DEPARTMENT OF INFORMATION TECHNOLOGY, NITK SURATHKAL

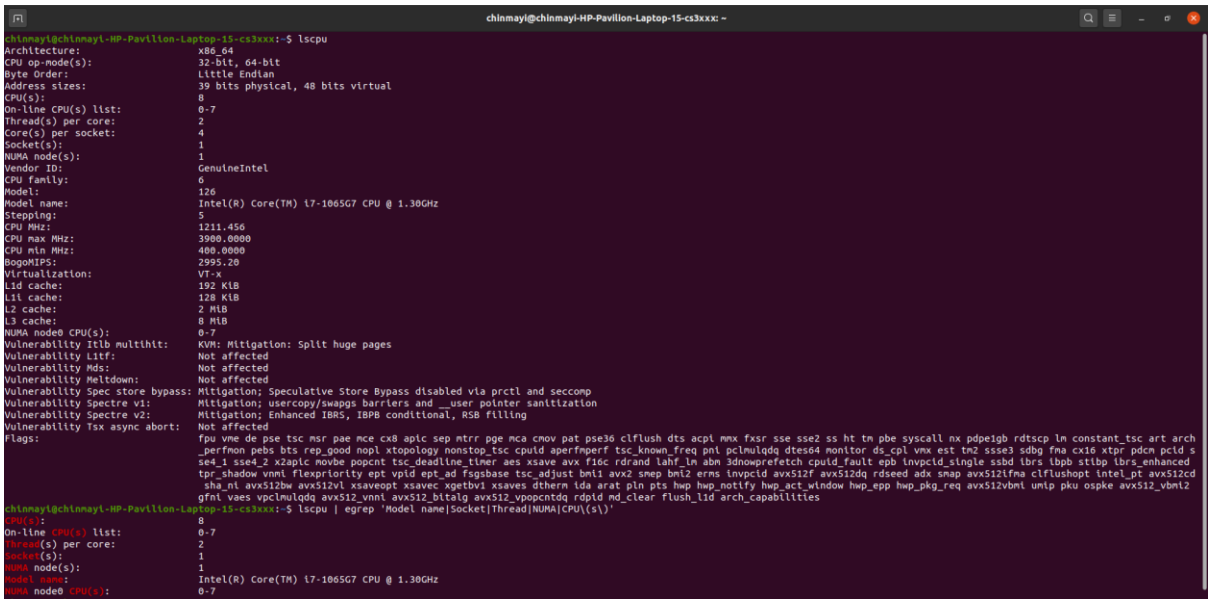
Parallel Computing

LAB 1

Name: Chinmayi C. Ramakrishna

Roll No: 181IT113

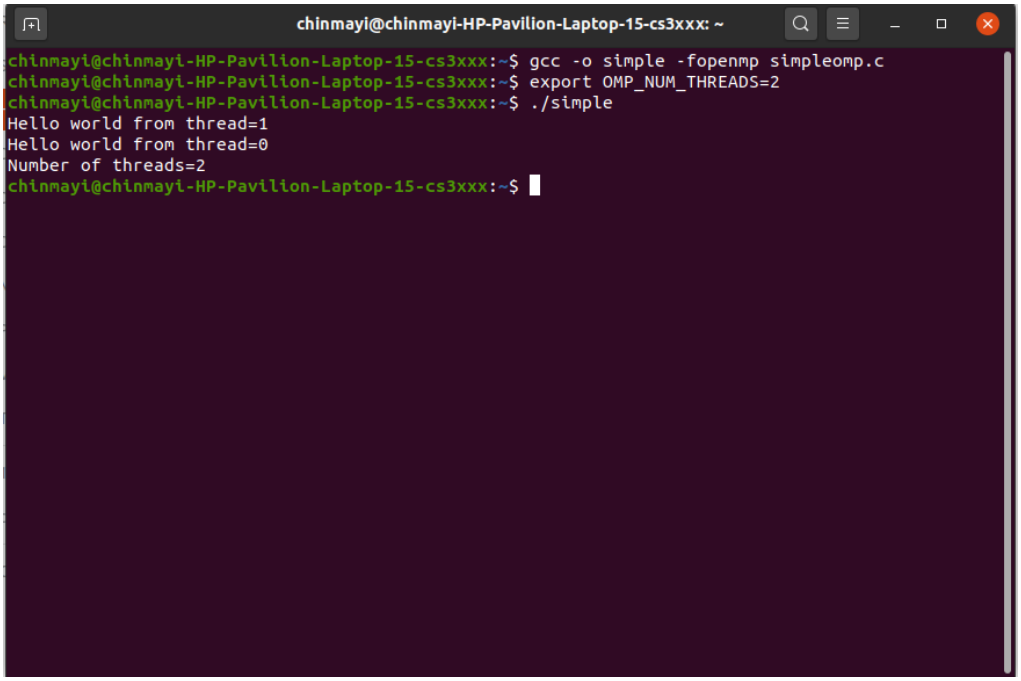
CPU Details:




```
chinmayi@chinmayi-HP-Pavilion-Laptop-15-cs3xxx:~$ lscpu -p
# The following is the parsable format, which can be fed to other
# programs. Each different item in every column has an unique ID
# starting from zero.
# CPU,Core,Socket,Node,,L1d,L1i,L2,L3
0,0,0,0,,0,0,0,0
1,1,0,0,,1,1,1,0
2,2,0,0,,2,2,2,0
3,3,0,0,,3,3,3,0
4,0,0,0,,0,0,0,0
5,1,0,0,,1,1,1,0
6,2,0,0,,2,2,2,0
7,3,0,0,,3,3,3,0
chinmayi@chinmayi-HP-Pavilion-Laptop-15-cs3xxx:~$ nproc --all
8
chinmayi@chinmayi-HP-Pavilion-Laptop-15-cs3xxx:~$ echo "Threads/core: $(nproc --all)"
Threads/core: 8
chinmayi@chinmayi-HP-Pavilion-Laptop-15-cs3xxx:~$
```

1.Write a C/C++ simple parallel program to display the *thread_id* and total number of threads.

a. **Using OMP_NUM_THREADS**



b. Using num_threads() method

simpleomp.c | ifparallel.c | num_threads.c | addarray.c

```c
/*simpleomp.c*/
#include<omp.h>
#include<stdio.h>
int main(){
int nthreads,tid;
#pragma omp parallel private(tid) num_threads(2)
{
tid=omp_get_thread_num();
printf("Hello world from thread=%d\n",tid);
if(tid==0)
{
    nthreads=omp_get_num_threads();
    printf("Number of threads=%d\n",nthreads);
}
}
}

```

Output:

```
chinmayi@chinmayi-HP-Pavilion-Laptop-15-cs3xxx: ~
chinmayi@chinmayi-HP-Pavilion-Laptop-15-cs3xxx:~$ gcc -o simple -fopenmp simpleomp.c
chinmayi@chinmayi-HP-Pavilion-Laptop-15-cs3xxx:~$ ./simple
Hello world from thread=0
Number of threads=2
Hello world from thread=1
chinmayi@chinmayi-HP-Pavilion-Laptop-15-cs3xxx:~$
```

c. **Using omp_set_num_threads() Method:**

simpleomp.c | ifparallel.c | num_threads.c | addarray.

