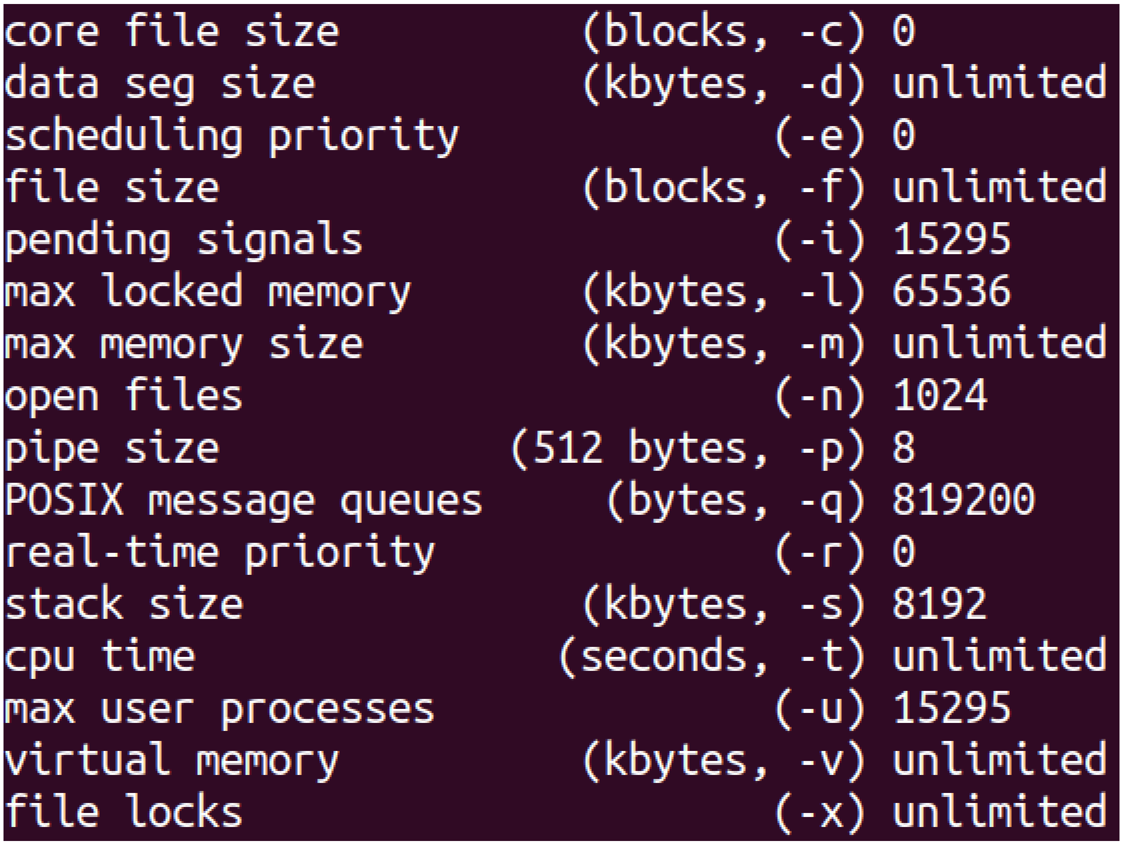
Operating Systems Laboratory Assignment 3

Group No - 53

19CS30050 - Suryam Arnav Kalra

19CS30025 - Kunal Singh

- The maximum size of matrix that we can multiply is around 9500 (r1*c2)
- This limit comes because the maximum user processes for our local machine is around 15295.



```
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 15295
max locked memory       (kbytes, -l) 65536
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) 15295
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

Output after running the command “ulimit -a”

- This shows that we can create around 62 percent child processes of the maximum number of processes for our system.
- The difference between the maximum number of processes (15925) and the maximum size of matrix (9500) we can multiply is because of other threads and processes that our system is executing concurrently and the resources are getting divided between our program and other processes.
- The number of cores in our system is 4 , so at any instant our system can run 4 processes at max parallely which decreases the total time of computation for matrix multiplication by a factor of 4.