IT301 – Parallel Computing

Assignment – 1

Name: Niraj Nandish

Roll No: 191IT234

1. Finding number of CPU s in system
    a. lscpu command

```
niraj@LAPTOP-66VR7FFQ:~$ 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
Vendor ID:                       GenuineIntel
CPU family:                      6
Model:                           142
Model name:                      Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz
Stepping:                        12
CPU MHz:                         2304.007
BogoMIPS:                        4608.01
Hypervisor vendor:               Microsoft
Virtualization type:             full
L1d cache:                       128 KiB
L1i cache:                       128 KiB
L2 cache:                        1 MiB
L3 cache:                        8 MiB
Vulnerability Itlb multihit:     KVM: Vulnerable
Vulnerability L1tf:              Not affected
Vulnerability Mds:               Not affected
Vulnerability Meltdown:          Not affected
Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp
Vulnerability Spectre v1:        Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2:        Mitigation; Enhanced IBRS, IBPB conditional, RSB filling
Vulnerability Tsx async abort:   Not affected
Flags:                           fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb
                                 rdtscp lm constant_tsc rep_good nopl xtopology cpuid pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 movbe popcnt aes xsave
  avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch invpcid_single ssbd ibrs ibpb stibp ibrs_enhanced fsgsbase bmi1 avx2
                                 smep bmi2 erms invpcid rdseed adx smap clflushopt xsaveopt xsavec xgetbv1 xsaves flush_l1d arch_capabilities
```

```
niraj@LAPTOP-66VR7FFQ:~$ 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
Model name:                      Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz
```

```
niraj@LAPTOP-66VR7FFQ:~$ 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
1,0,0,,,0,0,0,0
2,1,0,,,1,1,1,0
3,1,0,,,1,1,1,0
4,2,0,,,2,2,2,0
5,2,0,,,2,2,2,0
6,3,0,,,3,3,3,0
7,3,0,,,3,3,3,0
```

b. Run top or htop command to obtain the number of CPUs/cores in Linux

```
niraj@LAPTOP-66VR7FFQ: ~
top - 10:35:00 up 41 min,  0 users,  load average: 0.00, 0.00, 0.00
Tasks:   5 total,   1 running,   4 sleeping,   0 stopped,   0 zombie
%Cpu0  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu1  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu5  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu6  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu7  :  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   5943.4 total,   5455.5 free,     75.7 used,    412.1 buff/cache
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   5650.1 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1 root      20   0    1176    844    520 S   0.0   0.0   0:00.03 init
   58 root      20   0    1260    360     20 S   0.0   0.0   0:00.00 init
   59 root      20   0    1260    368     20 S   0.0   0.0   0:00.08 init
   60 niraj     20   0   10060   5020   3308 S   0.0   0.1   0:00.09 bash
  846 niraj     20   0   10900   3664   3128 R   0.0   0.1   0:00.01 top
```

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

```
niraj@LAPTOP-66VR7FFQ:~$ nproc --all
8
niraj@LAPTOP-66VR7FFQ:~$ 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.

```
niraj@LAPTOP-66VR7FFQ:~$ nano simpleomp.c
niraj@LAPTOP-66VR7FFQ:~$ gcc -o simple -fopenmp simpleomp.c
niraj@LAPTOP-66VR7FFQ:~$ export OMP_NUM_THREADS=2
niraj@LAPTOP-66VR7FFQ:~$ ./simple
Hello world from thread=0
Number of threads=2
Hello world from thread=1
```

Using OMP_NUM_THREADS

```
niraj@LAPTOP-66VR7FFQ:~$ nano simpleomp.c
niraj@LAPTOP-66VR7FFQ:~$ gcc -o simple -fopenmp simpleomp.c
niraj@LAPTOP-66VR7FFQ:~$ ./simple
Hello world from thread=0
Number of threads=6
Hello world from thread=1
Hello world from thread=3
Hello world from thread=4
Hello world from thread=2
Hello world from thread=5
```

Using num_threads()

```
niraj@LAPTOP-66VR7FFQ:~$ nano simpleomp.c
niraj@LAPTOP-66VR7FFQ:~$ gcc -o simple -fopenmp simpleomp.c
niraj@LAPTOP-66VR7FFQ:~$ ./simple
Hello world from thread=0
Number of threads=4
Hello world from thread=1
Hello world from thread=3
Hello world from thread=2
```

Using omp_set_num_threads()

3. Check the output of following program and note down the output in your observation book.

```
niraj@LAPTOP-66VR7FFQ:~$ nano ifparallel.c
niraj@LAPTOP-66VR7FFQ:~$ gcc -o ifparallel -fopenmp ifparallel.c
niraj@LAPTOP-66VR7FFQ:~$ ./ifparallel
Enter 0: for serial 1: for parallel
0
Serial val=0 id=0
niraj@LAPTOP-66VR7FFQ:~$ ./ifparallel
Enter 0: for serial 1: for parallel
1
Parallel val=1 id=0
Parallel val=1 id=1
```

4. Observe and record the output of following program

```
niraj@LAPTOP-66VR7FFQ:~$ nano num_threads.c
niraj@LAPTOP-66VR7FFQ:~$ gcc -o num_threads -fopenmp num_threads.c
niraj@LAPTOP-66VR7FFQ:~$ ./num_threads
Hello world from thread=0
Hello world from thread=2
Hello world from thread=3
Hello world from thread=1
niraj@LAPTOP-66VR7FFQ:~$ nano num_threads.c
niraj@LAPTOP-66VR7FFQ:~$ gcc -o num_threads -fopenmp num_threads.c
niraj@LAPTOP-66VR7FFQ:~$ ./num_threads
Hello world from thread=0
Hello world from thread=4
Hello world from thread=6
Hello world from thread=5
Hello world from thread=2
Hello world from thread=7
Hello world from thread=3
Hello world from thread=1
```