# LAB FAT

## 16BCE2205 JACOB JOHN

### Question 4

#### Source Code

#include <stdio.h>

#include <omp.h>

int tid;

int main()

{

int j, k=0,numt,i=1;

omp\_set\_num\_threads(4);

//parallelize

#pragma omp parallel

{

for(j = 0; j < 10; j++);

//explicit wait to prevent race conditions

k = k + i;

tid = omp\_get\_thread\_num();

if(tid == 0)

{

printf("Hello World! This is thread %d\n",tid);

}

numt = omp\_get\_num\_threads();

}

//introduce a synchronization point for programs

#pragma omp barrier

printf("%d was caculated by %d threads\n",k,numt);

omp\_set\_num\_threads(4);

//get threads that executed the program

#pragma omp parallel default(shared)

{

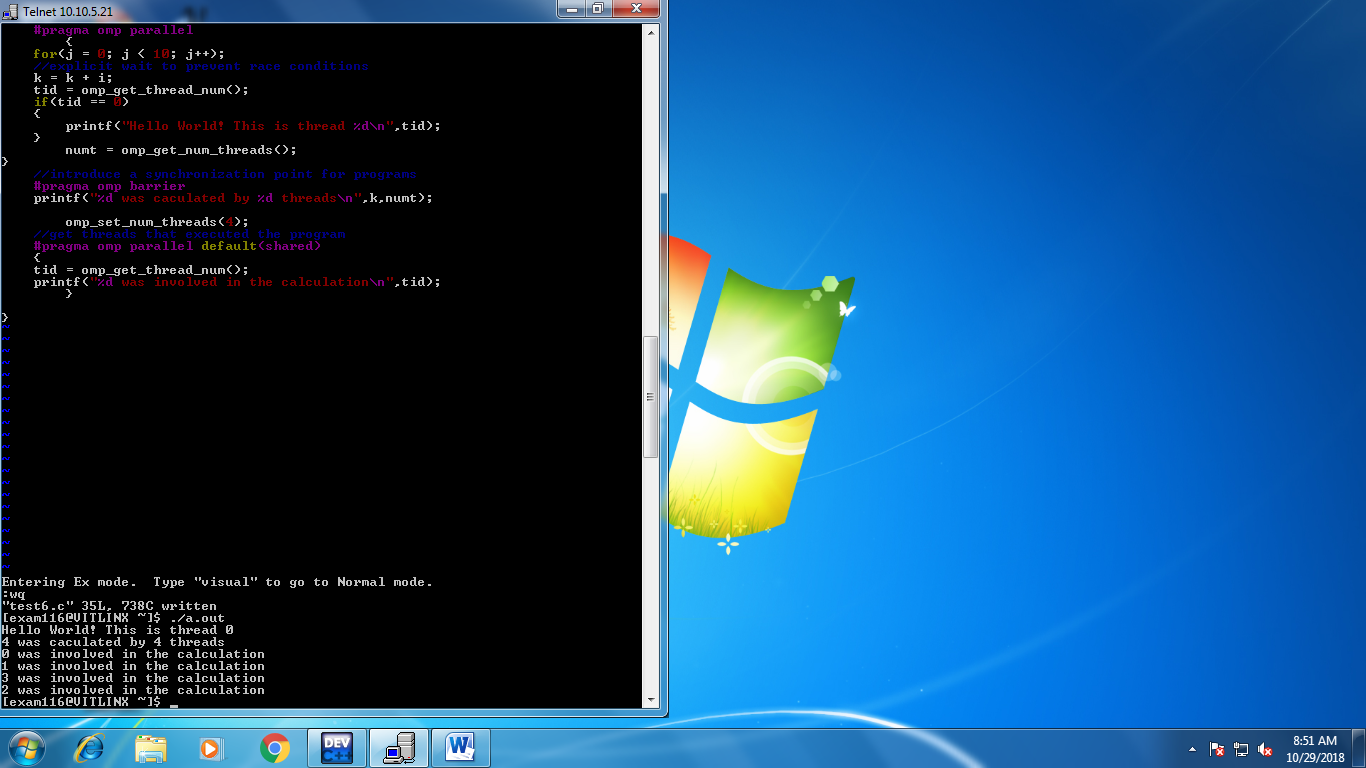
tid = omp\_get\_thread\_num();

printf("%d was involved in the calculation\n",tid);

}

}

#### Output



[exam116@VITLINX ~]$ ./a.out

Hello World! This is thread 0

4 was caculated by 4 threads

0 was involved in the calculation

1 was involved in the calculation

3 was involved in the calculation

2 was involved in the calculation