

Joshua Shuller

Yusuke Yusada

Kumaresh Rajan

## Project 4 Write Up

Project 4 involved making a chat program between two users on the same network. The existing connections from the network allowed the project to be done using FIFOs, named pipes. Each FIFO was created as globally writeable, but only the owner of the FIFO could read its content. The non-blocking flag was used, so that the open call wouldn't block other processes from using the pipe. This setup allowed any user to write to the available FIFO, yet that same owner would not be able to view the contents of the named pipe of the other user. As specified by the project description our group formatted the string as per instructions. The first four characters designated the type of message (connect, accept, ignore, or message). Receiving a message that didn't follow this standard resulted in telling the user that an invalid message was either sent or received. The next byte in the string was an integer that showed the length of the other user's FIFO, followed by the other user's name and path of his/her FIFO. This formatting is kept for most messages except for strings starting with "MMMM". In this case every field is the same except the 6<sup>th</sup> byte is the message length, followed by the users name, and finally the message itself. When a message is received a user has the option of accepting, rejecting, or ignoring the received message. There are two modes of chat that can be used in the program. If an argument from the command line was entered the program goes to active mode. The argument typed is the path name to the other users named pipe. When coding the program the username, FIFO name, and the message field were globally available. Doing this made the coding easier for the programmers.

The program has three main parts serve, connect, and session. Serve is used in passive mode and initializes the other users information. With this information the session carries on the chat between both users. In the active mode connect is used rather than serve. Like serve, connect initializes the other user's fields. The program was only designed to handle two users. Multiple users will not be able to communicate together. If another user tries to message users that are already talking than that user is given a reject message. Also, the program formats all messages for the user. No data structures were used in the program.