**CSC-3044 Operating Systems & System Programming**

**Laboratory 3 Report**

**NAME:** Joseph Leskey

1. **Which do you prefer to use: the library call or the errno utility? Why? When is either approach useful?**

The library call. When writing a program, it removes an unnecessary step for the user. The library call is useful for including functionality in a program. Errno is useful as a manual interpreter.

1. **Are there any differences when running this program in superuser mode (e.g., w/ sudo or as root)? Describe any differences in the lab report and the possible source for those differences.**

The only difference I noted running the program as a superuser was that two fewer processes were made. In this case, I would assume that the difference comes about due to the additional processes launched by `sudo`.

1. **What does this resource do?**

It defines the maximum number of processes that can be made in the current context.

1. **Experiment with each of the following values to setrlimit. Describe what happens in your lab report using these limits and any errors that are observed.**

**1 and 1**  
  
No processes record. Immediate error 11.

**10 and 20**

Two processes created before error 11.

**100 and 200**

Many processes created before error 11.

**1000 and 2000**Even more processes created before error 11.

**10000 and 20000**Operation was not permitted. These limits exceed existing bounds. Fell back to default. Many new processes before error 11.

**2000 and 1000**

Invalid argument. The hard limit is lower than the soft limit. Fell back to default. Many new processes before error 11.

1. **Capture any output using the screen utility, another utility of your choice, or redirection of standard and error output streams. Include the output with your submission in a file named lab3\_Q5.txt.**

Done.

1. **For any errors that occur, use the errno command to get the human-readable form of the error and describe the likely sources for this error in your lab report.**

My program already prints human-readable errors.

1. **What happens if you try running this program in superuser mode (e.g., sudo ./part1 1 1, etc.)? Note the output and behavior in your lab report.**

It does not seem to be quite as limited. It generates a large number of child processes before error 11.

1. **Capture any output using the screen utility, another utility of your choice, or redirection of standard and error output streams. Include the output with your submission in a file named lab3\_Q8.txt.**

Done.

1. **Do you observe any new behaviors in the process creation process for the limits you specified? Please answer in your lab write-up.**

I don’t believe so.

1. **What resource type did you use?**

I tried RLIMIT\_DATA and RLIMIT\_AS.

1. **Capture any output using the screen utility, another utility of your choice, or redirection of standard and error output streams. Include the output with your submission in a file named lab3\_Q11.txt.**

There did not seem to be any measurably new output.

1. **In your lab report, determine and clearly document the impact creating one, two, four, and eight threads have on the overall performance given a virtual machine using one, two, or four (or four+) cores as your host system capabilities permit. You may want to reboot your system in between each run to remove/limit the impact of caching.**
2. **If using an increasing number of threads takes less time, why is that?**
3. **If using an increasing number of threads takes more time, why is that as well?**
4. **In your lab report, determine and clearly document the impact of using various values of N as you attempt to find perfect numbers 28, 486, 8128, and beyond. Once again, you should try varying the number of virtual processors and rebooting between attempts.**