

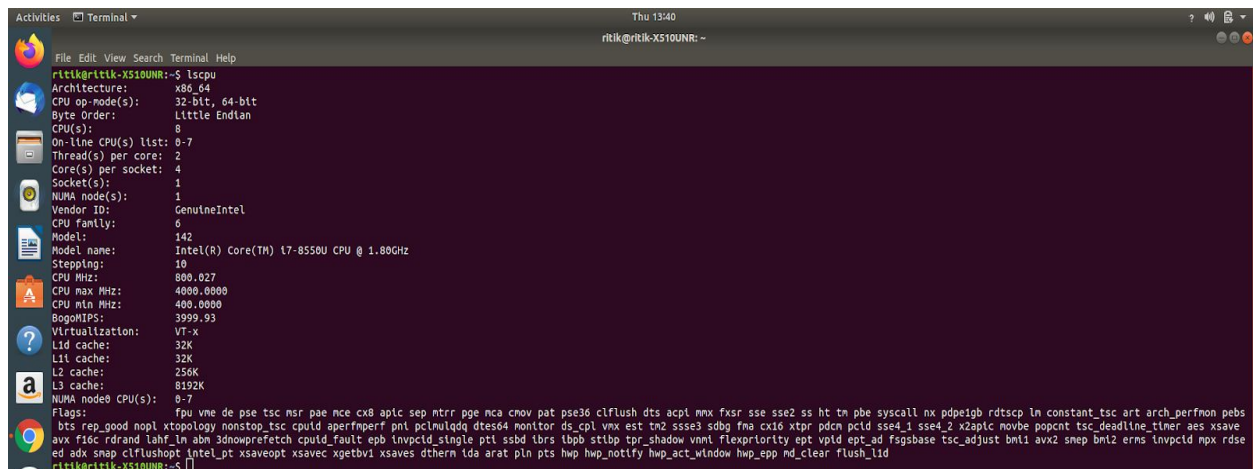
# DEPARTMENT OF INFORMATION TECHNOLOGY, NITK SURATHKAL

