**CSCI 3431: Operating System Winter 2022**

**Project 1-Inter Process Communication using Pipes**

**Date out: January 26, 2022, Due on: Feb 22, 2022 (11:59pm)**

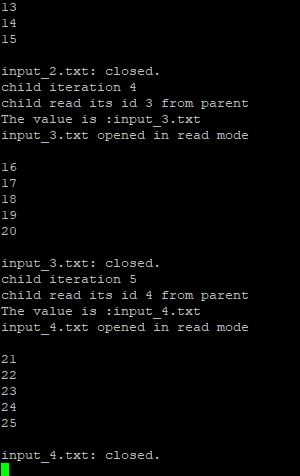
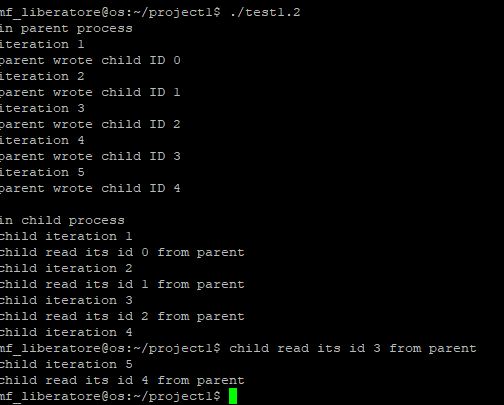
**Wonjun Park A00423142**

**Introduction**

For project 1 my partner was Marcus Liberatore. The purpose of this project was to become familiar with the mechanisms of inter-process communication used in UNIX. Using pipes, we communicate using pipes with reading and writing mechanisms. This allows a parent process (fork() > 0) to communicate with a child process (fork() == 0). Codes are defined at the beginning of the algorithm to manipulate our array. It is the main way of sending messages between the pipes. Much of the sources for the project came from the web and from things taught to us during class time. As for the division of tasks, I worked on the parent process and algorithm and Marcus worked on the child process. We combined the two sections together to create the finished product. I also worked on most of the error handling and testing as well.

**Task Discussion**

My portion of the project was the code for the parent process. I initialized the variables in the beginning of main(), such as the parent to child pipe (ptc[5][2];). These will be needed later on in the algorithm. My main approach was to use a for loop to fork() so that I can create the necessary amount of parent-child pairs. The parent child pairs establish communication to make sure that all the connections were working in the algorithm. Here are some early tests:



Once I was sure these connections were established, I progressed toward matching the sample output. Using the pivot, small, and large method from the child processes to match the sample output. The value from the child processes were checked to see whether k-value and m-value were the same. If they were, I exited out of the program. If not, determine which is bigger, then send the necessary code to the child. This was done using a while loop.

**Results**

Graphical user interface

Description automatically generated**Graphical user interface

Description automatically generated with medium confidence**The bugs make it so that it works sometimes only. However, upon various hours of bug fixing, I have yet to realize what the problem is. When it succeeds, it provides the correct pivot value as well as the amount of iterations it took, and the k-value, and m-value per iteration. However, when it fails, the program stops working and you have to control c out of the program.

**References**

* Week 5 - priority scheduling: C programming tutorial.pdf
* Week 3 - interprocess communications: 3.1-Process Communications pdf
* <https://stackoverflow.com/questions/39823004/c-checking-for-null-in-a-2d-array>
* <https://stackoverflow.com/questions/822323/how-to-generate-a-random-int-in-c>
* <https://stackoverflow.com/questions/42843988/sending-childs-process-id-to-parent-through-pipe>