High Performance Computing

Name: Thulasi Shylasri

HTNO: 2303A51876

Batch-14

Week-4

LAB Assignment-4 (04/02/2026)

Task 1: OpenMP Parallel Loops (From Serial to Parallel)

Objective: Confirm OpenMP-ready system.

Steps:

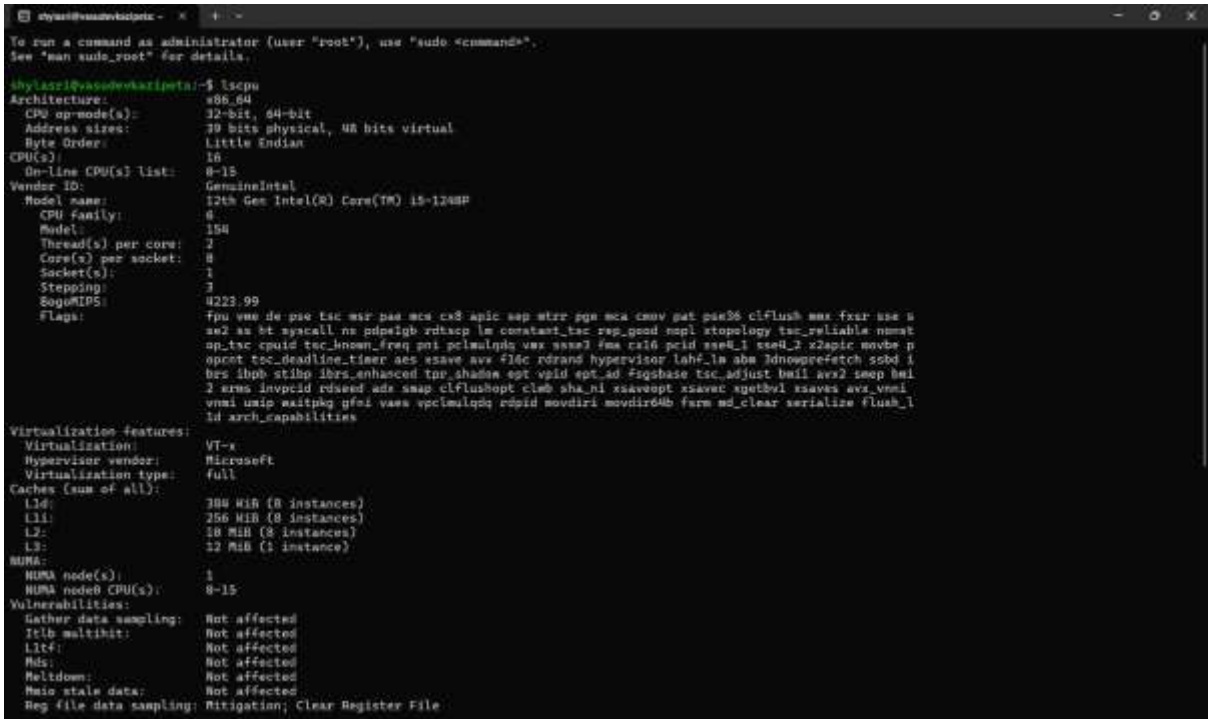
- Lscpu
- nproc
- gcc –version

Expected Observation:

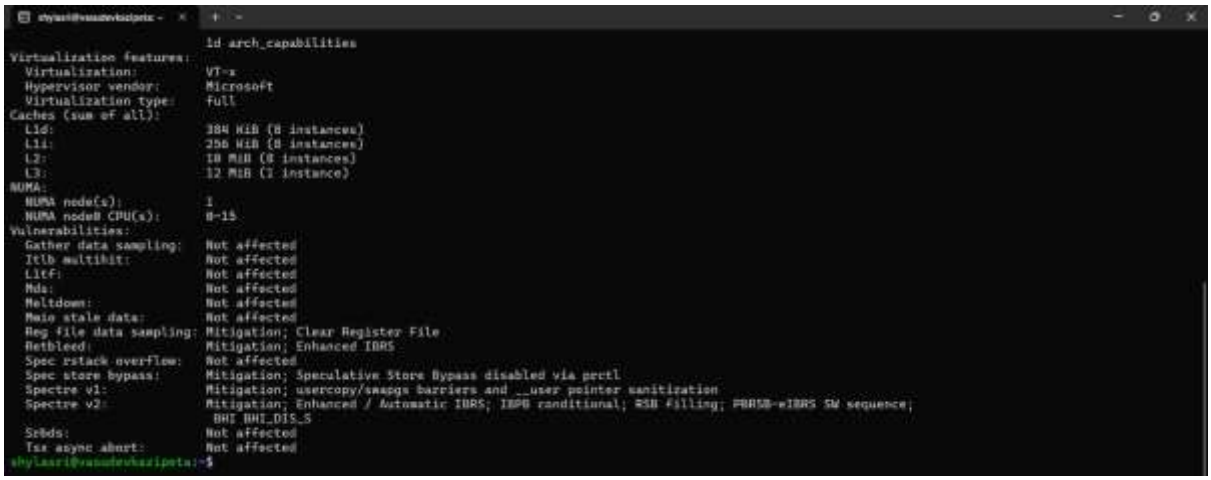
Multiple CPU cores available

GCC installed with OpenMP support

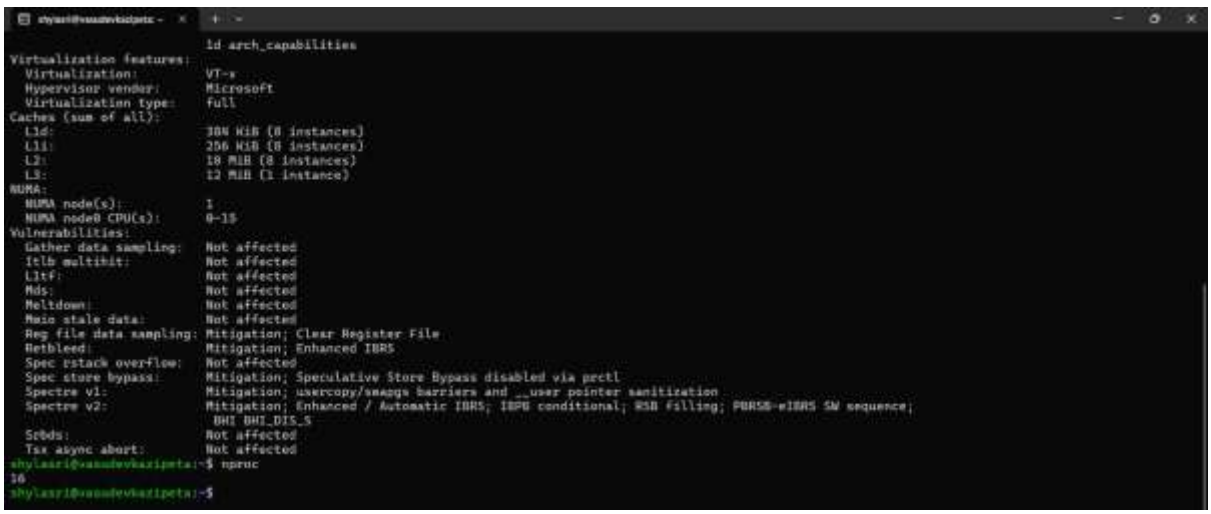
Screenshots:



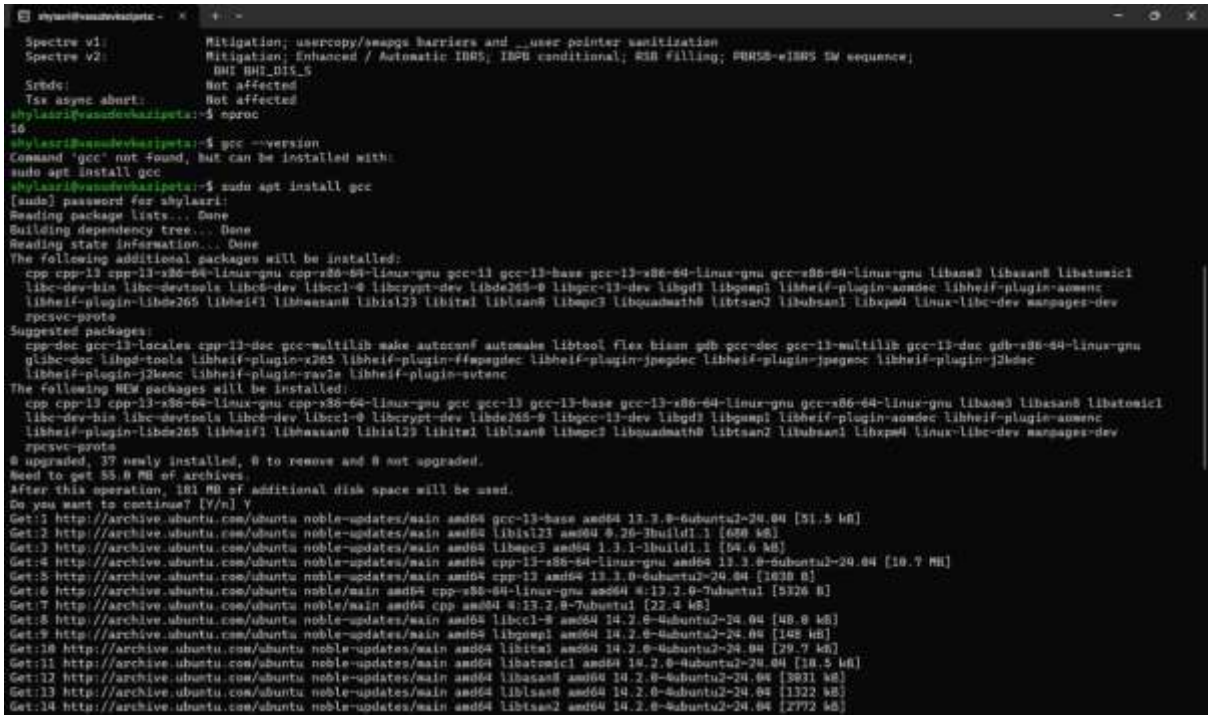
```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

shylasri@vasudevkazipeta:~$ lscpu
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
  Address sizes:         39 bits physical, 48 bits virtual
  Byte Order:            Little Endian
CPU(s):                  16
  On-line CPU(s) list:   0-15
Vendor ID:               GenuineIntel
  Model name:            12th Gen Intel(R) Core(TM) i5-1240P
    CPU family:          6
    Model:               154
    Thread(s) per core:  2
    Core(s) per socket:  8
    Socket(s):           1
    Stepping:            3
    BogoMIPS:            4223.99
    Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse s
                         se2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc rep_good nopl xtopology tsc_reliable nonst
                         op_tsc cpuid tsc_known_freq pni pclmulqdq vmx ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe p
                         opcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch ssbd i
                         brs ibpb stibp ibrs_enhanced tpr_shadow ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi
                         2 erms invpcid rdseed adx smap clflushopt clwb sha_ni xsaveopt xsavec xgetbv1 xsaves avx_vnni
                         vnmi umip waitpkg gfni vaes vpclmulqdq rdpid movdiri movdir64b fsrm md_clear serialize flush_l
                         1d arch_capabilities
Virtualization features:
  Virtualization:        VT-x
  Hypervisor vendor:     Microsoft
  Virtualization type:   full
Caches (sum of all):
  L1d:                   384 KiB (8 instances)
  L1i:                   256 KiB (8 instances)
  L2:                    10 MiB (8 instances)
  L3:                    12 MiB (1 instance)
NUMA:
  NUMA node(s):          1
  NUMA node0 CPU(s):     0-15
Vulnerabilities:
  Gather data sampling:  Not affected
  Itlb multihit:         Not affected
  L1tf:                  Not affected
  Mds:                   Not affected
  Meltdown:              Not affected
  Mmio stale data:       Not affected
  Reg file data sampling: Mitigation; Clear Register File
```



```
                         1d arch_capabilities
Virtualization features:
  Virtualization:        VT-x
  Hypervisor vendor:     Microsoft
  Virtualization type:   full
Caches (sum of all):
  L1d:                   384 KiB (8 instances)
  L1i:                   256 KiB (8 instances)
  L2:                    10 MiB (8 instances)
  L3:                    12 MiB (1 instance)
NUMA:
  NUMA node(s):          1
  NUMA node0 CPU(s):     0-15
Vulnerabilities:
  Gather data sampling:  Not affected
  Itlb multihit:         Not affected
  L1tf:                  Not affected
  Mds:                   Not affected
  Meltdown:              Not affected
  Mmio stale data:       Not affected
  Reg file data sampling: Mitigation; Clear Register File
  Retbleed:              Mitigation; Enhanced IBRS
  Spec rstack overflow:  Not affected
  Spec store bypass:     Mitigation; Speculative Store Bypass disabled via prctl
  Spectre v1:            Mitigation; usercopy/swapgs barriers and __user pointer sanitization
  Spectre v2:            Mitigation; Enhanced / Automatic IBRS; IBPB conditional; RSB filling; PBRSB-eIBRS SW sequence;
                          BHI BHI_DIS_S
  Srbds:                 Not affected
  Tsx async abort:       Not affected
shylasri@vasudevkazipeta:~$
```



```
                         1d arch_capabilities
Virtualization features:
  Virtualization:        VT-x
  Hypervisor vendor:     Microsoft
  Virtualization type:   full
Caches (sum of all):
  L1d:                   384 KiB (8 instances)
  L1i:                   256 KiB (8 instances)
  L2:                    10 MiB (8 instances)
  L3:                    12 MiB (1 instance)
NUMA:
  NUMA node(s):          1
  NUMA node0 CPU(s):     0-15
Vulnerabilities:
  Gather data sampling:  Not affected
  Itlb multihit:         Not affected
  L1tf:                  Not affected
  Mds:                   Not affected
  Meltdown:              Not affected
  Mmio stale data:       Not affected
  Reg file data sampling: Mitigation; Clear Register File
  Retbleed:              Mitigation; Enhanced IBRS
  Spec rstack overflow:  Not affected
  Spec store bypass:     Mitigation; Speculative Store Bypass disabled via prctl
  Spectre v1:            Mitigation; usercopy/swapgs barriers and __user pointer sanitization
  Spectre v2:            Mitigation; Enhanced / Automatic IBRS; IBPB conditional; RSB filling; PBRSB-eIBRS SW sequence;
                          BHI BHI_DIS_S
  Srbds:                 Not affected
  Tsx async abort:       Not affected
shylasri@vasudevkazipeta:~$ nproc
16
shylasri@vasudevkazipeta:~$
```



