DEPARTMENT OF INFORMATION TECHNOLOGY, NITK SURATHKAL

Parallel Programming

NAME:BHUVANESWAR DHARMASIVAM

ROLL NO:191T107

LAB 1 – 27th July 2021

Note:

parallel
Forms a team of threads and starts parallel execution.
#pragma omp parallel *[clause[[,]clause] ...]*
structured-block
clause:
if(*scalar-expression*)
num_threads(*integer-expression*)
default(**shared** | **none**)
private(*list*)
firstprivate(*list*)
shared(*list*)
copyin(*list*)
reduction(*reduction-identifier*: *list*)

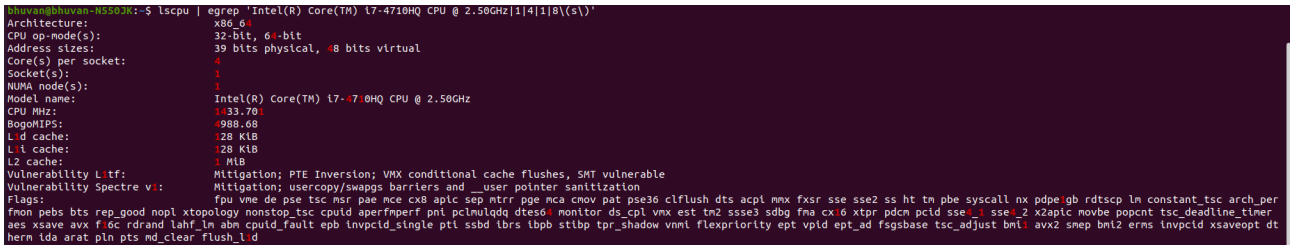
I. Finding number of CPU s in system **[3 Marks]**

a) lscpu command

```
$ lscpu
```

```
bhuvan@bhuvan-N550JK:~$ lscpu
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
Address sizes:                   39 bits physical, 48 bits virtual
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:                           60
Model name:                      Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz
Stepping:                        3
CPU MHz:                         1776.866
CPU max MHz:                     3500.0000
CPU min MHz:                     800.0000
BogoMIPS:                        4988.68
Virtualization:                  VT-x
L1d cache:                       128 KiB
L1i cache:                       128 KiB
L2 cache:                        1 MiB
L3 cache:                        6 MiB
NUMA node0 CPU(s):               0-7
Vulnerability Itlb multihit:     KVM: Mitigation: VMX disabled
Vulnerability L1tf:              Mitigation; PTE Inversion; VMX conditional cach
                                 e flushes, SMT vulnerable
Vulnerability Mds:               Mitigation; Clear CPU buffers; SMT vulnerable
Vulnerability Meltdown:          Mitigation; PTI
Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled v
                                 ia prctl and seccomp
Vulnerability Spectre v1:        Mitigation; usercopy/swapgs barriers and __user
                                  pointer sanitization
Vulnerability Spectre v2:        Mitigation; Full generic retpoline, IBPB condit
                                 ional, IBRS_FW, STIBP conditional, RSB filling
Vulnerability Srbds:             Mitigation; Microcode
Vulnerability Tsx async abort:   Not affected
Flags:                           fpu vme de pse tsc msr pae mce cx8 apic sep mtr
                                 r pge mca cmov pat pse36 clflush dts acpi mmx f
                                 xsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rd
                                 tscp lm constant_tsc arch_perfmon pebs bts rep_
                                 good nopl xtopology nonstop_tsc cpuid aperfmper
                                 f pni pclmulqdq dtes64 monitor ds_cpl vmx est t
                                 m2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 ss
                                 e4_2 x2apic movbe popcnt tsc_deadline_timer aes
                                  xsave avx f16c rdrand lahf_lm abm cpuid_fault
                                 epb invpcid_single pti ssbd ibrs ibpb stibp tpr
                                 _shadow vnmi flexpriority ept vpid ept_ad fsgsb
                                 ase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid
```

