BTN415 Lab 2 – TCP/IP Servers

In this lab, you will create your own TCP/IP server to accept connections from clients, receive messages, and send messages back, following a number of different scenarios.

# LEARNING OUTCOMES

Upon successful completion of this lab, you will have demonstrated the ability to:

* Create a server application
* Implement a reliable data communication using the TCP/IP standards and protocols

## For this lab, you should use as a starting point the code that was discussed during our lecture on TCP/IP Servers. This code can be downloaded from our course’s Github repository using this link: <https://github.com/marceljar/BTN415_Labs/blob/main/lab2/server.cpp>). After downloading the source code, you should modify the code in order to achieve what is asked in what follows.

## PART A – [1.5 marks]

## Update the source code in order for the server to be able to keep receiving messages from the client, as well as sending replies, with the message “Thanks for your message!”, after each received message.

**Before:**

A screenshot of a computer

Description automatically generated with medium confidence

**After:**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Output:**

**Graphical user interface

Description automatically generated**

## PART B – [1.5 marks]

## Update the source code in order for server to be able to keep receiving messages to the client, as well as sending replies, with the message “Thanks for your message!”, after each received message, until the client sends a message containing only: "[q]" (i.e., the sequence of characters [, q, and ]).

**Before:**

**A screenshot of a computer

Description automatically generated with medium confidence**

**After:**

A screenshot of a computer

Description automatically generated with medium confidence

**Output:**

A screenshot of a computer

Description automatically generated with medium confidence

## PART C – [2.0 marks]

## Update the source code in order for server, created in Part B, to be able to connect with a new client after a previous client quits, instead of exiting. In other words, after closing the socket from one client, the server should make itself ready to accept a connection from another client. *Hint: In order to accept new connection, the listening socket needs to be kept open.*

**Before:**

**A computer screen capture

Description automatically generated with medium confidence**

**After:**

A picture containing text, screenshot, monitor, electronics

Description automatically generated

**Output:**

**Graphical user interface, text

Description automatically generated with medium confidence**

# SUBMISSION INSTRUCTIONS

*Note that these instructions are in an increasing order of difficulty. You only need to submit one* ***cpp*** *file containing your source code, which is the one that solves the most difficult part you have had success or got the closest to the solution. For example, if you finished part C, simply submit your source code for this part. If you got stuck on part B, then submit your source code for this part.*