

Lorenzo Zafra - 1395521

CMPUT 379 Assignment 3 Project Report

Objectives:

The objective of the assignment is to learn how to develop a client-server program and the quirks that comes with such programs. This assignment specifically focuses on sockets for communication. In addition, this assignment teaches us how to catch and control signals with signal handlers. This assignment also taught us how to do various tests and experiments that deals with a client-server program as it can be quite tedious.

Design Overview:

- .c files - a3chat.c, server.c, client.c
 - server.c contains all the functions and functionality that deals with the server.
 - client.c contains all the functions and functionality that is needed by the client.
 - a3chat.c is the main program which initializes either the server or the client.
- Created a struct "conn" which holds various information about the clients and its connection. The server looks at this struct to find the fd, username, start time, consecutive keep alive misses, last activity and recipients of a specific client.
- Created an enum for all the error codes that a user may encounter.
- To pass messages between clients and servers, the client is passing a message in a custom format which is then parsed by the server.
- The server 'defragments' its pollfd array when a client disconnects.

Assumptions:

- Each user can only have 10 recipients (can be changed in server.h) per session. To refresh this list, they would need to close the session and re-open a new one.
- The max username length is 30 characters
- Maximum server message length is 240 characters
- Maximum user input is 512 characters
- Maximum number of clients is 5.

- Maximum number of crashed clients for activity report is 30. Can be changed in server.h

Project Status:

At time of submission, the assignment is working as specified. I had trouble figuring out how to defragment the connections array and the pollfd array whenever a client in the middle of the array disconnects.

I had issues when a client is sending a message to a client that has crashed, and the server hasn't detected the crash yet.

Another difficulty I was having is when the server is already serving the max amount of clients (say 5) and another client (6th) tries to connect. This client is not given an error but will wait for a client to disconnect. To fix this issue, I let the 6th client connect then the server sends it a message that the server is full.

Also, I was having issues when a client unexpectedly disconnects from the server and closes the socket on the client end but the socket is still open on the server end. The server then fails to read the corresponding file descriptor. I fixed this issue by using shutdown() instead of close().

The last difficulty I had with the project is in regards to the activity report. Poll() is returning -1 when alarm() sends the SIGALRM signal. I learned that this happens during poll() or read(), etc and is interrupted by a signal. I chose to ignore the EINTR errno.

Testing and Results:

With the help of 'tmux', I was able to multiplex multiple terminals and open multiple clients to test the functionalities of the program. From there, I would do various tests like messaging, closing, exiting, re-opening and adding recipients etc. Testing was successful and at time of submission I believe that there are no bugs.

I also tested that the client can connect from different hosts by ssh'ing to various lab machines.

In addition, I tested the client crashing by sending the SIGINT alarm. I would then wait for the server to count 1.5*5 seconds and see if the

server detects the crash since the client has not sent KEEPALIVE messages.

Also, to test that my signal handler to print the activity report is set properly, I applied the handler to the SIGINT signal and pressed "Ctrl+C" to see if it would print the report. Both of the tests here and above worked.

Acknowledgements:

- APUE 3rd Edition
- <https://linux.die.net/>
- <http://man7.org/linux/man-pages/man3/lockf.3.html>
- <http://beej.us/guide/bgnet/output/html/multipage/pollman.html>
- <http://www.cs.cmu.edu/afs/cs/academic/class/15213-f99/www/lab5-fifo.c>
- <http://pubs.opengroup.org/onlinepubs/7908799/xsh/open.html>
- <http://www.cs.rpi.edu/courses/fall96/netprog/lectures/html/ipc.html>
- <http://stackoverflow.com/questions/1716296/why-does-printf-not-flush-after-the-call-unless-a-newline-is-in-the-format-string>
- <http://www.csc.villanova.edu/~mdamian/threads/posixthreads.html>
- <https://www.linuxprogrammingblog.com/code-examples/sigaction>
-

How to Run:

To run the server:

```
`./a3chat -s [portnumber] [num clients]`
```

To run the client

```
`./a3chat -c [portnumber] [serveraddress]`
```