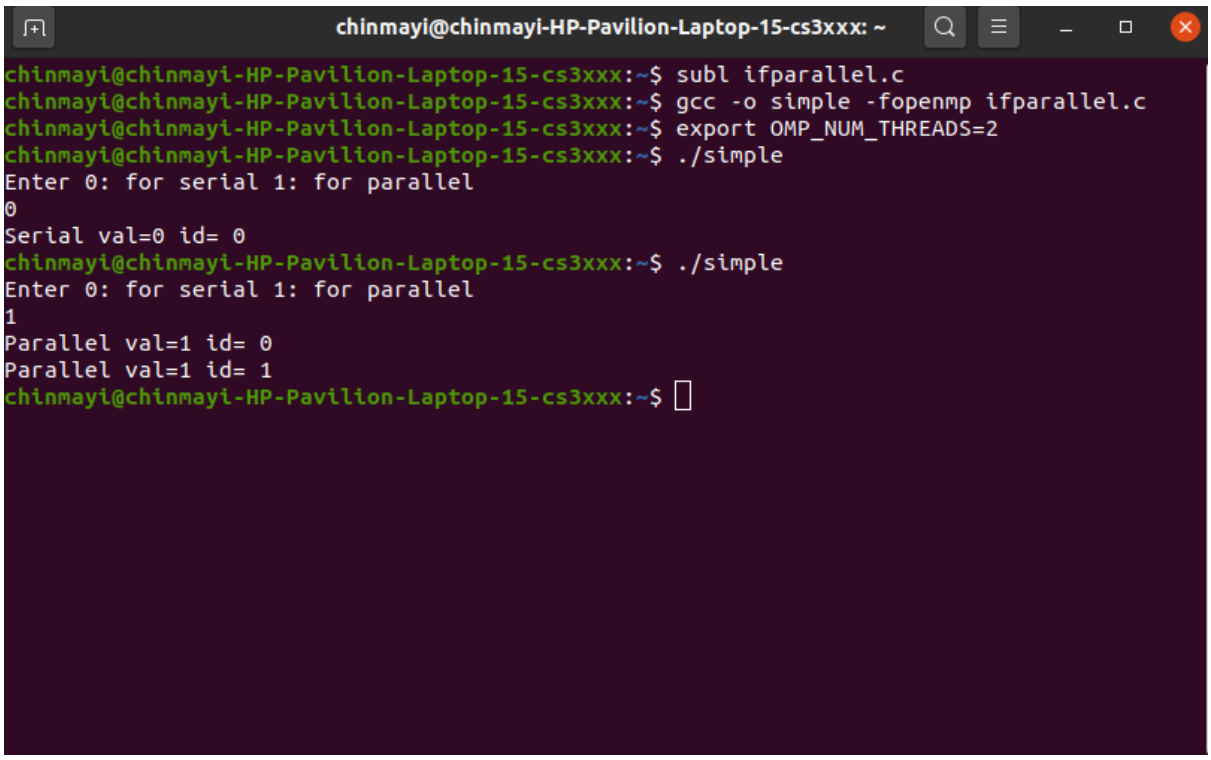
```c
/*simpleomp.c*/
#include<omp.h>
#include<stdio.h>
int main(){
int nthreads,tid;
opm_set_num_threads(2);
#pragma omp parallel private(tid)
{
tid=omp_get_thread_num();
printf("Hello world from thread=%d\n",tid);
if(tid==0)
{
    nthreads=omp_get_num_threads();
    printf("Number of threads=%d\n",nthreads);
}
}
}

