

Operating Systems Laboratory (CS39002)

Assignment 3

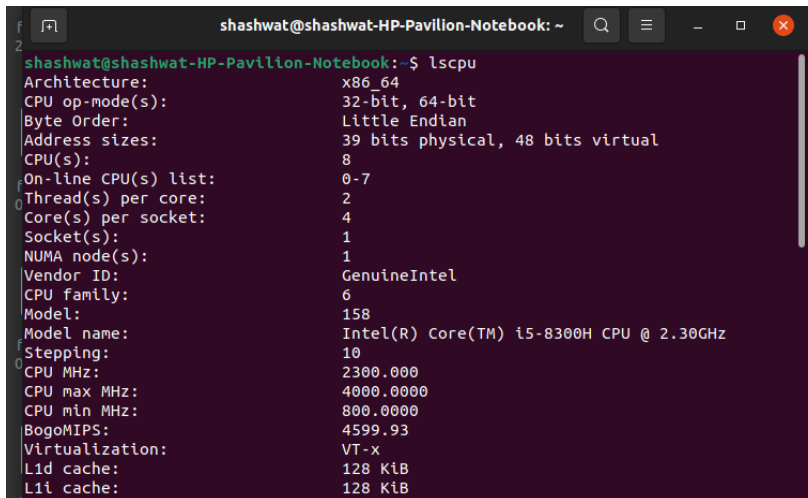
Group 23

Parth Tusham 19CS30034

Shashwat Shukla 19CS10056

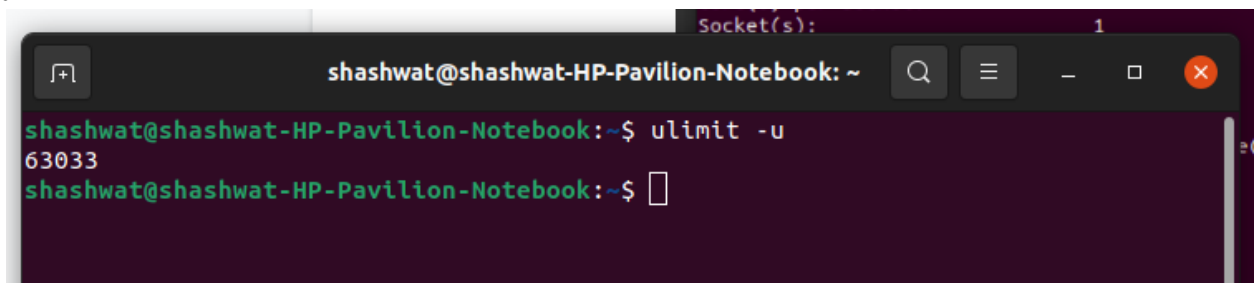
Task 1B

- 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 $r1 \cdot c2$ should be less than or equal to the number of cores in the machine to execute all the said processes concurrently in real.



```
shashwat@shashwat-HP-Pavilion-Notebook: ~  
shashwat@shashwat-HP-Pavilion-Notebook:~$ 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:                158  
Model name:            Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz  
Stepping:              10  
CPU MHz:               2300.000  
CPU max MHz:           4000.0000  
CPU min MHz:           800.0000  
BogoMIPS:              4599.93  
Virtualization:        VT-x  
L1d cache:             128 KiB  
L1i cache:             128 KiB
```

- In our case we have four cores hence $r1 \cdot c2$ 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.



```
shashwat@shashwat-HP-Pavilion-Notebook: ~  
shashwat@shashwat-HP-Pavilion-Notebook:~$ ulimit -u  
63033  
shashwat@shashwat-HP-Pavilion-Notebook:~$
```

- 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.