

Lab5: Part 2 (Programming)

Due Sep 23, 2020 by 11:59pm **Points** 12 **Submitting** a website url

Available until Sep 24, 2020 at 12:05am

This assignment was locked Sep 24, 2020 at 12:05am.

Creating a Basic Server

Submit via **CODE**

Due: Wednesday, September 23 @ 11:59 PM

[12 points for program]

In C++, we can create a basic server application using the Boost C++Library that communicates with a client after connecting. In this lab, you will create a server that returns the current date and time to a client. The server will be written in C++ and the client will be run as telnet to connect to the server at some endpoint.

The accompanying cpp file ([lab5.cpp](#)

<https://miamioh.instructure.com/courses/129645/files/17445391/download?wrap=1>) 

https://miamioh.instructure.com/courses/129645/files/17445391/download?download_frd=1) has the skeleton program with comments regarding what you need to implement in order to create the server.

The server should do the following:

1. Take a runtime argument for the port number (e.g., ./lab5 42000) **[2 points]**
 - Make sure you validate that a user supplied the port number and if not print “Must supply port number” and return -1
 - You can use result to have your program terminate as well returning 0 from main()
2. The program should also validate that the user supplied a valid input **[2 points]**
 - It should validate the user entered a valid number by completing the checkPort() function
 - If casting the number fails, the exception should print "Must supply port number" to standard error and return 0
 - Check that the port number is not a privileged port
 - If the user enters an invalid port number, print “Cannot use privileged ports” and return 0
3. Create the endpoint using user-defined port number **[2 points]**

4. Create the socket [2 points]
5. Wait listening for a client to connect [2 points]
6. Send the data/time to the client [2 points]
7. Disconnect

Note: Your main method should ALWAYS be returning 0. If it returns ANY other value, it is NOT correct.

To test connecting to the server, you will use telnet as the client, which will create a socket implicitly and connect to it then receive the data from the server. From the VM, you should use "telnet 127.0.0.1 [portNum]" to connect **when the server is running**, where [portNum] is the number of the port you gave to the server program.

Hint: You may use the course materials on networking, as well as the boost::asio documentation. Also, the stream you write to is the socket.

Some documentation links:

https://www.boost.org/doc/libs/1_66_0/doc/html/boost_asio/reference/ip_tcp.html ➞

(https://www.boost.org/doc/libs/1_66_0/doc/html/boost_asio/reference/ip_tcp.html)

https://www.boost.org/doc/libs/1_66_0/doc/html/boost_asio/reference/ip_tcp/acceptor.html ➞

(https://www.boost.org/doc/libs/1_66_0/doc/html/boost_asio/reference/ip_tcp/acceptor.html)

https://www.boost.org/doc/libs/1_66_0/doc/html/boost_asio/reference/ip_tcp/endpoint.html ➞

(https://www.boost.org/doc/libs/1_66_0/doc/html/boost_asio/reference/ip_tcp/endpoint.html)

https://www.boost.org/doc/libs/1_66_0/doc/html/boost_asio/reference/ip_tcp/socket.html ➞

(https://www.boost.org/doc/libs/1_66_0/doc/html/boost_asio/reference/ip_tcp/socket.html)

https://www.boost.org/doc/libs/1_66_0/doc/html/boost_asio/reference/write.html ➞

(https://www.boost.org/doc/libs/1_66_0/doc/html/boost_asio/reference/write.html) ➞

(https://www.boost.org/doc/libs/1_66_0/doc/html/boost_asio/reference/write.html)

SUBMISSION NOTE:

The CODE plugin will only be able to validate that the code can compile, and that it passes style checking.. It cannot execute a client process to connect and validate the server's behavior. Make sure you test the application with telnet as described above!

EXTRA CREDIT:

[2 points]

In question 2 of **Part 3** from the *written assignment*, you observed a certain (perhaps, initially unexpected) behavior. Modify the code to address what you observed (i.e., for some n uses of telnet, the results should be consistent).

Criteria	Ratings		Pts
Take a runtime argument for the port number. Make sure you validate that a user supplied the port number and if not print "Must supply port number"	2 pts Full Marks	0 pts No Marks	2 pts
The program should also validate that the user supplied a valid input. It should validate the user entered a valid number by completing the checkPort() function. If casting the number fails, the exception should print "Must supply port number" to standard error and return 0. Check that the port number is not a privileged port. If the user enters an invalid port number, print "Cannot use privileged ports" and return 0.	2 pts Full Marks	0 pts No Marks	2 pts
Create the endpoint using user-defined port number	2 pts Full Marks	0 pts No Marks	2 pts
Create the socket	2 pts Full Marks	0 pts No Marks	2 pts
Wait listening for a client to connect	2 pts Full Marks	0 pts No Marks	2 pts
Send the data/time to the client	2 pts Full Marks	0 pts No Marks	2 pts
In question 2 of Part 3 from the written assignment, you observed a certain (perhaps, initially unexpected) behavior. Modify the code to address what you observed (i.e., for some n uses of telnet, the results should be consistent).	2 pts Full Marks	0 pts No Marks	2 pts
Total Points: 14			

