

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-V550JN:~$ 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:                     1776.800
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 cache flushes, SMT vulnerable
Vulnerability Mds:           Mitigation: Clear CPU buffers; SMT vulnerable
Vulnerability Meltdown:      Mitigation: PTI
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: Full generic retpoline, IBPB conditional, 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 mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl vnx est tmm2 sse3 sdbg fma cx16 xtpr pcdn pcid sse4_1 sse4_2 xzapl movbe popcnt tsc_deadline_timer aes xsave avx fsgsbase tscrand lbrf lm abm cquid fault_60 epb invpcid_single pti ssbd lbrs lbpv stibp tpr_shadow vnmi flexpriority ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid
```

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

```
bhuvan@bhuvan-N550JK:~$ lscpu | egrep 'Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz|14|18|(s\)'
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Address sizes: 39 bits physical, 48 bits virtual
Core(s) per socket: 4
Socket(s): 1
NUMA node(s): 1
Model name: Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz
CPU MHz: 1433.79
BogoMIPS: 988.68
L1d cache: 32 KiB
L1i cache: 32 KiB
L2 cache: 1 MiB
Vulnerability L1tf: Mitigation: PTE Inversion; VMX conditional cache flushes, SMT vulnerable
Vulnerability Spectre_v1: Mitigation: usercopy/swapgs barriers and __user pointer sanitization
Flags: fpu vme de pse tsc mtr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc cpuid aperfperf pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse_4_1 sse_4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm cpushd_fault epb invpcid_single pti ssbd ibrs lbrp stibp tpr_shadow vnmi flexpriority ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid xsaveopt dt
erm ida arat pln pts mclear_flush_l1d
```

