Pawel Derkacz
151-00-5994
pd304

# Report 1 for Assignment 1 from Intro. To Computer Systems

Concerning what I have learned from this assignment, I know a lot more about piping now than I did previously. Other than that, I cannot really say – I have learned about multi-threading in a different course.

As for design decisions, there is a sweet spot to have the correct amount of forks for a given input. Too low an input amount, and the overhead for forking is not worth the effort. Too many inputs per process will also just bog the process down and have it consume a large amount of CPU time. I also attempted to have the parent do calculations right after forking, instead of beforehand, so that way there is more time for the child to perform its own duties.

Concerning efficiency, it is my belief that part C could be the best in terms of time consumption, given a very large input (larger than 100k) – but part A is better if only for the reason that it is easy to implement and runs in about O(3n) time. Again, part C would only be better if the calculations can be done simultaneously and in large quantities. Part B is absolutely horrendous requiring a lot of overhead with piping. The reason part D works much better than part B is due to the fact that I recoded it in a very different way, favoring more towards the way part C was coded. However, part D is seems to still be slower due to the fact that it requires that much more overhead – although it was the only multi-processing one to not crash when attempting 100k inputs.

The following are the timings taken in the Linux terminal by use of the command “time”.

Timings for **Part A**)

| # of numbers | 10 | 100 | 1000 | 10k | 100k |
|---|---|---|---|---|---|
| Real time | 0.002s | 0.001s | 0.002s | 0.002s | 0.021s |
| CPU time | 0.000s | 0.000s | 0.000s | 0.000s | 0.020s |

Timings for **Part B** @ **chunks of 5** numbers per process)

| # of numbers | 10 | 100 | 1000 | 10k * | 100k * |
|---|---|---|---|---|---|
| Real time | 0.003s | 0.006s | 0.168s | 7.862s | 8.482s |
| CPU time | 0.000s | 0.000s | 0.000s | 6.024s | 6.568s |

* 10k and 100k crashed due to too many open pipes.

Timings for **Part B** @ **chunks of 50** numbers per process)

| # of numbers | 10 | 100 | 1000 | 10k | 100k * |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| Real time | 0.001s | 0.003s | 0.005s | 0.134s | 8.402s |
| CPU time | 0.000s | 0.000s | 0.000s | 0.000s | 6.572s |

* 100k crashed due to too many open pipes

Timings for **Part C** @ **chunks of 5** numbers per process)

| # of numbers | 10 | 100 | 1000 | 10k | 100k * |
|---|---|---|---|---|---|
| Real time | 0.001s | 0.003s | 0.014s | 0.022s | 0.021s |
| CPU time | 0.000s | 0.000s | 0.012s | 0.016s | 0.020s |

* 100k crashed due to too many open pipes

Timings for **Part C @ chunks of 50** numbers per process)

| # of numbers | 10 | 100 | 1000 | 10k | 100k * |
|---|---|---|---|---|---|
| Real time | 0.002s | 0.001s | 0.006s | 0.021s | 0.038s |
| CPU time | 0.000s | 0.000s | 0.004s | 0.020s | 0.020s |

* 100k crashed due to too many open pipes

Timings for **Part D** @ **chunks of 5** numbers per process)

| # of numbers | 10 | 100 | 1000 | 10k | 100k |
|---|---|---|---|---|---|
| Real time | 0.003s | 0.004s | 0.019s | 0.022s | 0.105s |
| CPU time | 0.000s | 0.000s | 0.000s | 0.000s | 0.008s |

Timings for **Part D @ chunks of 50** numbers per process)

| # of numbers | 10 | 100 | 1000 | 10k | 100k |
|---|---|---|---|---|---|
| Real time | 0.001s | 0.003s | 0.006s | 0.013s | 0.029s |
| CPU time | 0.000s | 0.000s | 0.000s | 0.000s | 0.008s |