Task 2: Serial Baseline – Loop Computation

Objective: Establish serial execution time.

File: sum_serial.c

#include <stdio.h>

#include <time.h>

#define N 100000000

int main() {

double sum = 0.0;

clock_t start = clock();

for (long i = 0; i < N; i++) {

sum += i * 0.5;

}

clock_t end = clock();

double time_spent = (double)(end - start) /

CLOCKS_PER_SEC;

printf("Serial Sum: %f\n", sum);

printf("Execution Time: %f seconds\n", time_spent);

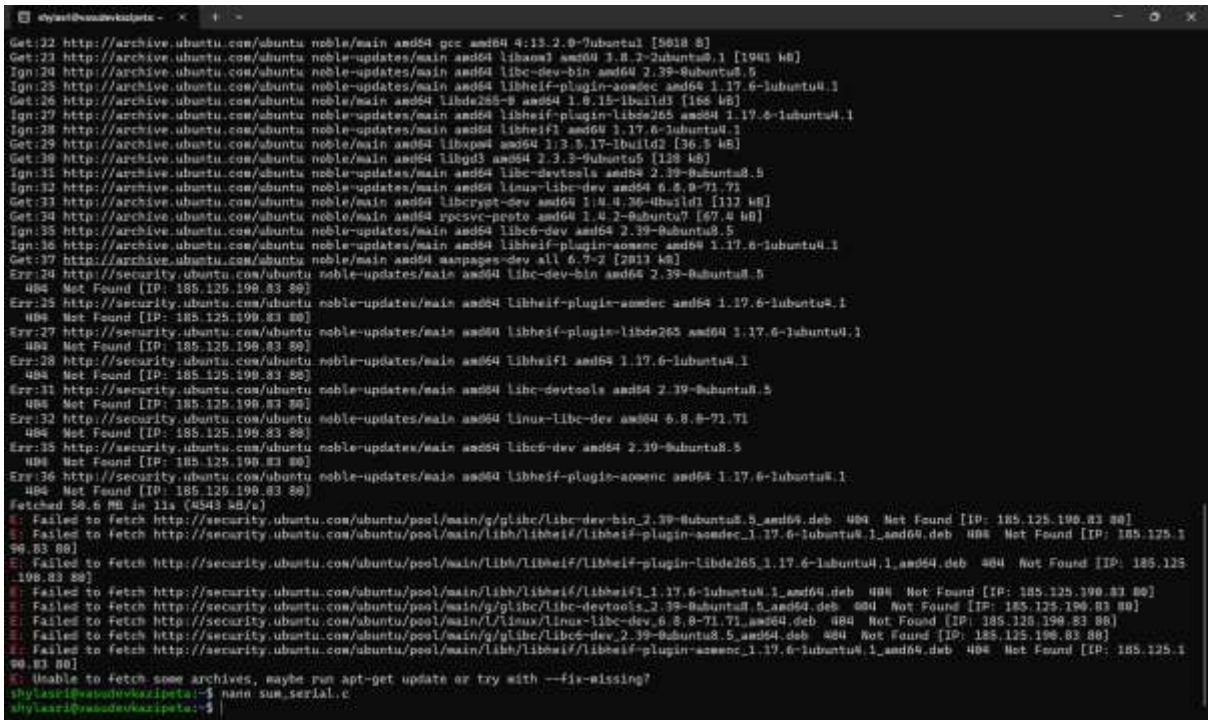
return 0;

Compile & Run:

> gcc sum_serial.c -o sum_serial

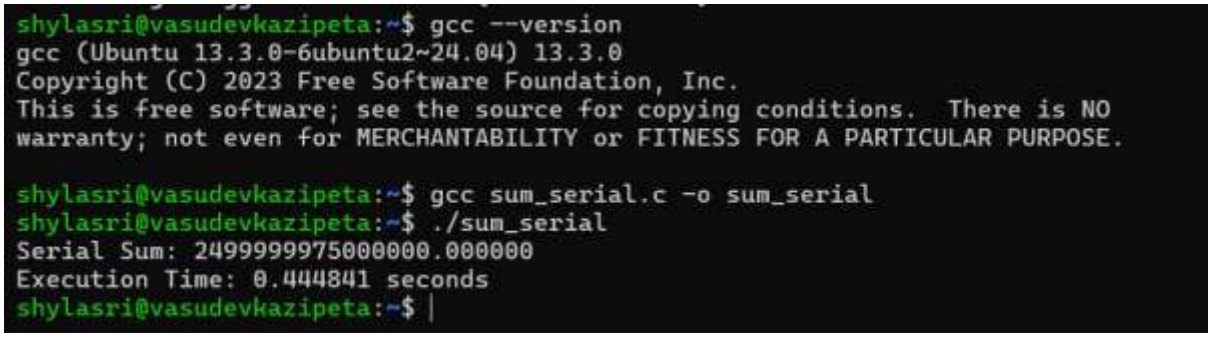
> ./sum_serial

Submission: Screenshot showing execution time.



```
Get:16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [536 B]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [28.8 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [6492 B]
Get:20 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Get:21 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [396 B]
Get:22 http://archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:23 http://archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:24 http://archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:25 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:26 http://archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:27 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:28 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:29 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1723 kB]
Get:30 http://archive.ubuntu.com/ubuntu noble-updates/main Translation-en [322 kB]
Get:31 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:32 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [16.4 kB]
Get:33 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1528 kB]
Get:34 http://archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [313 kB]
Get:35 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [386 kB]
Get:36 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.9 kB]
Get:37 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2547 kB]
Get:38 http://archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [503 kB]
Get:39 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:40 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [556 B]
Get:41 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [32.1 kB]
Get:42 http://archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [6816 B]
Get:43 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:44 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [496 B]
Get:45 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [40.4 kB]
Get:46 http://archive.ubuntu.com/ubuntu noble-backports/main Translation-en [9208 B]
Get:47 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7300 B]
Get:48 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [368 B]