## Parallel Programming

### LAB 1 -3<sup>rd</sup> August 2020

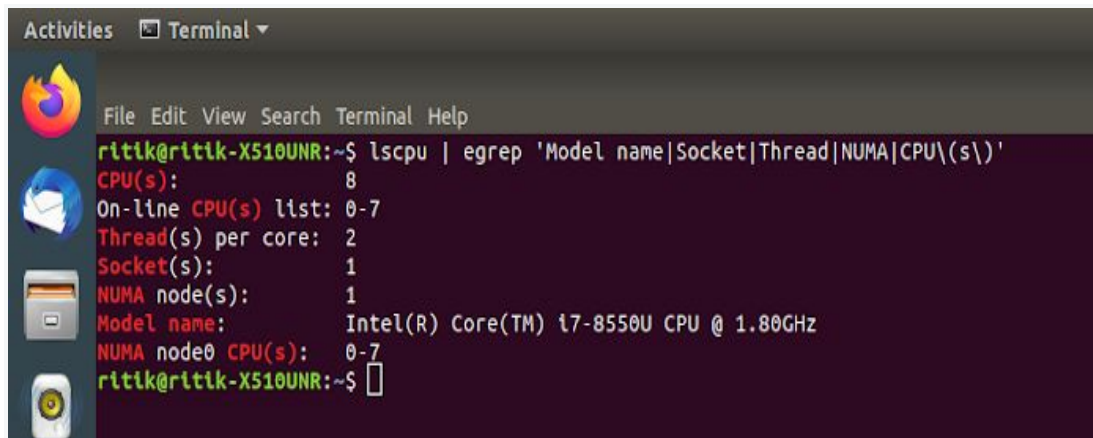
Finding number of CPU s in system

lscpu command



```
ritik@ritik-XS10UNR:~$ lscpu
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
CPU(s): 8
On-line CPU(s) list: 0-7
Thread(s) per core: 2
Core(s) per socket: 4
Socket(s): 1
NUMA node(s): 1
Vendor ID: GenuineIntel
CPU family: 6
Model: 142
Model name: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz
Stepping: 10
CPU MHz: 800.027
CPU max MHz: 4000.0000
CPU min MHz: 400.0000
BogoMIPS: 3999.93
Virtualization: VT-x
L1d cache: 32K
L1i cache: 32K
L2 cache: 256K
L3 cache: 8192K
NUMA node0 CPU(s): 0-7
Flags: fpu vme de pse tsc nsr pae mce cx8 apic sep mtrr pge nca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp ln constant_tsc art arch_perfmon pebs
bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave
avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb invpcid_single pti ssbd tbrs tbbp stibp cpr_shadow vnni flexpriority ept vpid ept_ad fsgsbase tsc_adjust bmt1 avx2 snep bmt2 erns invpcid npx rdse
ed adx snap clflushopt intel_pt xsaveopt xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp_notify hwp_act_window hwp_epp rd_clear flush_lid
```

```
lscpu | egrep 'Model name|Socket|Thread|NUMA|CPU(s) '
```



```
ritik@ritik-XS10UNR:~$ lscpu | egrep 'Model name|Socket|Thread|NUMA|CPU(s) '
CPU(s): 8
On-line CPU(s) list: 0-7
Thread(s) per core: 2
Socket(s): 1
NUMA node(s): 1
Model name: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz
NUMA node0 CPU(s): 0-7
ritik@ritik-XS10UNR:~$
```

lscpu -p

```
Activities Terminal
File Edit View Search Terminal Help
ritik@ritik-X510UNR:~$ 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
ritik@ritik-X510UNR:~$
```

