

SSMUAE Robert McIntyre

p149064

by Robert McIntyre

FILE	2783_ROBERT_MCINTYRE_SSMUAE_ROBERT_MCINTYRE_P149064_113614_1569337231.DOCX		
TIME SUBMITTED	10-MAY-2018 11:56PM (UTC+0100)	WORD COUNT	708
SUBMISSION ID	88922728	CHARACTER COUNT	3319

Robert McIntyre

P149064

System Software & Multi-User Applications
Engineering

TCP/IP Client Server Implementation and
Evaluation

Dr Tim Bashford

Table of Contents

No table of contents entries found.

Introduction

In the report I have described the parts of the assignment that have been attempted. In other sections I have described what I would have done if there was more time.

The Client

The Client uses sockets in a different way to the server. The client starts the communication between it and the server. The client has four major steps to connect with the server using TCP. These are creating the socket ready to connect to the server, establishing a connection, communicating with the server and closing all sockets.

User Joins server

The user joining the server is establishing a connection to the server client. This is done by checking that a connection has been made. This done by using AF_INET, inet_pton, checking the connection socket and reading the information from the server. Join object are created. Stream sockets are used in the TCP protocol as they provide reliable data streams. Sequence packed sockets could have been used. The program uses sockaddr_in which uses ipv4. Once the user has connected to the server the socket is destroyed. There are objects created to manage the connections and managing the sockets. This is done in most sections.

User Leaves Server

If the user decides to leave the server, the user is remove from a list. Before the user is disconnected the client sends a message to the server saying that they have left. This message includes the users nickname which has been converted to char*. The client needs to find out the number of bytes to send. The client checks to see if there is a message received from the server using the readn object. This reads everything from the buffer. Then checks whether the nickname has been removed from the list and safely deletes the objects.

Users sends a message

The process of sending a message is like sending a leave message. The QString needs to be converted to a char* for the message to be sent. The client will read a message that is sent from the server reading the number of bytes the server has sent.

Client Update

The Server

Listens for messages

The Server listens to the client to see if a message has been received. Server is placed in the listening mode and waits for a message. Once received the server will count the amount of bytes.

Receive Join Message

Pipes are used to relay the processes using the fork method. One process goes in and two process that are the same come out. The server uses a full duplex pipe where bytes can be sent both ways,

however, only one transmission can be sent at a time. In the pipes method if there is a -1 an error has occurred. A zero is returned for success. Once the server receives a message to join waits for a connect from the client. There is a timer to wait for a certain time. Pre-forking might have been a good idea for both the server and the client as it would create many processes to take care of all the threads. Every child process is iterative server. An alternative method would be to use dynamic process allocation if there is not a set number of processes. There should be a method of including one child process to be sent or received at a time such as semaphores and mutexes.

Receive Leave Message

Receive a Chat Message

When receiving a chat message threads will be used to pass information. Either joinable or detached threads are used. Loops are used to give threads a fair amount of use to all threads. The time slice will be measure by a certain amount of time.

Receive update request

Conclusion

The assignment was extremely challenging. Initially there were issues with 'Unicode 8' which meant that the program would not run. There was a way to work around the problem such as using QT 4 instead of QT 5. It was difficult to understand where to start with the project. Having finally understood how to start there was not enough time to finish the assignment.

FINAL GRADE

GENERAL COMMENTS

Instructor

33/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

GRADING FORM: SMU GENERIC_GJ

ROBERT MCINTYRE

33

RESEARCH-INFORMED LITERATURE



A reasonable, if slightly brief report - though a shame no sources were used.

2

KNOWLEDGE AND UNDERSTANDING



Some evidence of knowledge and understanding through the report and implementation.

5

ANALYSIS



Limited conclusions drawn, though a lack of synthesis due to the lack of sources.

4

PRACTICAL APPLICATION AND DEP.



An attempt made but, as you mention in the report, a long way from complete. A few errors generally, e.g. in //TODO Write size of encoded object to stream `bytesToSend = j->encode(&buffer);` You correctly call `encode` and get the size, but never actually send it. This means that, on the next line when you call: `size_t buffLength = myConnection->writen(buffer,bytesToSend);` You're sending the encoded object without having actually sent the size. The server'll then crash. I assume you were mid-debugging this, given the exit statement right after! Still, some progress made.

16

SKILLS FOR PROFESSIONAL PRAC.



A fairly well-written report with good spelling, punctuation and grammar.

6