Get:49 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [29.5 kB]
Get:50 http://archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [17.9 kB]
Get:51 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [10.5 kB]
Get:52 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1444 B]
Get:53 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:54 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:55 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:56 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 39.6 MB in 12s (3171 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
126 packages can be upgraded. Run 'apt list --upgradable' to see them.
shylasri@vasudevkazipeta:~$
```

```
Get:39 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:40 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [556 B]
Get:41 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [32.1 kB]
Get:42 http://archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [6816 B]
Get:43 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:44 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [496 B]
Get:45 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [40.4 kB]
Get:46 http://archive.ubuntu.com/ubuntu noble-backports/main Translation-en [9208 B]
Get:47 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7300 B]
Get:48 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [368 B]
Get:49 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [29.5 kB]
Get:50 http://archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [17.9 kB]
Get:51 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [10.5 kB]
Get:52 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1444 B]
Get:53 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:54 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:55 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:56 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 39.6 MB in 12s (3171 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
126 packages can be upgraded. Run 'apt list --upgradable' to see them.
shylasri@vasudevkazipeta:~$ sudo apt install gcc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libasan8 libatomic1
  libc-bin libc-dev-bin libc-devtools libc6 libc6-dev libcc1-0 libcrypt-dev libde265-0 libgcc-13-dev libgd3 libgomp1 libheif-plugin-aomdec
