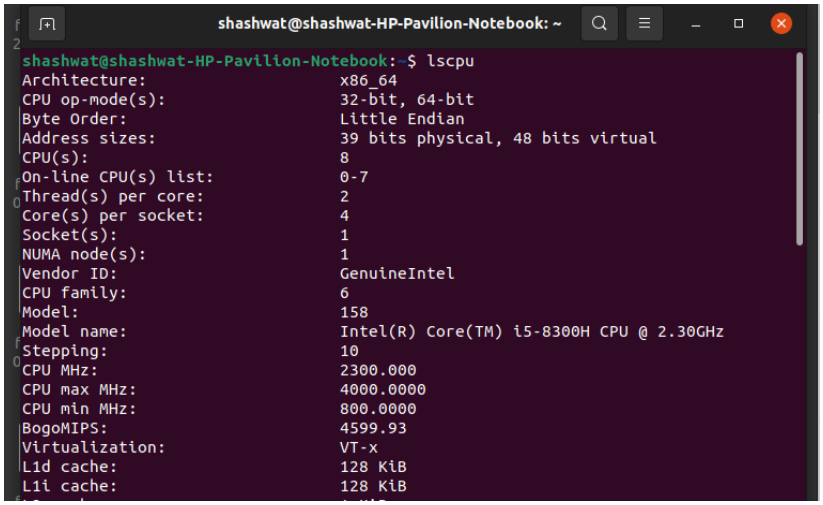
# Operating Systems Laboratory (CS39002)
# Assignment 3

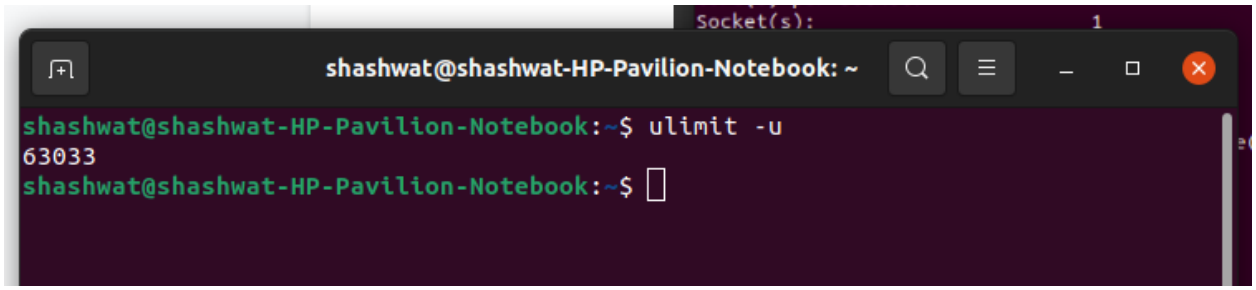
**Group 23**
**Parth Tusham 19CS30034**
**Shashwat Shukla 19CS10056**

**Task 1B**

- A multitasking operating system can easily switch processes to make it look like many processes are running in parallel), but in reality it can run only a single process per CPU core at a time. If the CPU is multicore, then multithreading or other techniques can be used to execute multiple processes concurrently.
- Hence r1*c2 should be less than or equal to the number of cores in the machine to execute all the said processes concurrently in real.




- In our case we have four cores hence r1*c2 should be less than or equal to 4
- Other than this if we consider that forking creates a concurrent process and not a computer changing tasks very rapidly, and fooling slow human beings into thinking it's doing several things at once.
- Then we can say The command ulimit -u shows the maximum number of processes that you can start.




- We did a practical test by executing a file in which within a while loop we were forking until we were unable to fork more and maintained a counter to see how many child processes were running sudo-concurrently.
- We found out that number to be near 20,000.