

LINUX CHAT USER MANUAL

VIVEK KALIA

TYLER TREPANIER-BRACKEN

TABLE OF CONTENTS

Contents

Purpose and Usage_____	1
Purpose _____	1
Usage_____	1
Basic Client Usage _____	2
Basic Server Usage _____	3
Installation _____	4
Installing the client_____	4
Installing the Server _____	5
Potential Issues:_____	6
Conclusion_____	7

Purpose and Usage

PURPOSE

This Linux application is a world class chat application that was originally created for the purposes of completing the requirements for assignment 3 for the course COMP4981.

This application showcases the usefulness of using the 'select()' functionality in the form of a multi-user chat system with a central server. After connecting to a server you are entered into a chatroom that anyone connected to the same server can talk in.

USAGE

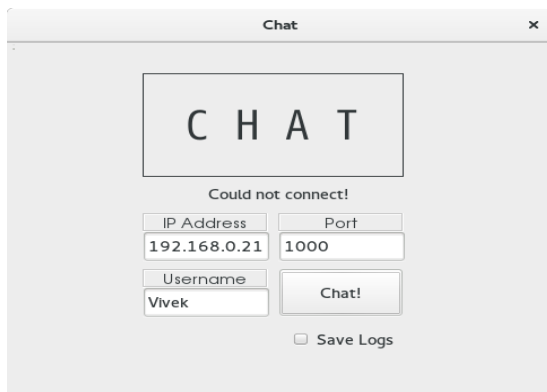
To use this application you must have a server running on a networked machine, as well as two to many client applications that are connected to the server. Once connected you will have access to chat with other users until your heart's content!

LINUX CHAT

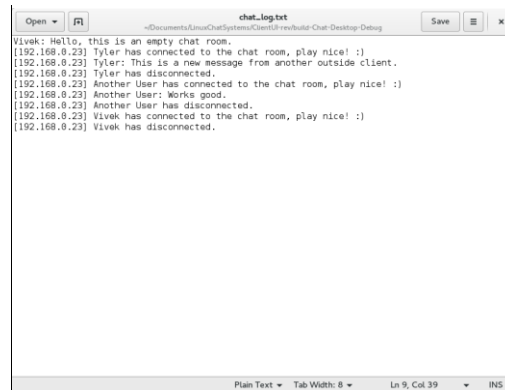
Basic Client Usage

After starting the client you will be prompted to enter in three inputs: the IP address of the server, the port the server is running on, and the username you would like to use for chatting. There is also a checkbox option for saving logs of your chat conversations. Make sure to enter your credentials correctly or you will get an error message and be unable to connect.

Example of error message:

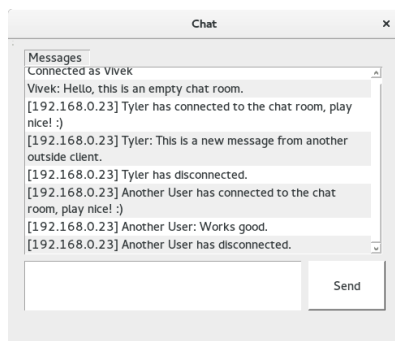


Example of log file:



After successfully connecting to the server you are then taken to a chat room where you can talk with other people. New users will prompt a connection notice, and exiting users will prompt a disconnect notice. Also all other participants IP addresses will be displayed.

Example of chatroom:

