**Task 1:**

**1.**

**Documentation of Commands and Steps**

**1. Compile and Run OpenMP Serial Program**

**Compilation Command:**

gcc -o openmp\_serial openmp\_serial.c -fopenmp\texttt{gcc -o openmp\\_serial openmp\\_serial.c -fopenmp}gcc -o openmp\_serial openmp\_serial.c -fopenmp

**Execution Command:**

./openmp\_serial\texttt{./openmp\\_serial}./openmp\_serial

* **Description:** Compiles and executes the serial version of the OpenMP program.
* **Output:**

Execution time: 0.014685 secondsNumber of threads used: 4\begin{align\*} \text{Execution time:} & \ 0.014685 \ \text{seconds} \\ \text{Number of threads used:} & \ 4 \end{align\*}Execution time:Number of threads used:​ 0.014685 seconds 4​

**2. Compile and Run OpenMP Matrix Multiplication Program**

**Compilation Command:**

gcc -o openmp\_matrix openmp\_matrix.c -fopenmp\texttt{gcc -o openmp\\_matrix openmp\\_matrix.c -fopenmp}gcc -o openmp\_matrix openmp\_matrix.c -fopenmp

**Execution Command:**

./openmp\_matrix\texttt{./openmp\\_matrix}./openmp\_matrix

* **Description:** Compiles and executes the matrix multiplication using the OpenMP framework.
* **Output:**

Matrix multiplication completed.Execution time: 0.215744 secondsNumber of threads used: 4\begin{align\*} \text{Matrix multiplication completed.} \\ \text{Execution time:} & \ 0.215744 \ \text{seconds} \\ \text{Number of threads used:} & \ 4 \end{align\*}Matrix multiplication completed.Execution time:Number of threads used:​ 0.215744 seconds 4​

**3. Compile and Run Serial Test Program**

**Compilation Command:**

gcc -o serial\_test serial\_test.c\texttt{gcc -o serial\\_test serial\\_test.c}gcc -o serial\_test serial\_test.c

**Execution Command:**

./serial\_test\texttt{./serial\\_test}./serial\_test

* **Description:** Compiles and executes the serial test program.
* **Output:**

Execution time: 0.016402 seconds\text{Execution time:} \ 0.016402 \ \text{seconds}Execution time: 0.016402 seconds

**4. Compile and Run MPI Matrix Multiplication Program**

**Compilation Command:**

mpicc -o mpi\_matrix mpi\_matrix.c\texttt{mpicc -o mpi\\_matrix mpi\\_matrix.c}mpicc -o mpi\_matrix mpi\_matrix.c