Top

```
Activities Terminal
Thu 13:43
ritik@ritik-X510UNR:~
Tasks: 354 total, 1 running, 269 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.8 us, 0.5 sy, 0.0 ni, 98.4 id, 0.2 wa, 0.0 hi, 0.1 si, 0.0 st
Kib Mem : 8036120 total, 1589684 free, 3035212 used, 3411224 buff/cache
Kib Swap: 2097148 total, 2018004 free, 79144 used, 3657980 avail Mem

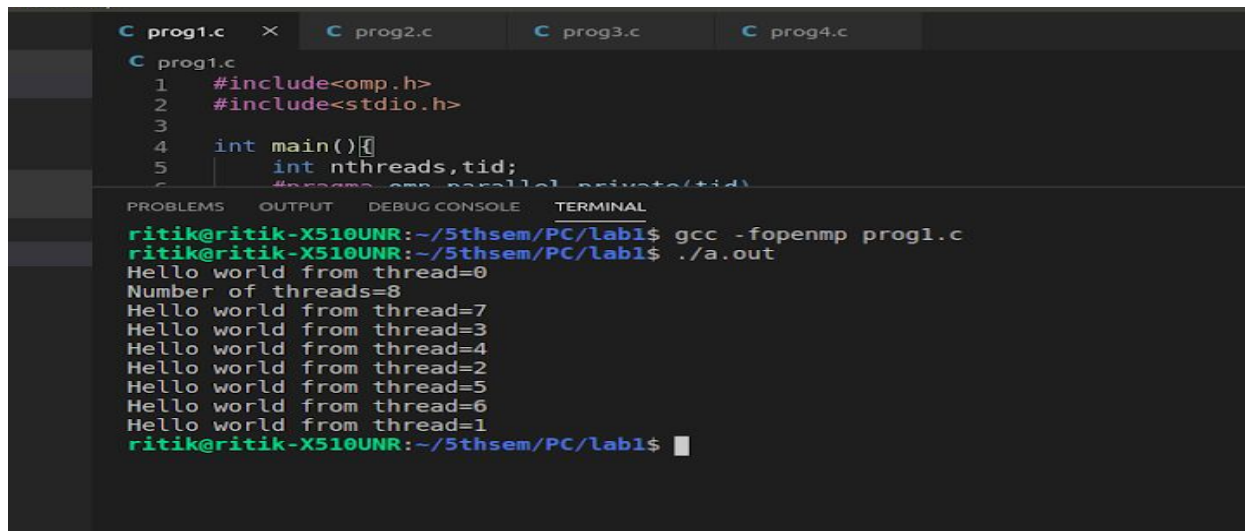
  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 3279 ritik     20   0 1068452 109936 69836 S   4.9   1.4   14:52.89 Xorg
25817 ritik    20   0 482488 37876 28884 S   4.3   0.5   0:00.91 gnome-terminal-
3470 ritik    20   0 4251968 444236 134690 S   3.6   5.5   15:58.45 gnome-shell
2117 gdm       20   0 3742844 152044 98410 S   0.7   1.9   0:33.25 gnome-shell
3454 ritik    20   0 220788 6632 5924 S   0.7   0.1   0:04.91 at-spi2-registr
23592 ritik    20   0 581660 64260 46476 S   0.7   0.8   0:00.40 code
26212 ritik    20   0 51452 4088 3332 R   0.7   0.1   0:00.06 top
  1 root      20   0 225604 9032 6580 S   0.3   0.1   0:29.66 systemd
1230 root      20   0 513272 18484 14424 S   0.3   0.2   0:06.38 NetworkManager
1719 nmopdb     20   0 1010276 69668 34112 S   0.3   0.9   1:15.03 nmopdb
2262 root      20   0 1717396 95116 48824 S   0.3   1.2   2:30.21 dockerd
3883 ritik    20   0 1809708 548044 380036 S   0.3   6.8   13:23.99 chrome
4397 ritik    20   0 4808888 88796 62392 S   0.3   1.0   0:04.68 chrome
4465 ritik    20   0 4808888 81988 62384 S   0.3   1.0   0:04.67 chrome
4497 ritik    20   0 4808888 84412 65076 S   0.3   1.1   0:04.72 chrome
4525 ritik    20   0 4809912 85448 64672 S   0.3   1.1   0:04.84 chrome
23551 root      20   0 0 0 0 S   0.3   0.0   0:02.00 kworker/u16:0-e
23613 ritik    20   0 3772092 92788 56616 S   0.3   1.2   0:05.72 vls-agent
23897 ritik    20   0 4975744 243588 122580 S   0.3   3.0   0:35.89 chrome
  2 root      20   0 0 0 0 S   0.0  0.0   0:00.03 kthreadd
  3 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 rcu_gp
  4 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 rcu_par_gp
  9 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 kworker/0:0H-kb
10 root      20   0 0 0 0 S   0.0  0.0   0:00.00 rw_percpu_wq
11 root      20   0 0 0 0 S   0.0  0.0   0:01.00 ksoftirqd/0
12 root      rt  0 0 0 0 S   0.0  0.0   0:29.89 rcu_sched
13 root      0 -20 0 0 0 S   0.0  0.0   0:00.10 migration/0
14 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 idle_inject/0
15 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 cpuhp/0
16 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 cpuhp/1
17 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 kworker/1:0H-kb
18 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 cpuhp/3
19 root      0 -20 0 0 0 S   0.0  0.0   0:00.02 ksoftirqd/1
21 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 cpuhp/2
22 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 idle_inject/2
23 root      0 -20 0 0 0 S   0.0  0.0   0:00.10 migration/2
24 root      0 -20 0 0 0 S   0.0  0.0   0:00.21 ksoftirqd/2
26 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 kworker/2:0H-kb
27 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 cpuhp/3
28 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 idle_inject/3
29 root      0 -20 0 0 0 S   0.0  0.0   0:00.08 migration/3
30 root      0 -20 0 0 0 S   0.0  0.0   0:00.28 ksoftirqd/3
32 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 kworker/3:0H-kb
33 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 cpuhp/4
34 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 idle_inject/4
35 root      0 -20 0 0 0 S   0.0  0.0   0:00.08 migration/4
36 root      0 -20 0 0 0 S   0.0  0.0   0:00.22 ksoftirqd/4
39 root      0 -20 0 0 0 S   0.0  0.0   0:00.00 cpuhp/5
ritik@ritik-X510UNR:~$
```

```
Activities Terminal
File Edit View Search Terminal Help
ritik@ritik-X510UNR:~$ nproc --all
8
ritik@ritik-X510UNR:~$ echo "Threads/core: $(nproc --all)"
Threads/core: 8
ritik@ritik-X510UNR:~$
```

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

```
#include<omp.h>
#include<stdio.h>
int main(){
    int nthreads,tid;
    #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);
        }
    }
}
```

o/p:



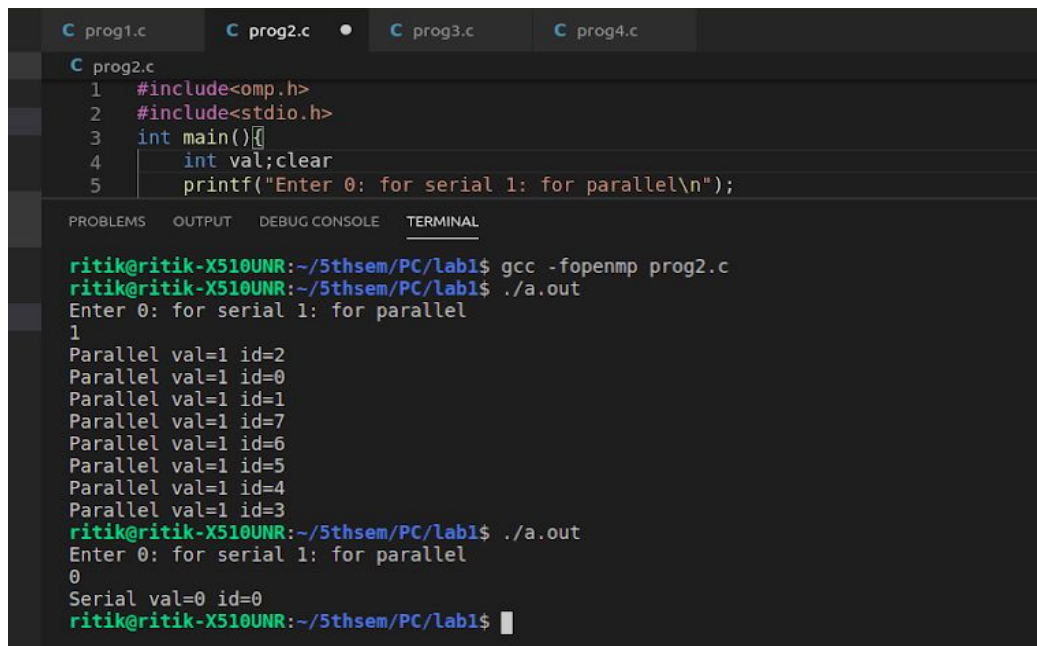
```
C prog1.c x C prog2.c C prog3.c C prog4.c
C prog1.c
1 #include<omp.h>
2 #include<stdio.h>
3
4 int main(){
5     int nthreads,tid;
6     #pragma omp parallel private(tid)
7
8     tid=omp_get_thread_num();
9     printf("Hello world from thread=%d\n",tid);
10    if(tid==0){
11        nthreads=omp_get_num_threads();
12        printf("Number of threads=%d\n",nthreads);
13    }
14 }
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
ritik@ritik-X510UNR:~/5thsem/PC/lab1$ gcc -fopenmp prog1.c
ritik@ritik-X510UNR:~/5thsem/PC/lab1$ ./a.out
Hello world from thread=0
Number of threads=8
Hello world from thread=7
Hello world from thread=3
Hello world from thread=4
Hello world from thread=2
Hello world from thread=5
Hello world from thread=6
Hello world from thread=1
ritik@ritik-X510UNR:~/5thsem/PC/lab1$
```

Here when we will change thread num no. of thread o/p changes accordingly.(default=8)

2. Check the output of following program:

```
#include<omp.h>
#include<stdio.h>
int main() {
    int val;
    printf("Enter 0: for serial 1: for parallel\n");
    scanf("%d",&val);
    #pragma omp parallel if(val)
    if(omp_in_parallel()) printf("Parallel val=%d id=%d\n",val,
omp_get_thread_num());
    else printf("Serial val=%d id=%d\n",val, omp_get_thread_num());
}
```

o/p:



```
C prog1.c  C prog2.c  C prog3.c  C prog4.c
C prog2.c
1  #include<omp.h>
2  #include<stdio.h>
3  int main(){
4      int val;clear
5      printf("Enter 0: for serial 1: for parallel\n");

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
ritik@ritik-X510UNR:~/5thsem/PC/lab1$ gcc -fopenmp prog2.c
ritik@ritik-X510UNR:~/5thsem/PC/lab1$ ./a.out
Enter 0: for serial 1: for parallel
1
Parallel val=1 id=2
Parallel val=1 id=0
Parallel val=1 id=1
Parallel val=1 id=7
Parallel val=1 id=6
Parallel val=1 id=5
Parallel val=1 id=4
Parallel val=1 id=3
ritik@ritik-X510UNR:~/5thsem/PC/lab1$ ./a.out
Enter 0: for serial 1: for parallel
0
Serial val=0 id=0
ritik@ritik-X510UNR:~/5thsem/PC/lab1$
```

Here we have both serial and parallel execution. So serial has 1 thread but parallel has 8 thread as default.