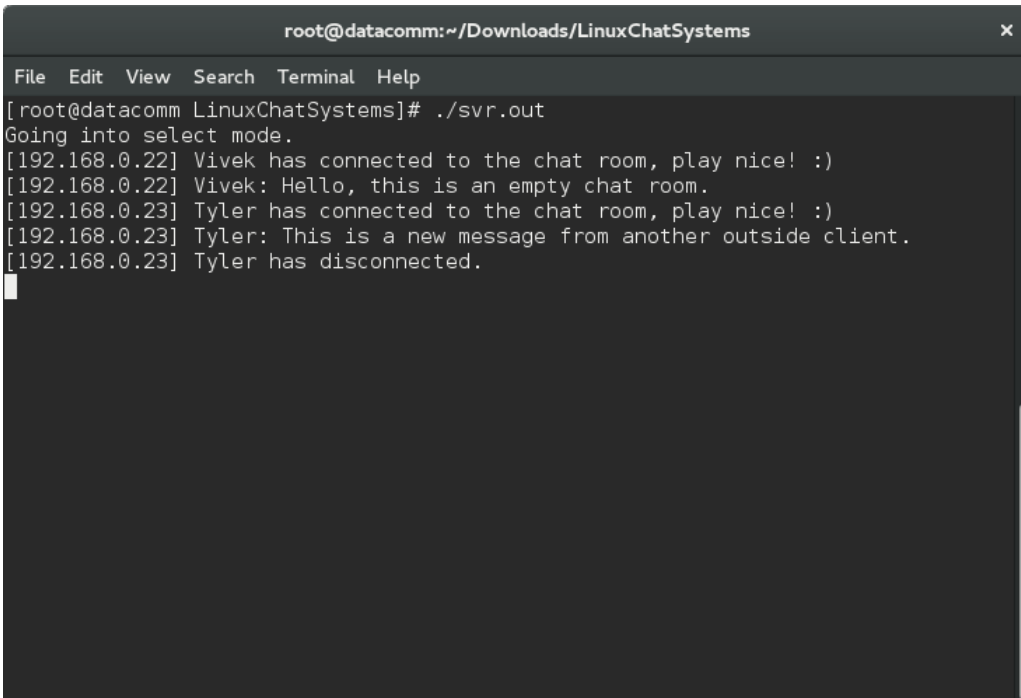


LINUX CHAT

Basic Server Usage

The server is a basic command line application there is not much direct user management or use of the server. When running the server you can specify a specific port number, however it is mainly doing work in the background processing requests, and logging them in the prompt.

Example of running server:



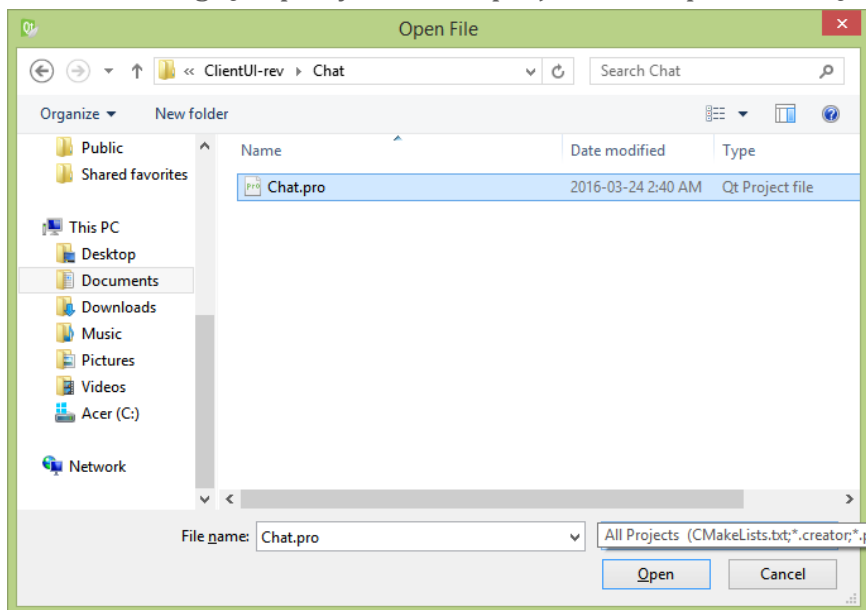
```
root@datacomm: ~/Downloads/LinuxChatSystems
File Edit View Search Terminal Help
[root@datacomm LinuxChatSystems]# ./svr.out
Going into select mode.
[192.168.0.22] Vivek has connected to the chat room, play nice! :)
[192.168.0.22] Vivek: Hello, this is an empty chat room.
[192.168.0.23] Tyler has connected to the chat room, play nice! :)
[192.168.0.23] Tyler: This is a new message from another outside client.
[192.168.0.23] Tyler has disconnected.
```

Installation

Installing both the client and server is very easy if you are building and compiling in QtCreator. After building all you have to do is run the executable or run from QtCreator itself.

