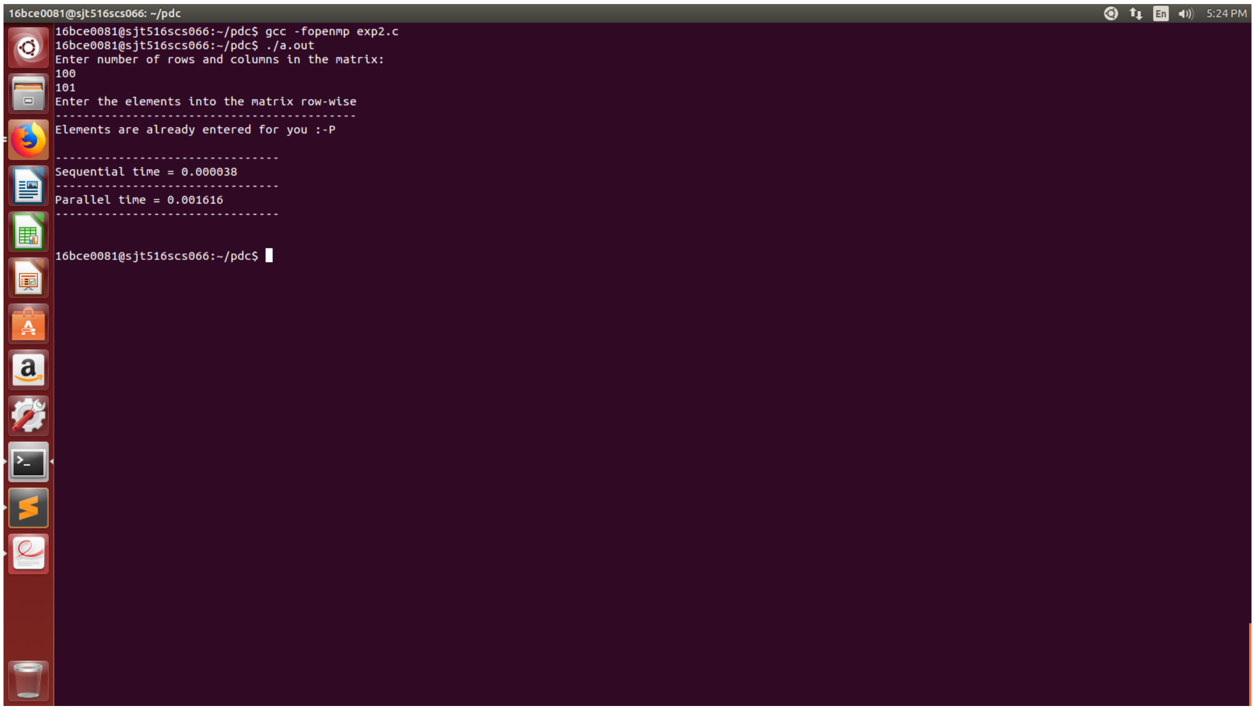


```
16bce0081@sjt516scs066:~/pdc$ gcc -fopenmp exp1.c
exp1.c: In function ‘main’:
exp1.c:11:3: error: for statement expected before ‘{’ token
   {
   ^
16bce0081@sjt516scs066:~/pdc$ gcc -fopenmp exp1.c
16bce0081@sjt516scs066:~/pdc$ ./a.out
Iteration no. = 1
Thread ID: 0
Iteration no. = 2
Thread ID: 0
Iteration no. = 3
Thread ID: 0
Iteration no. = 4
Thread ID: 0
Iteration no. = 5
Thread ID: 0
Iteration no. = 11
Thread ID: 2
Iteration no. = 12
Thread ID: 2
Iteration no. = 13
Thread ID: 2
Iteration no. = 14
Thread ID: 2
Iteration no. = 15
Thread ID: 2
Iteration no. = 6
Thread ID: 1
Iteration no. = 7
Thread ID: 1
Iteration no. = 8
Thread ID: 1
Iteration no. = 9
Thread ID: 1
Iteration no. = 10
Thread ID: 1
Iteration no. = 16
Thread ID: 3
Iteration no. = 17
Thread ID: 3
Iteration no. = 18
Thread ID: 3
Iteration no. = 19
Thread ID: 3
Iteration no. = 20
Thread ID: 3
16bce0081@sjt516scs066:~/pdc$ 
```

2. Program to test execution times of sequential and parallel programming using matrix transpose.

```
#include <stdio.h>
#include <omp.h>


int main(){
        printf("Enter number of rows and columns in the matrix:\n");
        int r, c;
        scanf("%d%d", &r, &c);

        int a[r][c];
        int b[c][r];
        int i, j;

        printf("Enter the elements into the matrix row-wise\n");
        printf("------------------------------------------\n");

        for(i=0;i<r;i++){
                for(j=0;j<c;j++){
                        a[i][j] = i + j;
                }
        }

        printf("Elements are already entered for you :-P \n\n");
```

```
    double start0 = omp_get_wtime();

    for(i=0;i<r;i++){
            for(j=0;j<c;j++){
                b[i][j] = a[j][i];
            }
        }

    double stop0 = omp_get_wtime();

    double start1 = omp_get_wtime();
    #pragma omp parallel
    {
        #pragma omp for
        for(i=0;i<r;i++){
            for(j=0;j<c;j++){
                b[i][j] = a[j][i];
            }
        }
    }

    double stop1 = omp_get_wtime();

    printf("--------------------------------\n");
    printf("Sequential time = %f\n", (stop0-start0));
    printf("--------------------------------\n");
    printf("Parallel time = %f\n", (stop1-start1));
    printf("--------------------------------\n");
    printf("\n\n");

    return 0;
}
```



```
16bce0081@sjt516scs066:~/pdc$ gcc -fopenmp exp2.c
16bce0081@sjt516scs066:~/pdc$ ./a.out
Enter number of rows and columns in the matrix:
100
101
Enter the elements into the matrix row-wise
-------------------------------------------
Elements are already entered for you :-P

--------------------------------
Sequential time = 0.000038
--------------------------------
Parallel time = 0.001616
--------------------------------


16bce0081@sjt516scs066:~/pdc$ 
```