**Execution Command:**

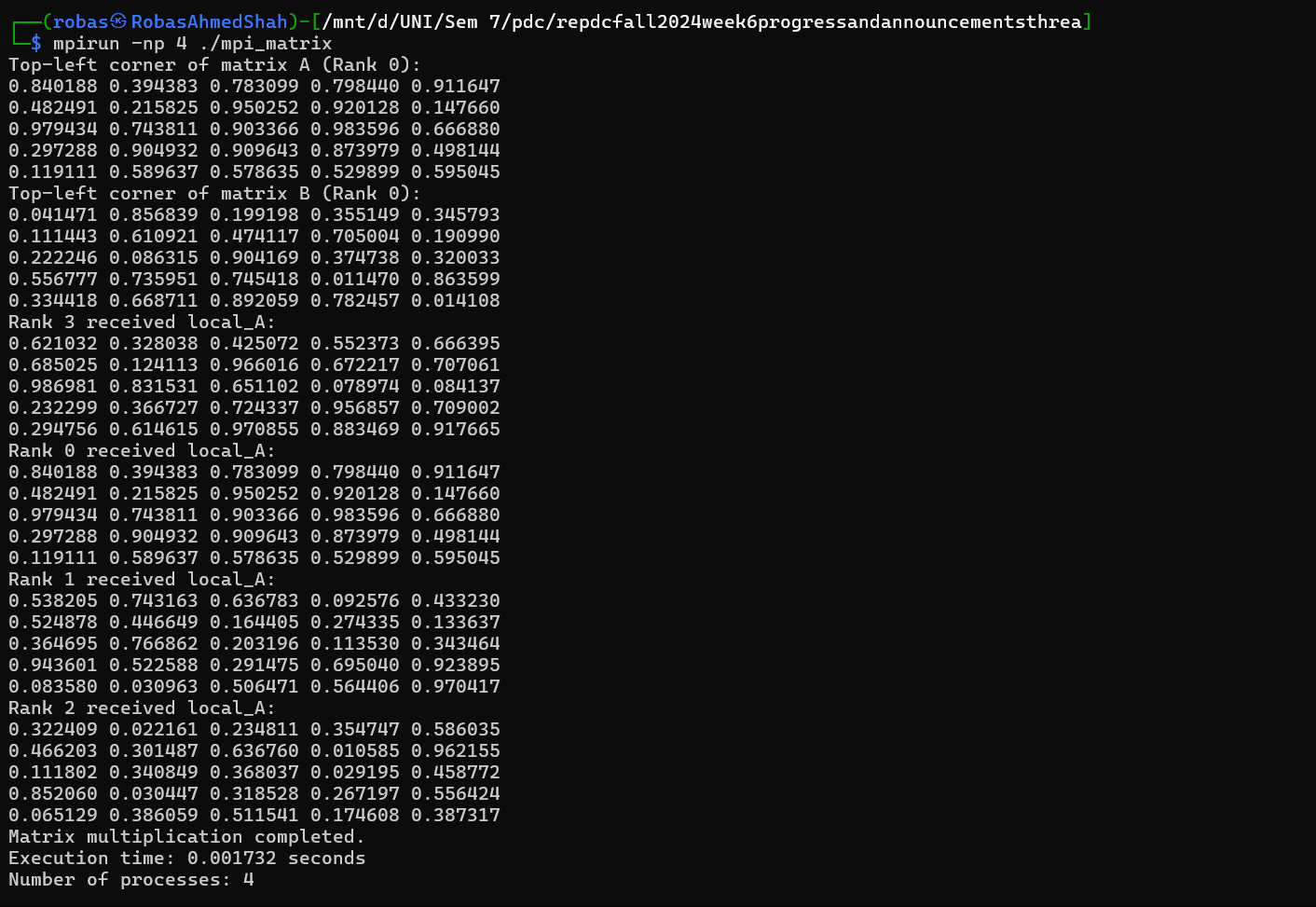
mpirun -np 4 ./mpi\_matrix\texttt{mpirun -np 4 ./mpi\\_matrix}mpirun -np 4 ./mpi\_matrix

* **Description:** Compiles and executes the MPI version of the matrix multiplication program using 4 processes.
* **Output:**

Top-left corner of matrix A (Rank 0):0.840188 0.394383 0.783099 0.798440 0.9116470.482491 0.215825 0.950252 0.920128 0.1476600.979434 0.743811 0.903366 0.983596 0.6668800.297288 0.904932 0.909643 0.873979 0.4981440.119111 0.589637 0.578635 0.529899 0.595045Matrix multiplication completed.Execution time: 0.001732 seconds

Number of processes: 4

****

****

**2.**

**Hardware Characteristics**

**CPU Information**

* **Model:** Intel(R) Core(TM) i7-8665U CPU @ 1.90GHz
* **Number of cores:** 4
* **Number of threads:** 8 (2 threads per core)
* **Clock speed:** 1.90 GHz

**Cache Information**

* **L1d cache:** 128 KiB (4 instances)
* **L1i cache:** 128 KiB (4 instances)
* **L2 cache:** 1 MiB (4 instances)
* **L3 cache:** 8 MiB (1 instance)