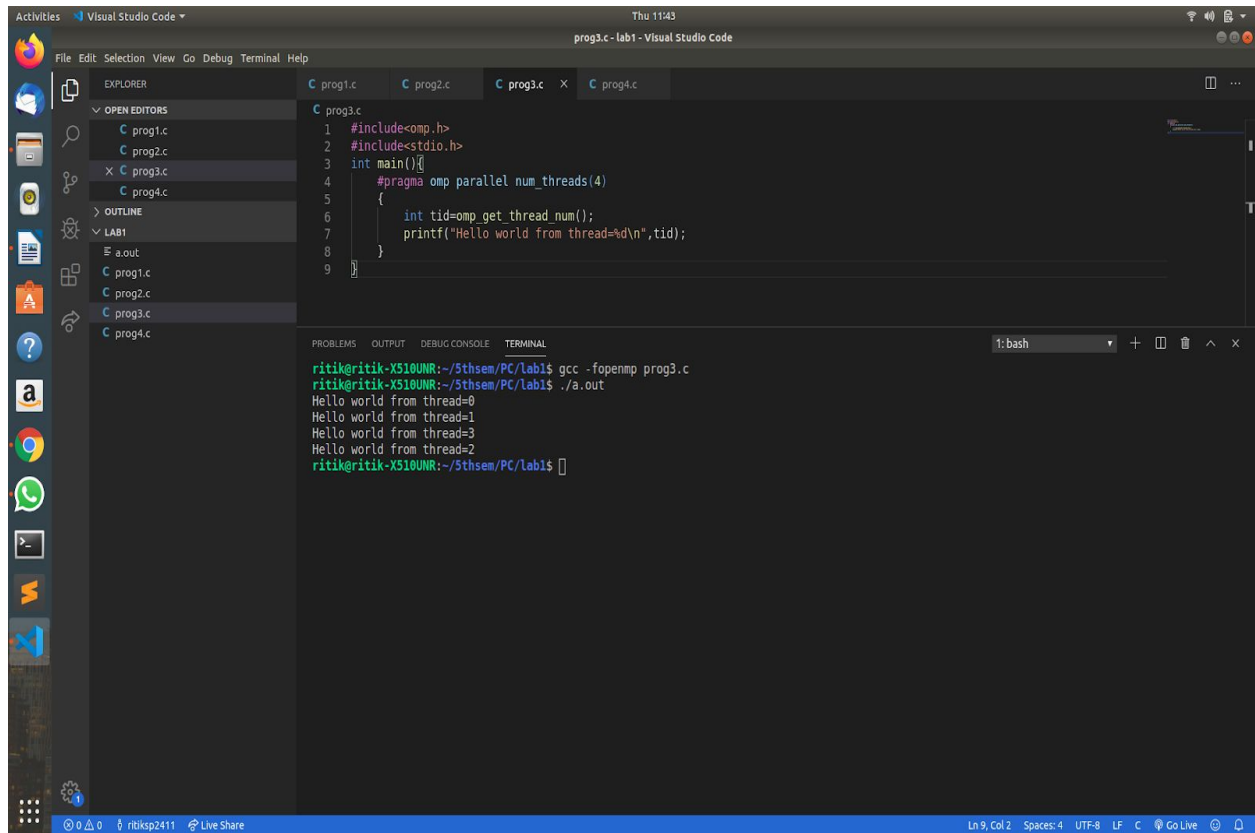
For changing the no. of threads:

export OMP\_NUM\_THREADS=2 (mention it in terminal before running)

num\_threads() (mention it in line where we write #pragma)

omp\_set\_num\_threads() (mention it inside the main code)

3. Observe and record the output of the following program.



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left lists files: prog1.c, prog2.c, prog3.c, and prog4.c. The main editor displays the code for prog3.c:

```
1 #include<omp.h>
2 #include<stdio.h>
3 int main()
4 {
5     #pragma omp parallel num_threads(4)
6     {
7         int tid=omp.get_thread_num();
7         printf("Hello world from thread=%d\n",tid);
8     }
9 }
```

The TERMINAL panel at the bottom shows the execution of the program:

```
ritik@ritik-XS10UNR:~/5thsem/PC/lab1$ gcc -fopenmp prog3.c
ritik@ritik-XS10UNR:~/5thsem/PC/lab1$ ./a.out
Hello world from thread=0
Hello world from thread=1
Hello world from thread=3
Hello world from thread=2
ritik@ritik-XS10UNR:~/5thsem/PC/lab1$
```

The status bar at the bottom indicates the current position is Ln 9, Col 2, with UTF-8 encoding and LF line endings.