  libheif-plugin-aomenc libheif-plugin-libde265 libheif1 libhwasan0 libisl23 libitm1 liblsan0 libmpc3 libquadmath0 libtsan2 libubsan1 libxpm4
  linux-libc-dev locales manpages-dev rpcsvc-proto
Suggested packages:
  cpp-doc gcc-13-locales cpp-13-doc gcc-multilib make autoconf automake libtool flex bison gdb gcc-doc gcc-13-multilib gcc-13-doc gdb-x86-64-linux-gnu
  glibc-doc libnss-nis libnss-nisplus libgd-tools libheif-plugin-x265 libheif-plugin-ffmpegdec libheif-plugin-jpegdec libheif-plugin-jpegenc
  libheif-plugin-j2kdec libheif-plugin-j2kenc libheif-plugin-rav1e libheif-plugin-svtenc
The following NEW packages will be installed:
  cpp cpp-13 cpp-13-x86-64-linux-gnu cpp-x86-64-linux-gnu gcc gcc-13 gcc-13-base gcc-13-x86-64-linux-gnu gcc-x86-64-linux-gnu libasan8 libatomic1
  libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libde265-0 libgcc-13-dev libgd3 libgomp1 libheif-plugin-aomdec libheif-plugin-aomenc
  libheif-plugin-libde265 libheif1 libhwasan0 libisl23 libitm1 liblsan0 libmpc3 libquadmath0 libtsan2 libubsan1 libxpm4 linux-libc-dev manpages-dev
  rpcsvc-proto
The following packages will be upgraded:
  libc-bin libc6 locales
3 upgraded, 37 newly installed, 0 to remove and 123 not upgraded.
Need to get 12.6 MB/63.2 MB of archives.
After this operation, 181 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```



Task 3: OpenMP Parallel Loop

Objective: Parallelize the loop using OpenMP.

File: sum_openmp.c

#include <stdio.h>

#include <omp.h>

#define N 100000000

int main() {

double sum = 0.0;

double start = omp_get_wtime();

#pragma omp parallel for reduction(+:sum)

for (long i = 0; i < N; i++) {

sum += i * 0.5;

}

double end = omp_get_wtime();

printf("Parallel Sum: %f\n", sum);

printf("Execution Time: %f seconds\n", end - start);

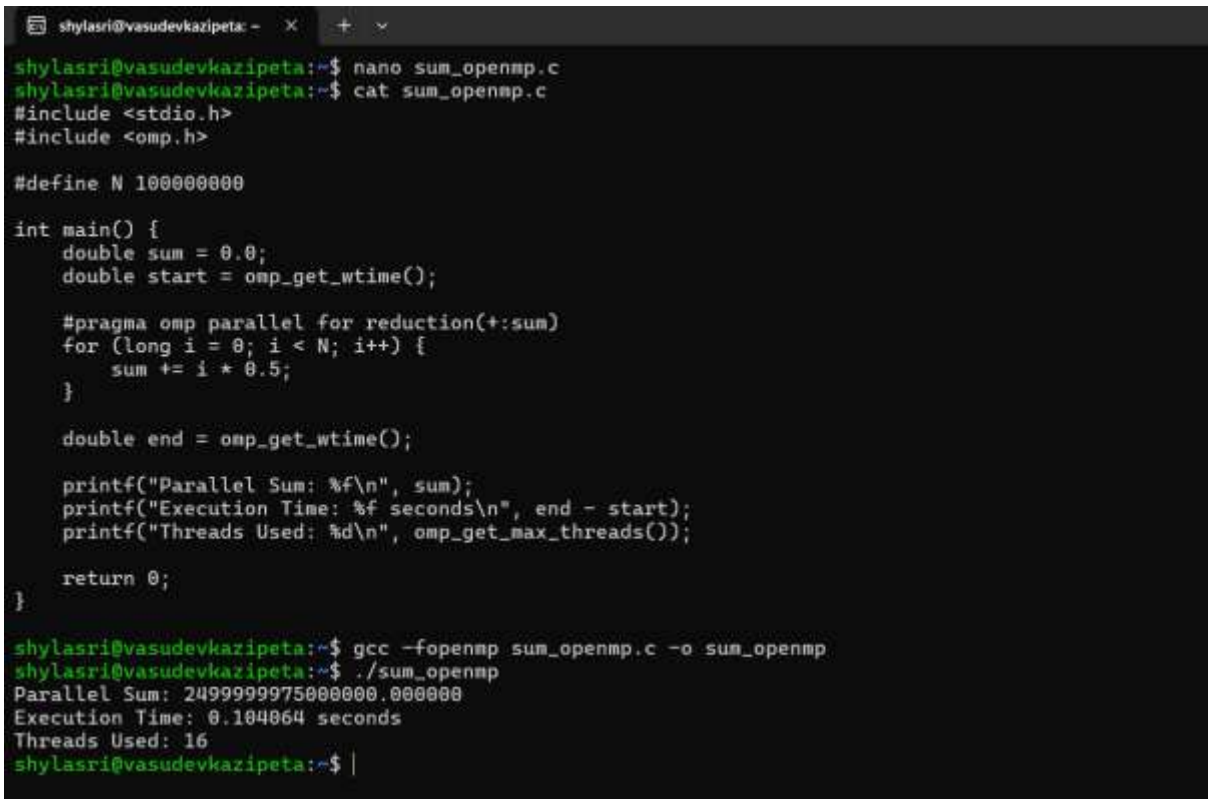
printf("Threads Used: %d\n", omp_get_max_threads());

return 0;

}

Compile:

> gcc -fopenmp sum_openmp.c -o sum_openmp



Task 4: Thread Scaling Experiment

Run with different thread counts:

export OMP_NUM_THREADS=1

./sum_openmp

export OMP_NUM_THREADS=2

./sum_openmp

export OMP_NUM_THREADS=4

./sum_openmp

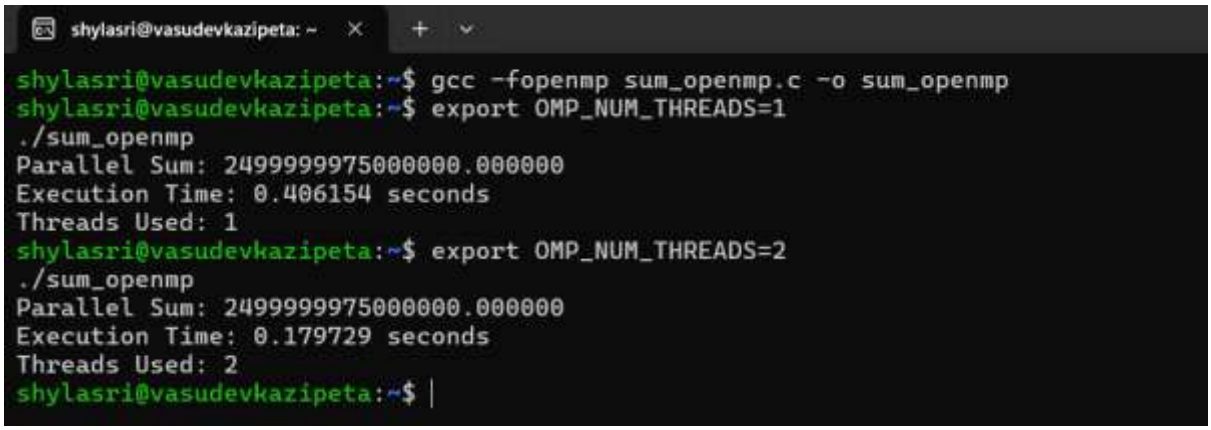
export OMP_NUM_THREADS=8

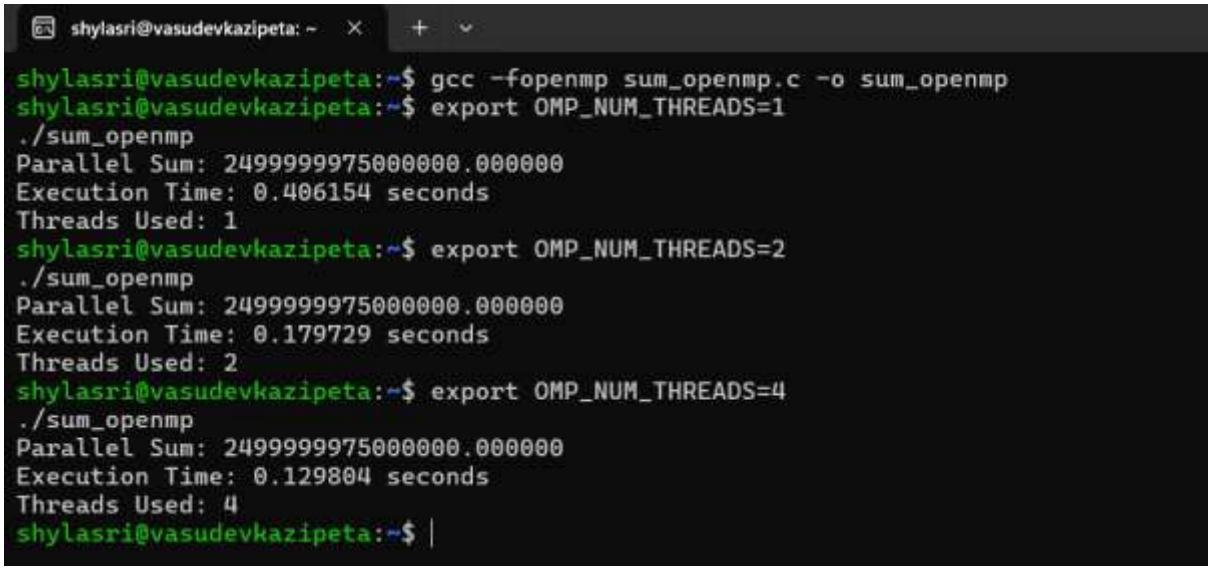
./sum_openmp

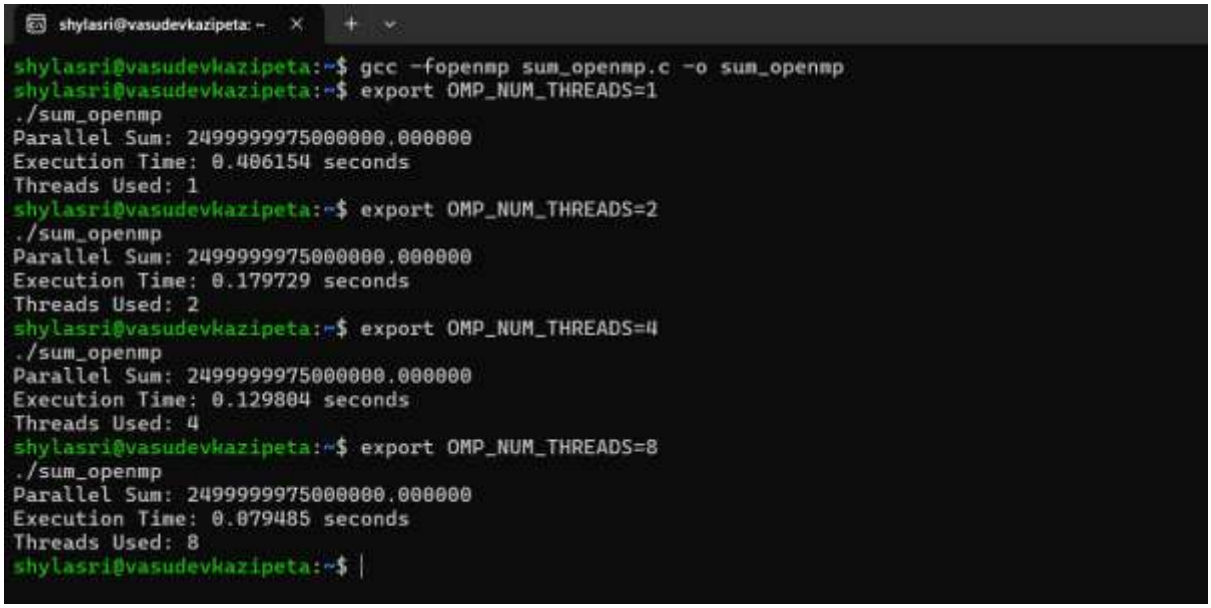
Record the readings in a below table:

Threads Execution Time (s)

Threads	Execution Time (s)