```

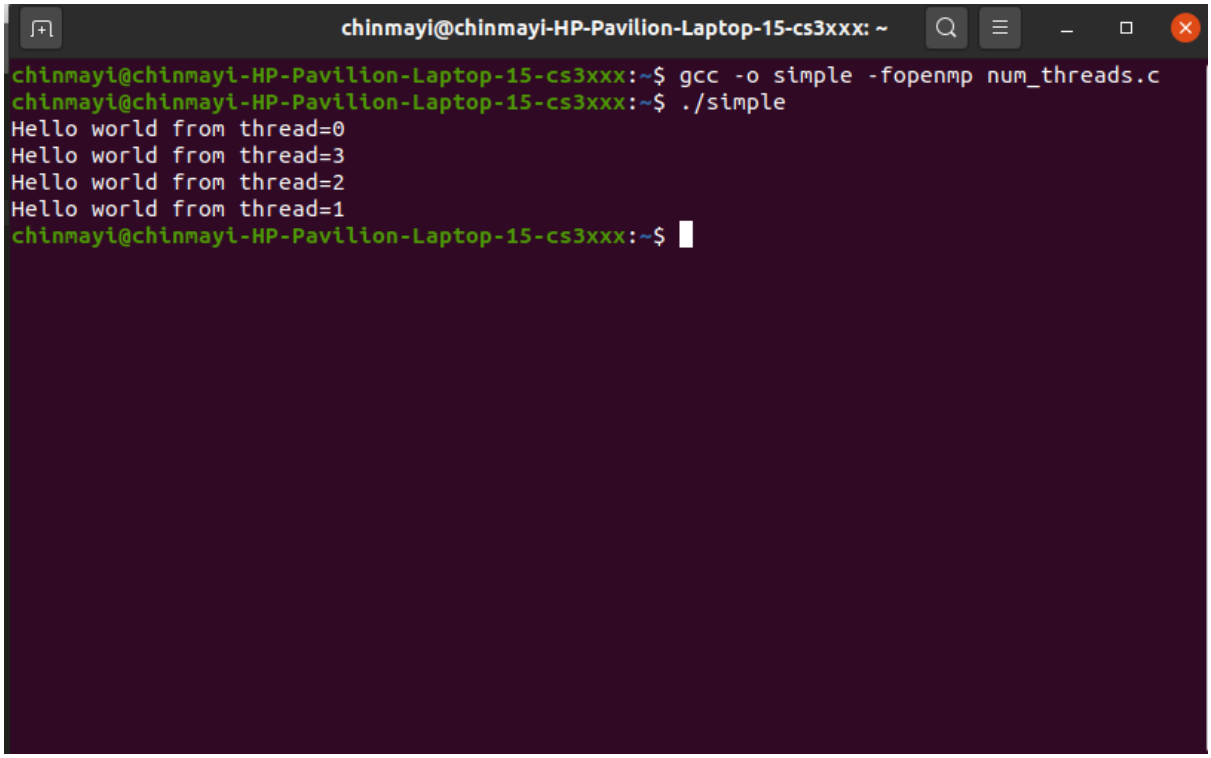
Output:

```
chinmayi@chinmayi-HP-Pavilion-Laptop-15-cs3xxx: ~
chinmayi@chinmayi-HP-Pavilion-Laptop-15-cs3xxx:~$ gcc -o simple -fopenmp simpleomp.c
chinmayi@chinmayi-HP-Pavilion-Laptop-15-cs3xxx:~$ ./simple
Hello world from thread=0
Number of threads=2
Hello world from thread=1
chinmayi@chinmayi-HP-Pavilion-Laptop-15-cs3xxx:~$
```