```
$ 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
```

```
bhuvan@bhuvan-N550JK:~$ top
top - 10:30:42 up 10 min, 1 user, load average: 0.69, 0.49, 0.25
Tasks: 270 total, 2 running, 268 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.9 us, 0.7 sy, 0.0 ni, 97.1 id, 0.1 wa, 0.0 hi, 0.1 st, 0.0 st
MiB Mem : 15895.6 total, 13179.0 free, 922.4 used, 1793.4 buff/cache
MiB Swap: 7629.0 total, 7629.0 free, 0.0 used, 14376.9 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR   S  CPU   MEM      TIME+ COMMAND
1657 bhuvan    20   0 1174792 87136 53668 S   7.0  0.5  0:26.95 Xorg
1810 bhuvan    20   0 4284224 244668 82980 S   7.0  1.5  0:29.10 gnome-shell
3357 bhuvan    20   0 815984 44172 35176 S   2.3  0.3  0:06.41 gnome-screensho
2692 bhuvan    20   0 971768 52104 39512 S   2.0  0.3  0:07.71 gnome-terminal-
890 root      20   0 393256 13528 10268 S   1.7  0.1  0:00.18 udiskd
1837 bhuvan    20   0 393576 8552 6840 S   0.7  0.1  0:01.15 lbus-daemon
118 root      20   0 0 0 0 I   0.3  0.0  0:00.38 kworker/u6:1-phy0
860 root      20   0 81908 3696 3360 S   0.3  0.0  0:00.05 irqbalance
1364 root      20   0 0 0 0 I   0.3  0.0  0:00.64 kworker/u17:2-1915_flip
1577 bhuvan    20   0 322700 9590 7796 S   0.3  0.1  0:00.15 gvfs-udisks2-vo
1961 bhuvan    20   0 472416 10232 8816 S   0.3  0.1  0:00.07 gsd-sharing
2207 bhuvan    20   0 498764 24732 18712 S   0.3  0.2  0:00.21 xdg-desktop-por
3331 bhuvan    20   0 20656 4100 3268 R   0.3  0.0  0:00.20 top
1 root      20   0 167848 11764 8384 S   0.0  0.1  0:01.31 systemd
2 root      20   0 0 0 0 S   0.0  0.0  0:00.00 kthread
3 root      20   0 0 0 0 I   0.0  0.0  0:00.00 rcu_gp
4 root      20   0 0 0 0 I   0.0  0.0  0:00.00 rcu_par_gp
6 root      20   0 0 0 0 I   0.0  0.0  0:00.00 kworker/0:0H-kblockd
7 root      20   0 0 0 0 I   0.0  0.0  0:00.07 kworker/0:1-events
8 root      20   0 0 0 0 I   0.0  0.0  0:00.25 kworker/u16:0-events_unbound
9 root      20   0 0 0 0 I   0.0  0.0  0:00.00 mm_percpu_wq
10 root     20   0 0 0 0 S   0.0  0.0  0:00.02 ksoftirqd/0
11 root     20   0 0 0 0 I   0.0  0.0  0:00.41 rcu_sched
12 root     rt   0 0 0 0 S   0.0  0.0  0:00.00 migration/0
13 root     -51  0 0 0 0 S   0.0  0.0  0:00.00 idle_inject/0
14 root     20   0 0 0 0 S   0.0  0.0  0:00.00 cpuhp/0
15 root     20   0 0 0 0 S   0.0  0.0  0:00.00 cpuhp/1
16 root     -51  0 0 0 0 S   0.0  0.0  0:00.00 idle_inject/1
17 root     rt   0 0 0 0 S   0.0  0.0  0:00.12 migration/1
18 root     20   0 0 0 0 S   0.0  0.0  0:00.01 ksoftirqd/1
20 root     20   0 0 0 0 I   0.0  0.0  0:00.00 kworker/1:0H-kblockd
21 root     20   0 0 0 0 S   0.0  0.0  0:00.00 cpuhp/2
22 root     -51  0 0 0 0 S   0.0  0.0  0:00.00 idle_inject/2
23 root     rt   0 0 0 0 S   0.0  0.0  0:00.12 migration/2
24 root     20   0 0 0 0 S   0.0  0.0  0:00.02 ksoftirqd/2
25 root     20   0 0 0 0 I   0.0  0.0  0:00.41 kworker/2:0-events
26 root     20   0 0 0 0 I   0.0  0.0  0:00.00 kworker/2:0H-kblockd
27 root     20   0 0 0 0 S   0.0  0.0  0:00.00 cpuhp/3
28 root     -51  0 0 0 0 S   0.0  0.0  0:00.00 idle_inject/3
29 root     rt   0 0 0 0 S   0.0  0.0  0:00.12 migration/3
30 root     20   0 0 0 0 S   0.0  0.0  0:00.02 ksoftirqd/3
32 root     20   0 0 0 0 I   0.0  0.0  0:00.00 kworker/3:0H-kblockd
33 root     20   0 0 0 0 S   0.0  0.0  0:00.00 cpuhp/4
34 root     -51  0 0 0 0 S   0.0  0.0  0:00.00 idle_inject/4
35 root     rt   0 0 0 0 S   0.0  0.0  0:00.12 migration/4
36 root     20   0 0 0 0 S   0.0  0.0  0:00.02 ksoftirqd/4
38 root     20   0 0 0 0 I   0.0  0.0  0:00.00 kworker/4:0H
```

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()

```

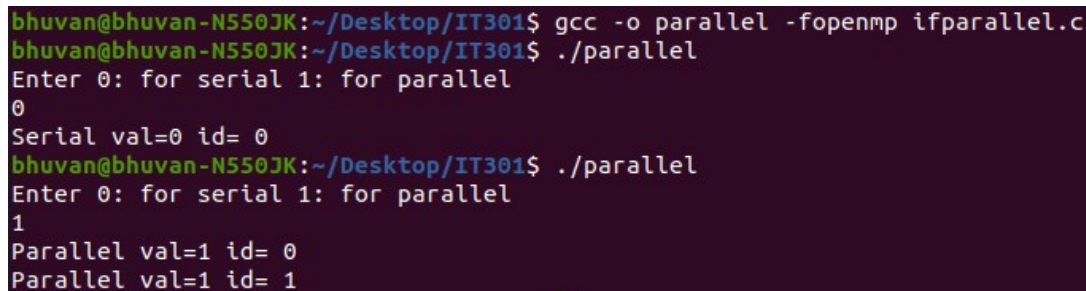
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ gcc -o simple -fopenmp simpleomp2.c
bhuvan@bhuvan-N550JK:~/Desktop/IT301$ ./simple
Hello world from thread=0
Number of threads=8
Hello world from thread=3
Hello world from thread=4
Hello world from thread=7
Hello world from thread=2
Hello world from thread=1
Hello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