Project Report CMPUT 379 Assignment 2

Karan Sharma

Objectives:

The objective of the assignment was split into two parts. The first objective was to become familiar with multithreaded programs and seeing how multithreaded programming can allow tasks to run asynchronously. You can have two tasks running side by side in parallel and both tasks can share memory meaning they can both be using and changing variables. The second objective was to become familiar with pipes and how different processes can interact with each other using pipes, and more specifically the half-duplex FIFO pipes. Also the use of I/O multiplexing functions such as poll() to be able to queue I/O from different sources and only deal with the ones that are ready and have information ready.

Design Overview:

Part 1 was straightforward, I used a global variable called 'flag' to allow the thread to keep track of when it was allowed to take user input. The flag was set to 0 when the secondary thread was allowed to take user input, and set to 1 whenever the primary thread was printing the nlines. Otherwise everything else from printing the nlines to processing the user input were done in their respective main, or thread functions. Error checking was also included whenever any kind of function that could fail was called.

Part 2 was a little bit more difficult. It starts with processing the arguments given by the user when running the program, some error checking is done to ensure they had entered valid arguments and then depending on the arguments the server or client is called. Starting with the client, I made sure the file was open and then used a while loop to iterate through and grab every line in the inputFile. Any lines that are newlines are empty lines are ignored, every other line is processed and sent to the server. Firstly the client tokenizes the line to be transmitted to figure out the id and command. If the Id matches the user and the command is not delay, it is sent to the server to be processed. Otherwise if the command is delay, the client itself processes the delay. Lastly if the Id does not match the client then the line is ignored and continues to the next line.
The server side is a bit more modular, the server starts by opening all the pipes and adding their file descriptors to the poll, along with adding the keyboard input.

I then iterate over each polling descriptor and only process the ones that have something to be read from using the poll() function. Once again any lines that have a '#' or newline are ignored and they are skipped. If the packettype is valid then we once again tokenize and then using a separate function process the packet and see if the command is a valid action, if it is return true, else return false. The appropriate message is then written using a write only pipe back to the respective client.

Project Status:

I believe the programs work as intended. For the first part there is interaction as well as independent actions between both threads. For the second part the clients and server are able to communicate with each other, and multiple client can use the server at once.

Part 1 was fairly straight forward and the threads were easy to work with. For part 2 the main difficulty was trying to get I/O multiplexing to work. I was unfamiliar with how poll() and select() worked. I attempted to use select() at first and after countless hours of no success I decided to attempt poll() and was able to get it working as intended.

Testing and Results:

The main way I would test is using cout statements within my code, and also using VSCodes built in debugger. The debugger allowed me to step through each line and see all the variables at every moment in time. Another pretty common problem I encountered was the interaction between c++ strings and c char arrays. Most of the functions used are c functions and as such do not accept c++ strings. This caused quite a bit of confusion as I would have to constantly swap between the two, and would often spend lots of time just trying to figure out how to get the right format for the function arguments.

Acknowledgments:

APUE 3E Textbook
https://stackoverflow.com/questions/15102992/what-is-the-difference-between-stdin-and-stdin-fileno
https://stackoverflow.com/questions/4184468/sleep-for-milliseconds
https://www.geeksforgeeks.org/snprintf-c-library/
https://en.cppreference.com/w/cpp/chrono/duration/duration_cast

https://stackoverflow.com/questions/347949/how-to-convert-a-stdstring-to-const-char-or-char
https://cplusplus.com/reference/vector/vector/