I have approached this project in writing logic by implementing a pseudocode per each file. I started with logger file, where it would take in a line of command, then parsing it for first word and everything else. The first word would be in [ ] and last will be outside of it. I moved on to the encryption program, where it would also take in a line of code, parsing it per any of the word PASSKEY, ENCRYPT, DECRYPT, and QUIT. Parsing the first word for this program was essential in telling the program what to do. I was able to find this by using an string array where it stored PASSKEY, ENCRYPT, DECRYPT, and QUIT values to linearly search for functions.

Before I started working on the main driver program, I made a logic board for the pipes and forks on the communications for the files.

Diagram, schematic

Description automatically generated

By utilizing this diagram, I was quickly able to make the pipes to connect all the programs.

I have organized the project this way because writing logic and pseudocode helps me greatly in implementing any type of coding.

The driver program was by far the hardest among all, where linking the communications between the other two programs seemed initially impossible by running into a plethora of problems.

The problems I have ran into were: connecting to the UTD SSH, g++ command line, Segmentation fault, loss of memory, and passing information between each processes.

Some of these problems were fixed: SSH server was fixed by communicating with UTD IT, g++ command line was fixed by installing ubuntu, passing information between each processes was fixed by restarting the code from scratch.

From this project, I was able to firmly grasp the idea of process with their respected virtual address space. This has also allowed me to understand how a process is created, as well as scheduled approach to programming. It was also cool to note how encryption and logger had no idea about each other’s existence.

I did not complete the project by not being able to make the processes fully communicate each other with the driver. This was due to some unsolvable issues that I had while building this project. Unending segmentation faults, invalid access to the files in which I have created, as well as files being unusable at some point. It also came to a conclusion that running cs1 and cs2 was borderline impossible, where both of those servers crashed every 5 minutes for me that prohibited access and running any type of command.

Text

Description automatically generatedGraphical user interface, text, application

Description automatically generatedText

Description automatically generated

At some point, I lost 3 days’ worth of work by an unexpected ‘blue screen’ in which the error code was ‘System\_Service\_exception’. This gives some clue on why I have been having troubling debugging and running the code.

Shape

Description automatically generated with medium confidence

While installing essential functions for this project, I have run into a storage problem on my laptop, which I believe it was the underlying issue for this project.

If I had more time, I would investigate in a efficient way in running shell type programs, as well as an IDE in which allows me to run command lines in.