CAB403 Assignment Report

Eugene Martens

N10318313

Tasks Completed
Able to implement all three 3 tasks

- Next not implemented properly via threads
- Channels bugged within linux terminal, but work with cygwin
- Thread Syncing has issues

Group
Worked Alone

Data structures
Created a shared memory block that all process wrote to and from that can be found in shared.h/shared.c

Client data was handled through the use of a localized linked list that stored channel substrictions. When functions NEXT and LIVEFEED are called, client subbed linked list is matched to retrieve relevant data and update posts and read index of the subscription.

Forking
Upen receiving a connection from a client, server will invoke a fork process, that includes a server chat handler that will communicate with the client and accept commands and post responses via read and write.

Critical Section resolution
All channels in the shared memory space include a mutex lock, when clients send to the channel the mutex is locked until writing is done.

Threading
Two threads are initiated at the beginning of each server process fork, one for live feed and another next. The threads are kept open using a while(no signet), live and next flags are used to call the functions within the threads.

Both Client and Server has dedicated threads to handle parallel input streams, and read and write functions are synchronized so as to not bleed into chat functions.

Although there are still syncing issues with read functions.

Instructions
A make file is included in the project file that you can call within the terminal via “make”
Both server and client as well as any dependencies are compiled and ready to use.