1	0.406154
2	0.179729
4	0.129804
8	0.079485







Task 5 — Speedup & Efficiency Calculation

Formula:

Speedup

$$S(p) = \frac{T1}{Tp}$$

Efficiency

$$E(p) = \frac{S(p)}{p}$$

Where, T1 = 0.406154 seconds

Observation

- Execution time decreases as threads increase.
- Speedup improves with thread count.
- Efficiency decreases when thread count increases due to overhead and resource sharing.

Threads	Time (s)	Speedup	Efficiency
1	0.406154	1.00	1.00
2	0.179729	2.26	1.13
4	0.129804	3.13	0.78
8	0.079485	5.11	0.64

Observation

Execution time decreases as threads increase.

Speedup improves with thread count.

Efficiency decreases when thread count increases due to overhead and resource sharing.

Task 6:

1. Why does OpenMP work well for this loop?

OpenMP works well because each loop iteration is independent and can be executed in parallel without affecting other iterations. The workload is evenly divided among threads, and the reduction clause safely combines partial results.

2. What happens if loop iterations are dependent?

If loop iterations are dependent, parallel execution can produce incorrect results because some iterations may require results from previous iterations. This leads to data races unless synchronization is added, which reduces performance.

3. Why does speedup stop improving after some threads?

Speedup stops improving due to thread creation overhead, synchronization costs, memory bandwidth limitations, and the finite number of CPU cores available on the system.

4. Is OpenMP suitable for multi-node systems? Why/why not?

No, OpenMP is not suitable for multi-node systems because it uses shared memory. Multi-node systems require distributed memory programming models such as MPI.

Observation:

In this experiment, OpenMP significantly reduced execution time by parallelizing an independent loop across multiple threads. As the number of threads increased, performance improved due to better CPU utilization. However, the speedup gradually saturated because of hardware limitations, thread management overhead, and memory access constraints. This shows that while OpenMP is highly effective for shared-memory parallelism, optimal performance depends on problem structure and system resources. OpenMP is best suited for multi-core, single-node systems with independent computations.