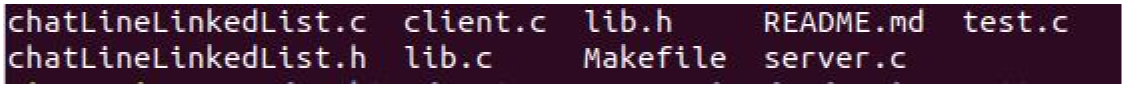
Client and Server Chat

This report contains information on the functionality of the Assignment 2, client and server chat.

Structure

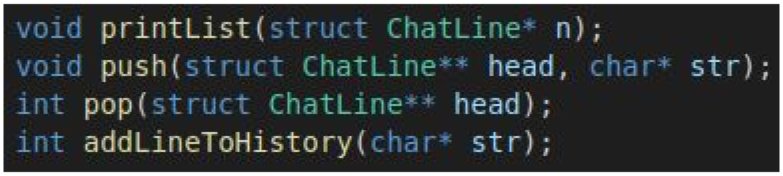
The file structure starts off with the starter code of the client.c server.c and lib.c.

However another file is added called chatLineLinkedList.c. This file contains the list for handling the history of the chat.



```
chatLineLinkedList.c  client.c  lib.h     README.md  test.c
chatLineLinkedList.h  lib.c     Makefile  server.c
```

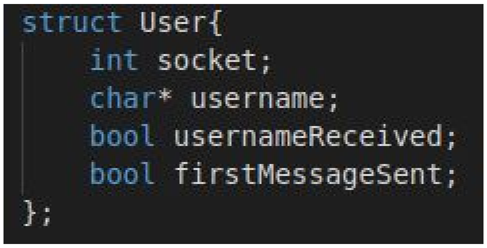
The ChatLineLinkedList file contains 4 helper functions to handle the Chat history list.



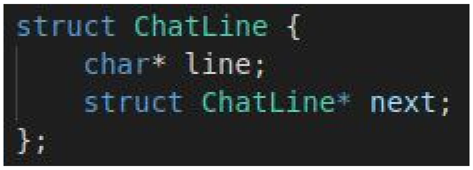
```
void printList(struct ChatLine* n);
void push(struct ChatLine** head, char* str);
int pop(struct ChatLine** head);
int addLineToHistory(char* str);
```

Data Structures

There are two types of data structures that are used in this assignment: Structs and Linked List. The structs that were used are for the list of clients that contains all the clients that have been accepted by Server. The other struct is a simple ChatLine struct that is used as a Linked List to handle the Chat history. The Linked List keeps a max number of lines where it removes the old messages and adds the new. The current `MAX_NUM_OF_LINES` is set to 12.



```
struct User{
    int socket;
    char* username;
    bool usernameReceived;
    bool firstMessageSent;
};
```



```
struct ChatLine {
    char* line;
    struct ChatLine* next;
};
```

Functionalities

The Server handles multiple clients using the select( ) system call. It keeps track of a fd_set of read file descriptors that adds and checks which of the clients are ready to be read.

On the other hand the client side uses a fork. The child process handles the writing functionality which receives input from the User and sends the input to the Server. The parent process handles the reading streams from the Server that is constantly received from other clients.