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

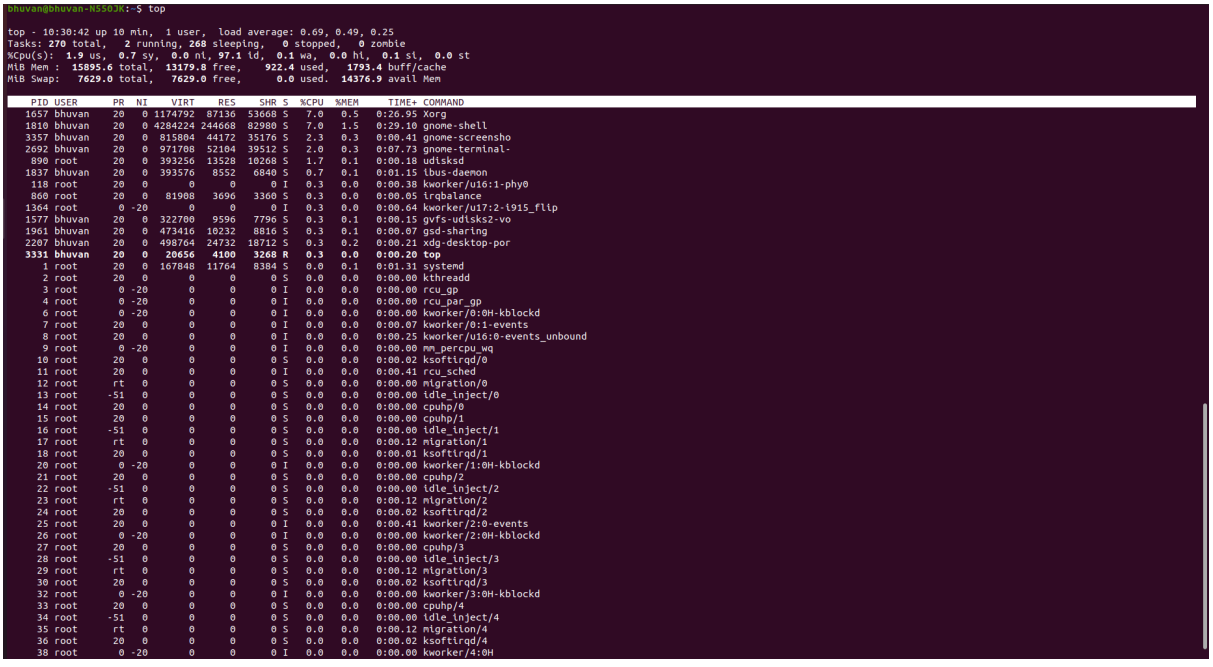



```
$ lscpu -p
```

```
bhuvan@bhuvan-N550JK:~$ 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
```

b) Run top or htop command to obtain the number of CPUs/cores in linux

```
$top
```



c) Execute nproc and print the number of CPUs available on Linux

```
$ nproc –all
```

```
bhuvan@bhuvan-N550JK:~$ echo "Threads/core: $(nproc --all)"
Threads/core: 8
```

```
$ echo "Threads/core: $(nproc –all)"
```

```
bhuvan@bhuvan-N550JK:~$ echo "Threads/core: $(nproc --all)"
Threads/core: 8
```

2.Write a C/C++ simple parallel program to display the *thread_id* and total number of threads. [3 Marks]

```
/*simpleomp.c*/
#include<omp.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);
}
}
}
```

Execute the program as follows:

$gcc –o simple –fopenmp simpleomp.c

$export OMP_NUM_THREADS=2

$./simple

Note down the output in your observation book.

Number of threads in a parallel region is determined by the *if* clause, *num_threads(), omp_set_num_threads(), OMP_NUM_THREADS.*

Use these various methods to set number of threads and mention the method of setting the same.

```
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ gcc -o simple -fopenmp simpleomp.c
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ export OMP_NUM_THREADS=2
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ ./simple
Hello world from thread=0
Number of threads=2
Hello world from thread=1
```

If clause

```
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ gcc -o if -fopenmp ifclause.c
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ ./if
Enter 0: for serial 1: for parallel
0
Serial val=0 id= 0
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ ./if
Enter 0: for serial 1: for parallel
1
Parallel val=1 id= 4
Parallel val=1 id= 0
Parallel val=1 id= 7
Parallel val=1 id= 5
Parallel val=1 id= 3
Parallel val=1 id= 6
Parallel val=1 id= 1
Parallel val=1 id= 2
```

num_threads()

```
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ gcc -o threads -fopenmp num_threads.c
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ ./threads
Hello world from thread=2
Hello world from thread=3
Hello world from thread=0
Hello world from thread=4
Hello world from thread=5
Hello world from thread=1
```

OMP_NUM_THREADS

```
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ gcc -o simple -fopenmp simpleomp.c
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ export OMP_NUM_THREADS=2
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ ./simple
Hello world from thread=0
Number of threads=2
Hello world from thread=1
```

omp_set_num_threads()

```
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ gcc -o simple -fopenmp simpleomp2.c
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ ./simple
Hello world from thread=0
Number of threads=8
Hello world from thread=3
Hello world from thread=4
Hello world from thread=7
Hello world from thread=2
Hello world from thread=1
Hello world from thread=5
Hello world from thread=6
```

2. Check the output of following program and Note down the output in your observation book. [2 Marks]

/*ifparallel.c*/

```
#include<omp.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());
}
}
```

```
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ gcc -o parallel -fopenmp ifparallel.c
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ ./parallel
Enter 0: for serial 1: for parallel
0
Serial val=0 id= 0
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ ./parallel
Enter 0: for serial 1: for parallel
1
Parallel val=1 id= 0
Parallel val=1 id= 1
```

3.Observe and record the output of following program [2 Marks]

/*num_threads.c*/

```
#include<omp.h>
int main(){
#pragma omp parallel num_threads(4)
{
int tid=omp_get_thread_num();
printf("Hello world from thread=%d\n",tid);
}
}
```

Change the num_threads and observe the result.

No of threads=4

```
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ gcc -o threads -fopenmp num_threads.c
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ ./threads
Hello world from thread=3
Hello world from thread=1
Hello world from thread=0
Hello world from thread=2
```

No of threads=8

```
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ gcc -o threads -fopenmp num_threads.c
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ ./threads
Hello world from thread=7
Hello world from thread=2
Hello world from thread=1
Hello world from thread=3
Hello world from thread=5
Hello world from thread=6
Hello world from thread=0
Hello world from thread=4
```