Pedro Ochoa
Dr. Kong
Operating Systems
05/08/24

Fork and Pipe

When beginning to create this project, I was nervous about attempting to create a program in C. Bringing in the concepts that we had learned in class helped immensely when breaking down the ways to tackle this problem. Now in the instructions it states, we would typically think having more processes would speed up the task that we are being asked. That is not true from the examples I had run with the different processes and different files.

I began with just going with practice problems to begin learning the fundamentals of coding in C. Then I began coding the manner of which my program would work with the blocks/chunks of lines from each file, and aggregating their results per child process. Unfortunately I continued running into problems with having the final total summing all the child processes. I was able to have the number of children be decided by the user input, utilizing the ubuntu terminal in linux.

Like it stated in the instructions as well, the manner in which someone structures their code would affect the completion speed for the program to execute. WIthin most of my examples, having only 1 process child typically was the fastest way of completing the task of summing all the numbers. With File 3, test 1 and 4 do have unique similarities, but I found this project to display the usefulness for a task to be broken down, possibly completing it at a faster pace.

	File 1	File 2	File 3
1 Process	0.000358	0.003204	0.033272
2 Processes	0.000445	0.003316	0.035894
4 Processes	0.000863	0.014673	0.033595