2. ifparallel.c



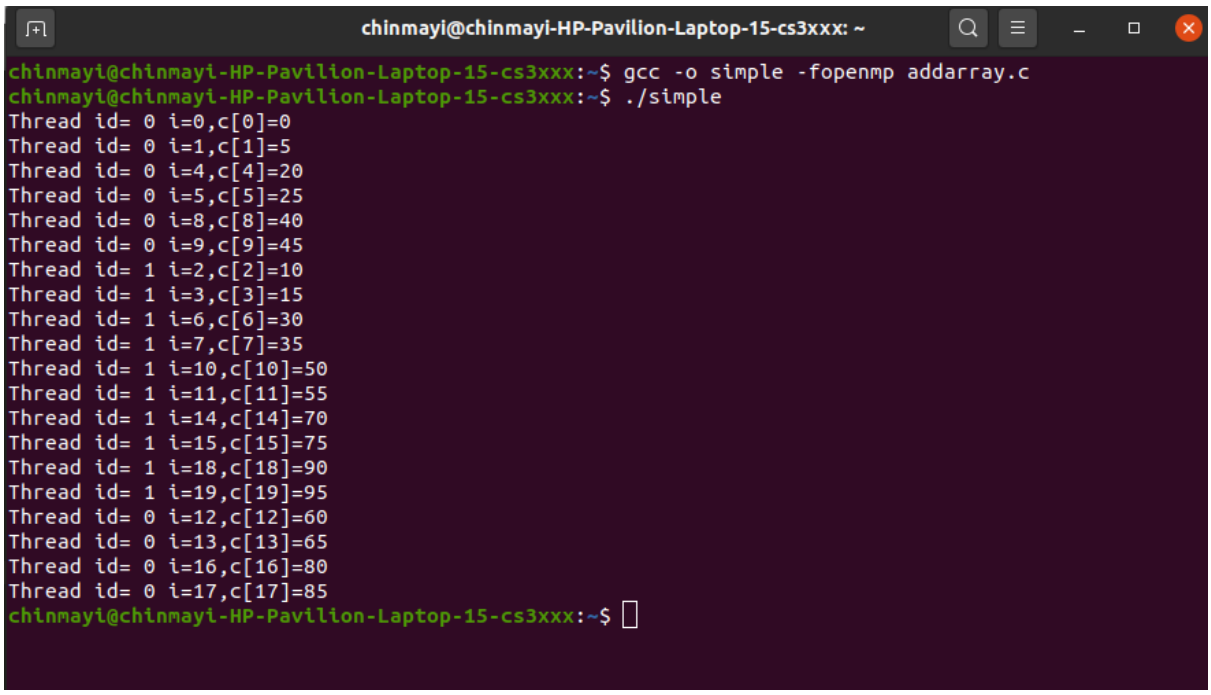
3. num_threads.c



4.Write a C/C++ parallel program for adding corresponding elements of two arrays.

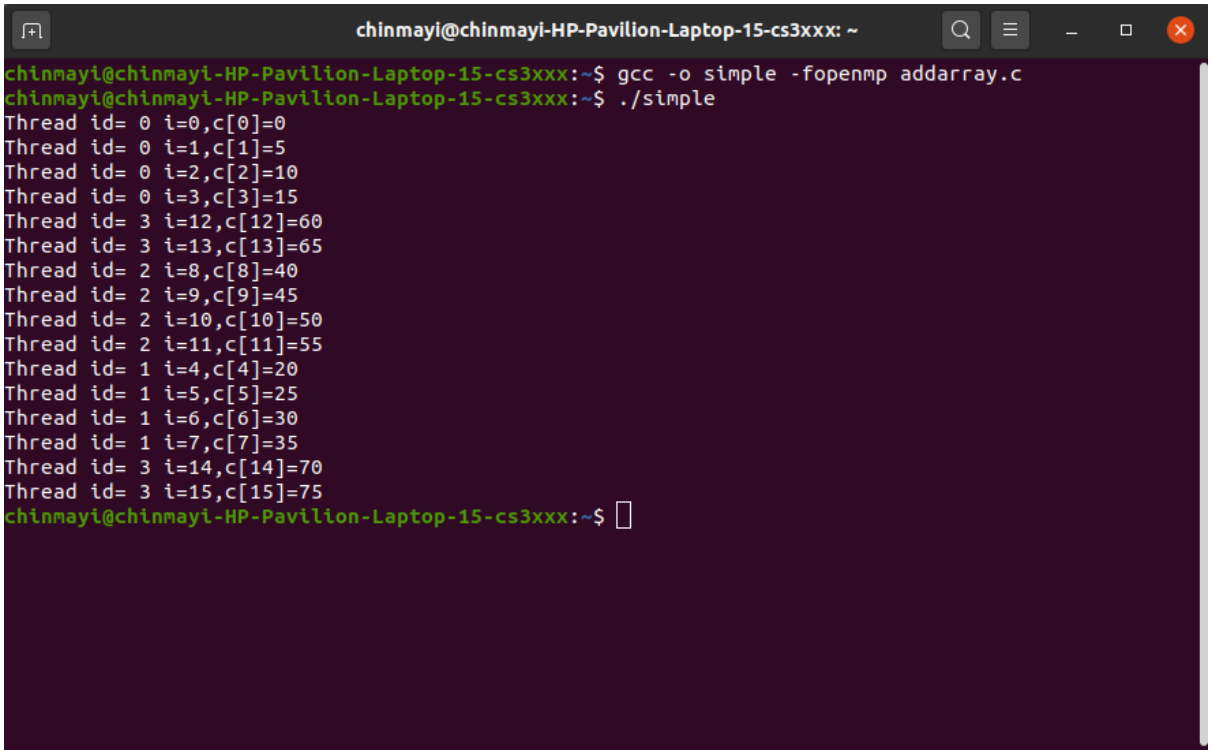
a. Chunk=2 n=20

Each thread id has 10 i values.



b. Chunk= 4 n=16

Each thread id has 4 i values.



c. Chunk = 5 n= 15

Each thread id has 5 i values.