Mahidhar Reddy Dwarampudi | CSE | November 8, 2016

Multi Group Chat Server

Computer Networks project

# Abstract

This is a local multi group chat server. As the name suggests, it is a simple group chat server where a group of people can create groups and chat on them.

# Introduction

The clients connect to the main chat server. They create chat groups on it. To which they connect and chat on them.

In the group chat server, the server is setup on a local machine with a certain port number. The IP address and port number are shared with the people who want to chat. People joining the group will join with a temporary username. They can send messages to the group and receive unread messages from the group. They can exit the group when they are done.

# Problem Statement

Make a server which manages many group chats.Make a server group chat server, which can take new clients. Only people on the group can see the messages sent by other members of the group and be able to send messages to the group. Clients can also join other groups by searching them.

# Objectives

Make a main server program which manages group chats. Make a server side c program and a client side program. The server should be able to manage all the messages sent be all the clients. The Client should be able to receive and send messages from the client.

# Background

The main server uses UDP protocol to which the client sends messages to create a group a or searches for a group and joins the group. Group Server and client are implemented using TCP protocol for chatting. Library functions such as select(), send(), recv(),sendto(),recvfrom(),sprintf() and atoi() are used. These are standard library functions and the code compiles and runs on any system running on Linux and has a C compiler. The system should also have networking enabled.

# Implementation

The main server uses UDP protocol to which the clients send messages to create new groups or join existing groups while it returns the group port number to which they can join. The group server uses TCP protocol.

The server simply implements a TCP server and binds to a certain port for connections. It keeps listening. All the connections are added to master file descriptor where it keeps waiting for data. Even the listener is added to select, so that when a client asks for a new connection it can read from the listener and add the client to the group. It takes messages from all the clients and stores them. Also it maintains a record of what client has received what messages.

While the client is a simple client. It connects to the server and it can send messages and receive unread messages from the server.

The server should be called with port number and IP address arguments. The client explicitly the IP address and the port number of the server to connect to it. Errors are shown accordingly in both the programs.

# DFDs



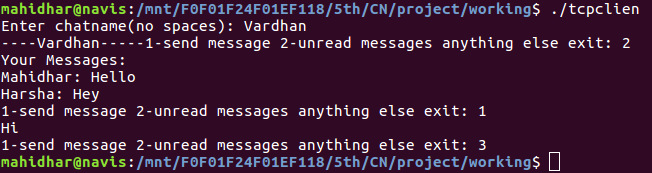
LEVEL O DFD

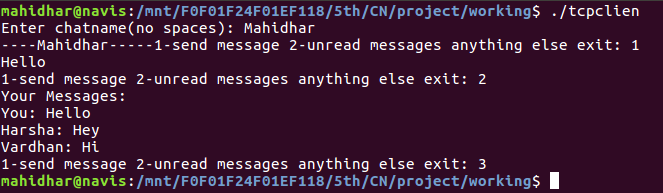


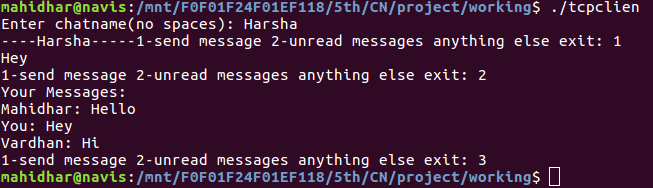
LEVEL l DFD

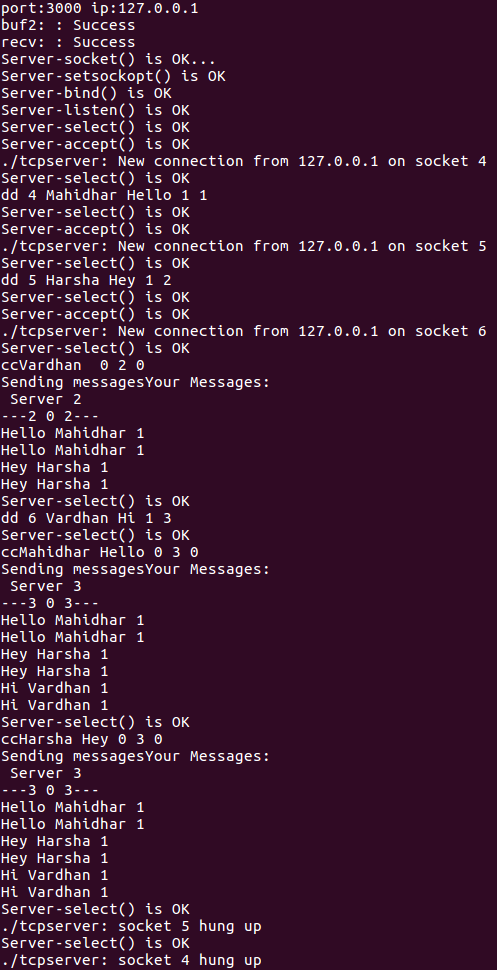
# Results

The following images show the working of both server and client. It is 3 clients connected to the server sending and receiving messages.









# Limitation

* Each group can have up to only 1000 clients.
* The client user name cannot be more than 25 characters
* The message sent by the client should not contain more than 1024 characters.
* The main server can’t hold more than 1000 chat servers.
* The chat server’s name shouldn’t be more than 25 characters.

# References

* Basic Linux Programming, 4th Edition by Neil Matthew & Richard Stones
* Beej’s Guide to Network Programming by Brian “Beej Jorgenson” Hall