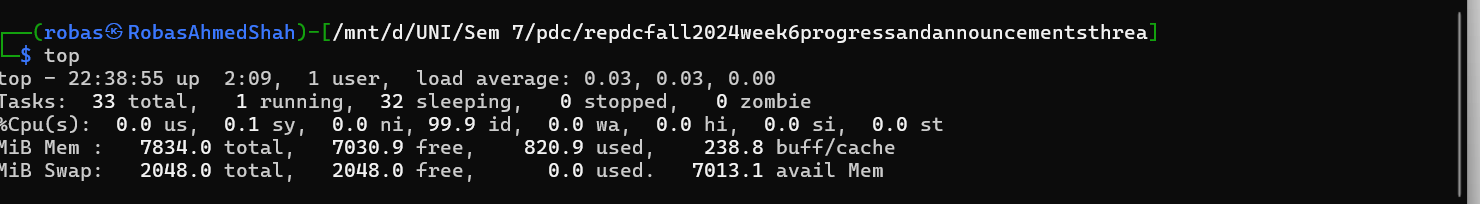
**Memory Information**

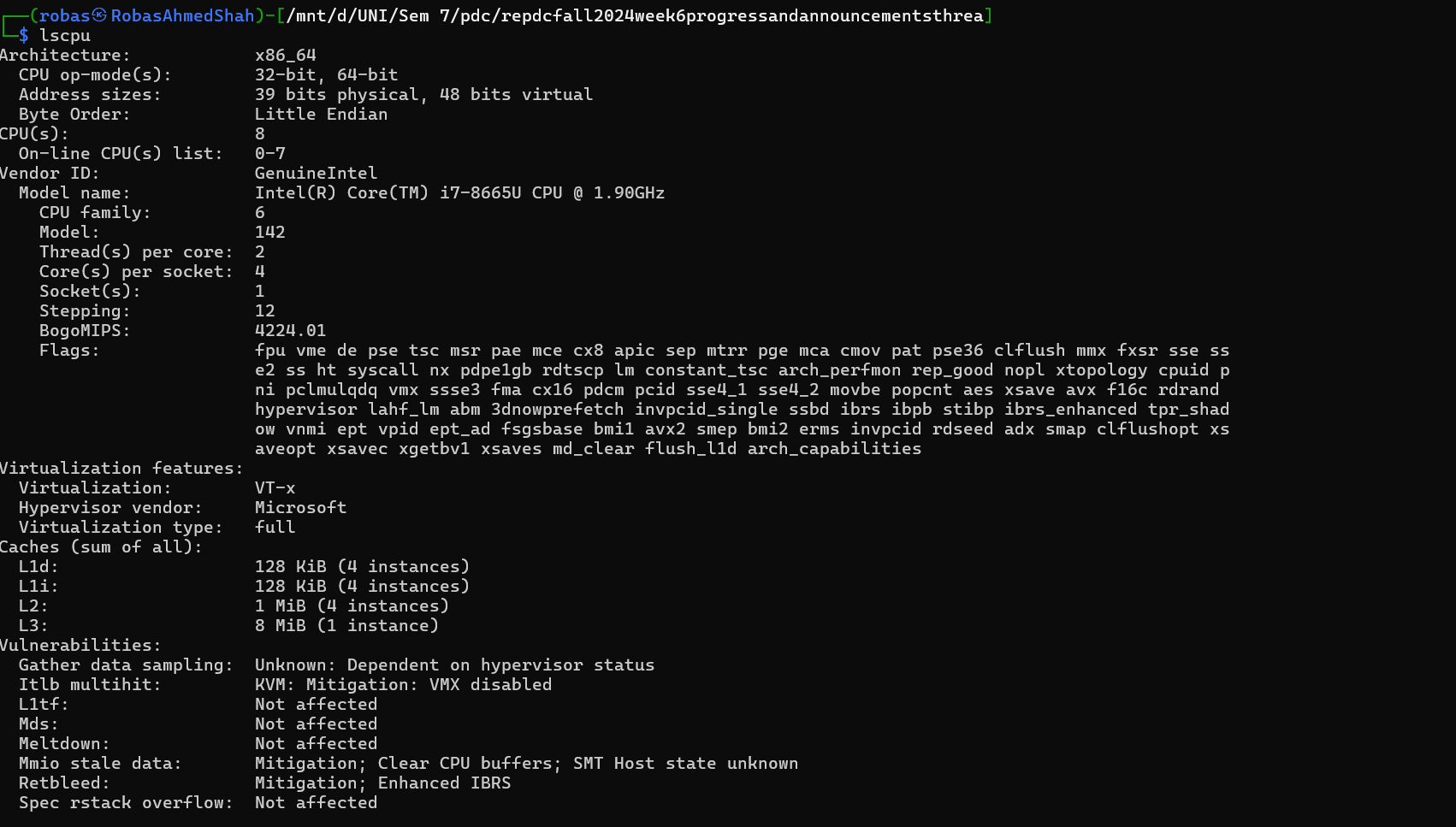
* **Total RAM:** 7834 MiB
* **Available RAM:** 7033 MiB