Here we have num\_thread(4) so we have only 4 threads .

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

```
#include<omp.h>
#include<stdio.h>
int main() {
    int i,n,chunk;
    int a[20],b[20],c[20];
    n=20;
    chunk=2;
    /*initializing array*/
    for(i=0;i<n;i++)
    {
        a[i]=i*2;
        b[i]=i*3;
    }
    #pragma omp parallel for default(shared) private(i)
    schedule(static,chunk)
    for(i=0;i<n;i++)
    {
        c[i]=a[i]+b[i];
        printf("Thread id= %d i=%d,c[%d]=%d\n",
omp_get_thread_num(),i,i,c[i]);
    }
}
```

o/p

```
ritik@ritik-X510UNR:~/5thsem/PC/lab1$ gcc -fopenmp prog4.c
ritik@ritik-X510UNR:~/5thsem/PC/lab1$ ./a.out
Thread id= 1 i=2,c[2]=10
Thread id= 1 i=3,c[3]=15
Thread id= 1 i=18,c[18]=90
Thread id= 1 i=19,c[19]=95
Thread id= 5 i=10,c[10]=50
Thread id= 5 i=11,c[11]=55
Thread id= 0 i=0,c[0]=0
Thread id= 0 i=1,c[1]=5
Thread id= 0 i=16,c[16]=80
Thread id= 0 i=17,c[17]=85
Thread id= 6 i=12,c[12]=60
Thread id= 6 i=13,c[13]=65
Thread id= 2 i=4,c[4]=20
Thread id= 2 i=5,c[5]=25
Thread id= 4 i=8,c[8]=40
Thread id= 4 i=9,c[9]=45
Thread id= 7 i=14,c[14]=70
Thread id= 7 i=15,c[15]=75
Thread id= 3 i=6,c[6]=30
Thread id= 3 i=7,c[7]=35
ritik@ritik-X510UNR:~/5thsem/PC/lab1$
```

The chunk size is the measure how many iterations should be given to one thread initially. After that if we need more then they are assigned in a round robin manner. Here my PC has 8 thread and chunk size is 2 so 16 out of 20 iterations are assigned. Remaining 4 are assigned in round robin so 0th,1st thread gets 2 more iterations. So if chunk size is 4 only 5 threads are assigned iterations because we need 20 in total. So if chunk size is equal to no. of iterations, it is the same as serial execution. Thus chunk size changes execution changes.

o/p when chunk size is 4 :

```
ritik@ritik-X510UNR:~/5thsem/PC/lab1$ gcc -fopenmp prog4.c
ritik@ritik-X510UNR:~/5thsem/PC/lab1$ ./a.out
Thread id= 0 i=0,c[0]=0
Thread id= 0 i=1,c[1]=5
Thread id= 0 i=2,c[2]=10
Thread id= 0 i=3,c[3]=15
Thread id= 1 i=4,c[4]=20
Thread id= 1 i=5,c[5]=25
Thread id= 1 i=6,c[6]=30
Thread id= 1 i=7,c[7]=35
Thread id= 2 i=8,c[8]=40
Thread id= 2 i=9,c[9]=45
Thread id= 2 i=10,c[10]=50
Thread id= 2 i=11,c[11]=55
Thread id= 3 i=12,c[12]=60
Thread id= 3 i=13,c[13]=65
Thread id= 3 i=14,c[14]=70
Thread id= 3 i=15,c[15]=75
Thread id= 4 i=16,c[16]=80
Thread id= 4 i=17,c[17]=85
Thread id= 4 i=18,c[18]=90
Thread id= 4 i=19,c[19]=95
ritik@ritik-X510UNR:~/5thsem/PC/lab1$
```

-Ritik Pansuriya(181IT237)