**FILE ACCESS**

* Having multiple processes access and modify the file at the same time was actually not hard. I just had both the producer and consumer processes infinitely try to do their respective jobs, and it just ended up working.
* Both processes could have the file open simultaneously, but the modifications wouldn't be seen by the other program until it was saved and reopened.
* I ensured that both processes accessed the same file by hardcoding the name of the file in, and having the programs print an error if they couldn't open the file for some reason.
* One challenge I had was that originally, both programs opened the file once at the start, then just wrote to it on repeat, never closing it. This caused issues, particularly with consume(), which needed the file to be opened separately for reading and then writing to work.
* Another challenge was my original implementation of the Consumer. It would read the last line and delete it, however, when it was run in parallel with the Producer, it would never read the contents of the file. To mitigate, I made the Consumer just delete the whole contents of the file.

**OBSERVATIONS**

* Due to my implementation of the Consumer, it was able to keep up with any amount of Producers just fine. I did notice that when I had the file open in VSCode, I could see the file fill up and empty out. With more producers than consumers, the file would fill up more prior to being empty. With an even amount, the file filled up very little.
* As more producers and consumers are run, the threads in my CPU each had more work. I did not program any threading, but it would seem that my OS automatically assigned different processes to different threads.
* Memory usage also increased with more processes, as is to be expected.
* I didn't notice much performance issue until I reached about 75 processes. The new terminals that opened were each getting slower, both with opening and with the program. Background tasks slowed down. I still haven't found a limit to how many processes I can run, but given that I have a very powerful CPU, I think the number is pretty high.