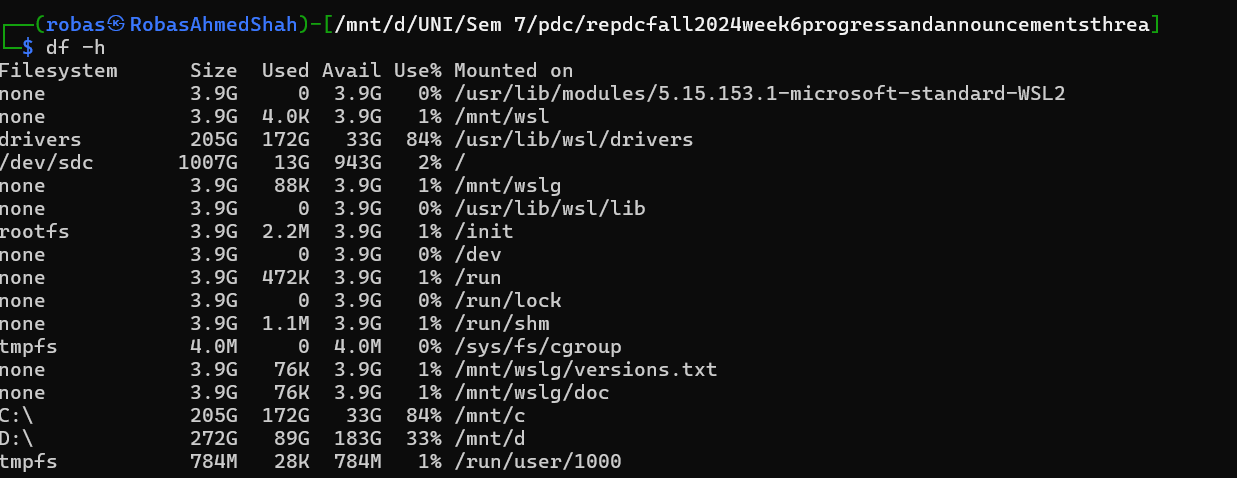
**Disk Information**

* **Total disk space:** 1007 GiB (for /dev/sdc, your main filesystem)
* **Available disk space:** 943 GiB (for /dev/sdc, your main filesystem)

**Additional Observations**

* **Load Average:** 0.05, 0.03, 0.00
* **Tasks:** 33 total (1 running, 32 sleeping)
* **Memory Usage:** 818.8 MiB used, 238.8 MiB buff/cache
* **Swap Usage:** 2048 MiB total, 2048 MiB free
* **CPU Utilization:** 0.0% user, 0.0% system, 0.0% nice, 100.0% idle





3.Click to open CPUZ-Report

