

Multichannel Chat User Guide

Intro: This is an activity for our subject network computing. We decided to do this user guide in English with the purpose of improving our English while practising the socket programming. As a professor you can request to us the translated copy of this guide if it is more comfortable to you.

Introducing the nickname as we will see it is better to not using '/' and '!' as starter characters but if they are used there should be no problem.

How does it work?

There are few commands, just the necessary in order to get an idea of what a multichannel chat is and how it work. There is no fiability at the moment of repeating commands they are the following:

(We reserved the character '/' and '!' in each message to interpret commands)

/nc yourChannel

This command creates a channel with the name that you have indicated but the channel must be without any blank space otherwise we cannot grant the wanted performance.

/join Channel

This command is used to join the wanted existent channel. (Note: if you do this command twice with the same client, you will receive doubled messages).

/Channel "Entry your message."

Anytime you want to send a message into a channel you must type this command. It is not necessary to be connected to a channel in order to send messages into it but we are assuming no one will use this command without being logged into a channel, pure logic.

!users

List the online users.

!channels

Lists the available channels.

/fc

Ends the communication with the server. Disconnects the client from the server.

Known errors of fiability

First of all, nicknames with exactly the same string will give problems of replicating messages or even worse failures. Secondly, using already used strings for channels will also give inconsistent data to the program and it will create two channels so in this case we cannot grant the wanted performance. Finally the server will serve forever so you have to kill the process with Ctrl+c otherwise you can connect the clients you wish.

Others

Private messages are not implemented just because we ran out of time not for the added difficult it was supposed to be.

The server outputs data when a client connects. This can be used to monitor the processes you are running in execution time.

When you use the general channel you do not receive echo from the server in contrast if you are logged in a channel so you can check all the messages sent there. The general channel was thought only to enter commands on not oriented communications on any specific topic.