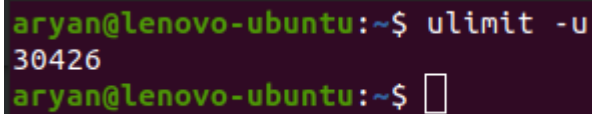


In our solution we fork child processes for each of the cells in the final matrix, but this is not unlimited as there is a kernel imposed limit on the number of user processes that can be spawned as shown below.

A terminal window with a dark purple background. The prompt is 'aryan@lenovo-ubuntu:~\$'. The command 'ulimit -u' has been entered, and the output '30426' is displayed on the next line. The prompt is now 'aryan@lenovo-ubuntu:~\$' followed by a cursor.

```
aryan@lenovo-ubuntu:~$ ulimit -u
30426
aryan@lenovo-ubuntu:~$
```

Also we can run into memory limitations as each process needs additional memory which again might cut short the number of processes. On stress testing our matrix multiplication program we were able to fork 9907 processes ($r1*c2 \leq 9907$), which is significantly less than the max user processes allowed.

However these processes are executed concurrently i.e. they are not truly parallel and a lot of context switching happens thus only few processes are executed truly parallel which is limited by the number of cores in the system.

Details :

Group 16

Vinit Raj, 19CS10065

Aryan Agarwal, 19CS30005