**Design and Algorithm**

The goal of this program is to implement the command:

ps -A | grep argv[1] | wc -l

where argv[1] is passed from the command line.

I start of by creating two pipes. **pipeFdsOne** to connect from the child (wc -l) to the grandchild (grep argv[1]) and **pipeFdsTwo** to connect from the grandchild (grep argv[1]) to the great grandchild (ps -A). I then fork the **pid** a total of three times to create all three child processes. Along the way, checking to make sure each previous fork was successful and that I am at the current child. Upon success of each fork, corresponding pipes are created to connect the processes together. Once I have created all the child processes I am ready to do I/O redirection and execute. We start with the great grandchild and close all off unused pipe ends and the write to **pipeFdsTwo** and then execute its command **ps -A**. The grandchild then reads from **pipeFdsTwo** and redirects the input to **pipFdsOne** and is then able to execute its command **grep argc[1]** where argv[1] is passed in from the command line. The child is then able to retrieve from **pipeFdsOne** and is able to execute **wc -l.** The output of the full command is then prnted to the console.

**Flow Chart**

Diagram

Description automatically generated

This flow chart shows how the I/O redirection takes place between all the processes through the pipes.

**Testing**

I tested my program using multiple cout statements to visualize which parts of the program were being executed. I also implemented a lot of error checking within my program so it would be easier to understand exactly where I am getting an error and what has caused it.

**Output**

Text

Description automatically generated