**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);

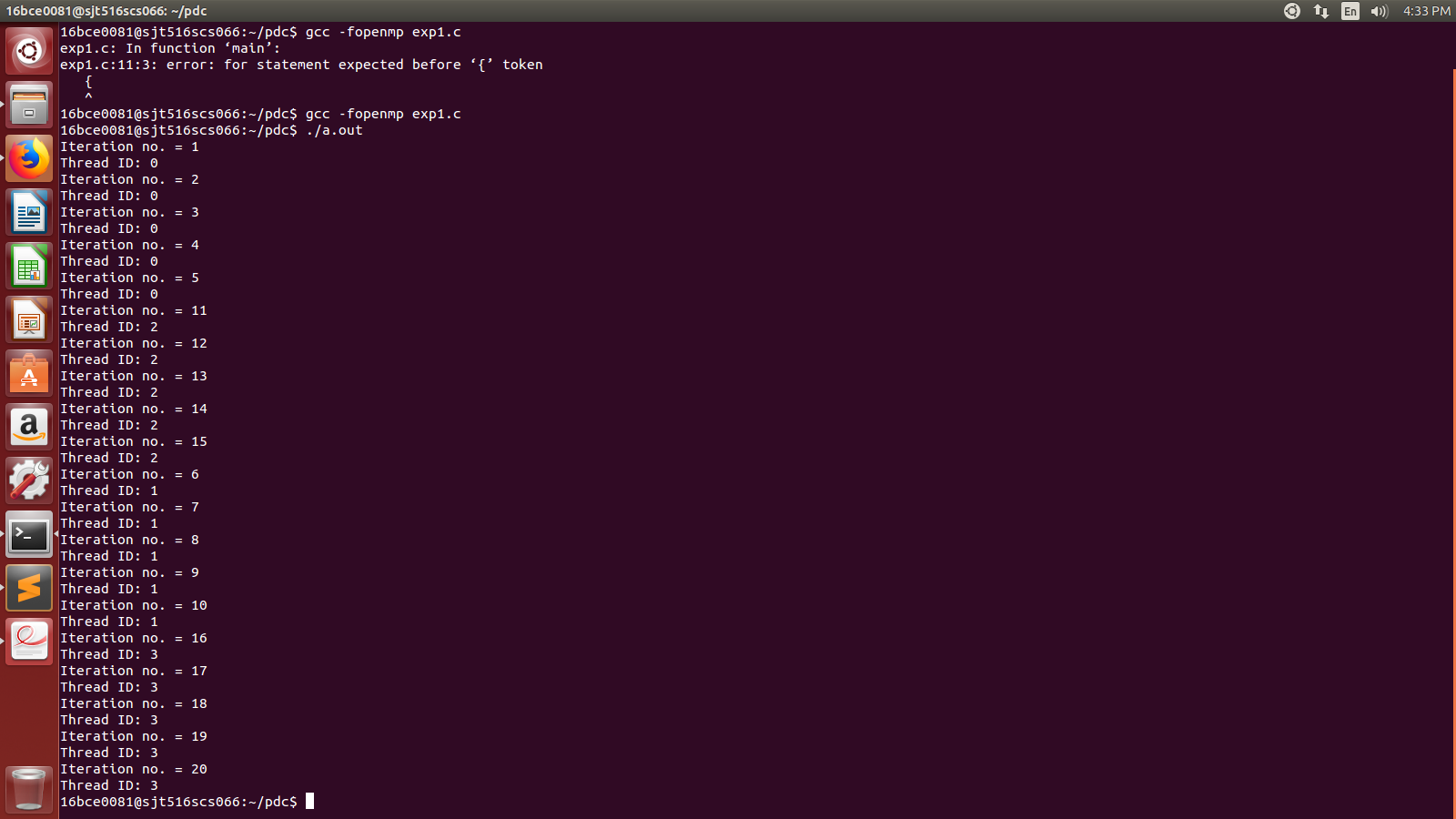
printf("Thread ID: %d\n", omp\_get\_thread\_num());

}

}

return 0;

}



1. 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;

}