INSTALLING THE CLIENT

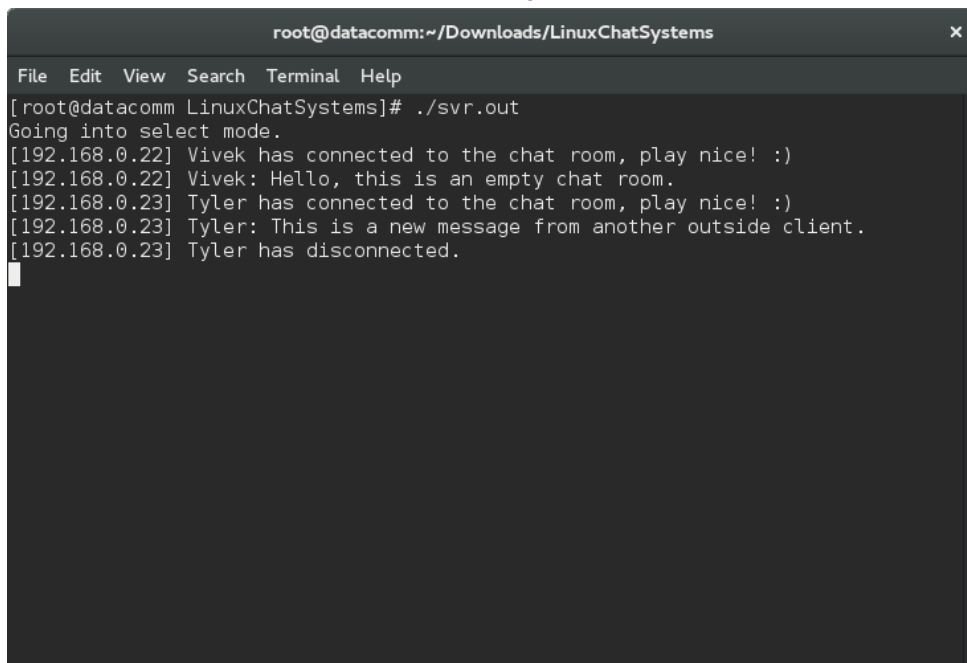
1. Boot up a Linux machine.
2. Download QtCreator from <http://www.qt.io/download/> (please select a suitable download for your use).
3. After installing Qt, open your client project's Chat.pro file in Qt and build.



4. You can look for the executable in the debug folder and run, or run straight from QtCreator.

INSTALLING THE SERVER

1. Boot up a Linux machine.
2. Navigate to your server folder and enter 'make' in the terminal.
3. This will create an executable file that you can run in the terminal.



A terminal window titled "root@datacomm: ~/Downloads/LinuxChatSystems" with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal output shows the execution of the server script: [root@datacomm LinuxChatSystems]# ./svr.out. The script enters "select mode" and logs several events: a connection from 192.168.0.22 (Vivek), a message from 192.168.0.22 ("Hello, this is an empty chat room."), a connection from 192.168.0.23 (Tyler), a message from 192.168.0.23 ("This is a new message from another outside client."), and a disconnection from 192.168.0.23 (Tyler).

```
root@datacomm: ~/Downloads/LinuxChatSystems
File Edit View Search Terminal Help
[root@datacomm LinuxChatSystems]# ./svr.out
Going into select mode.
[192.168.0.22] Vivek has connected to the chat room, play nice! :)
[192.168.0.22] Vivek: Hello, this is an empty chat room.
[192.168.0.23] Tyler has connected to the chat room, play nice! :)
[192.168.0.23] Tyler: This is a new message from another outside client.
[192.168.0.23] Tyler has disconnected.
```

Notice the “./svr.out” that is the executable.

4. (Optional) You can specify a port number when executing your server.
(ie. “./svr.out 9999”)

Potential Issues:

1. There can be more than one user with the same name chatting. While it shouldn't be a huge deal as the user's IP addresses are displayed, there can be a possibility of a person imitating another person which is simply just annoying
2. When you are connected as a client to the server and the server shuts down your client application will shut down also, there is currently no warning for why your client will close.

Conclusion

We hope that you have a pleasant experience using this application, it is an open source project available for editing by anyone!

<https://github.com/Tmanthegamer/LinuxChatSystems>