Assignment 5

Adv C Programming Comprehensive Assignment

Title: Mastering Advanced C Programing Concepts - Socket API

Objective: The objective of this assignment is to reinforce your understanding of advanced C programming concepts.

Assignment Task:

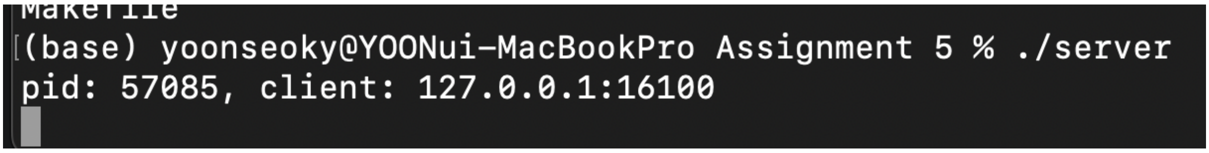
Implementing a Server-Client program based on the Echo Server example using the socket API in the lecture sides.

The Socket API was introduced in the lecture and a simple Echo Server example demonstrated a TCP server-client implementation in C. This involves two main components: a server program that listens for incoming connections and a client program that initiates a connection to the server.

Using the example, you should implement a server-client program to send a message (max 50 characters) from a client to a server. The server receives it and print to the terminal. The server also can send a message back to the client.

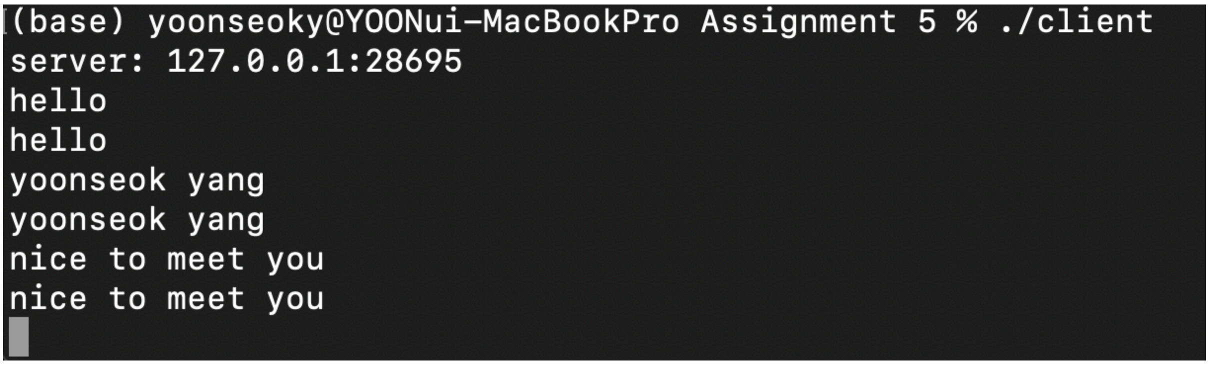
In the server-client program, you need to open two terminals and execute the client and the server, respectively. For example, when you execute the echo server example, you can observe this behavior below on the server and client terminals.

Server:



```
Makefile
(base) yoonseoky@YOONui-MacBookPro Assignment 5 % ./server
pid: 57085, client: 127.0.0.1:16100
```

Client:



```
(base) yoonseoky@YOONui-MacBookPro Assignment 5 % ./client
server: 127.0.0.1:28695
hello
hello
yoonseok yang
yoonseok yang
nice to meet you
nice to meet you
```

Download assignment5.zip that includes server.c, client.c, util.c and util.h. Compile the code with your implementation and execute it. You should capture a screenshot of the execution result from the C code.

Instructions for the implementation

1. Error handling:

- Correct or insert any missing error cases and handle properly. You can print a error message in the error cases.

2. Exit condition:

- When sending a “exit” message from the client or the server, exit both of them.
- If the message is more than 50 characters long, print an error message and terminate both the server and the client.

3. Comment:

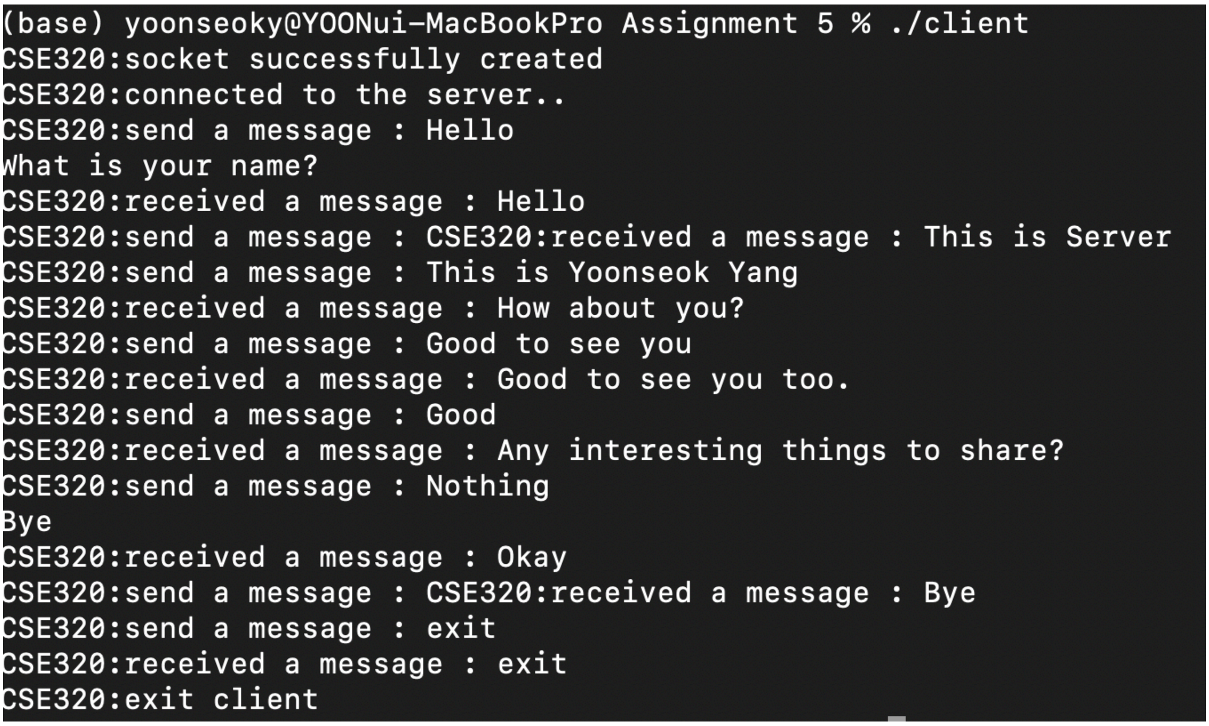
- Add comments on any of your lines. Describe why you added them in details.

4. Compile:

- Use Makefile to compile: Ex) make server; make client

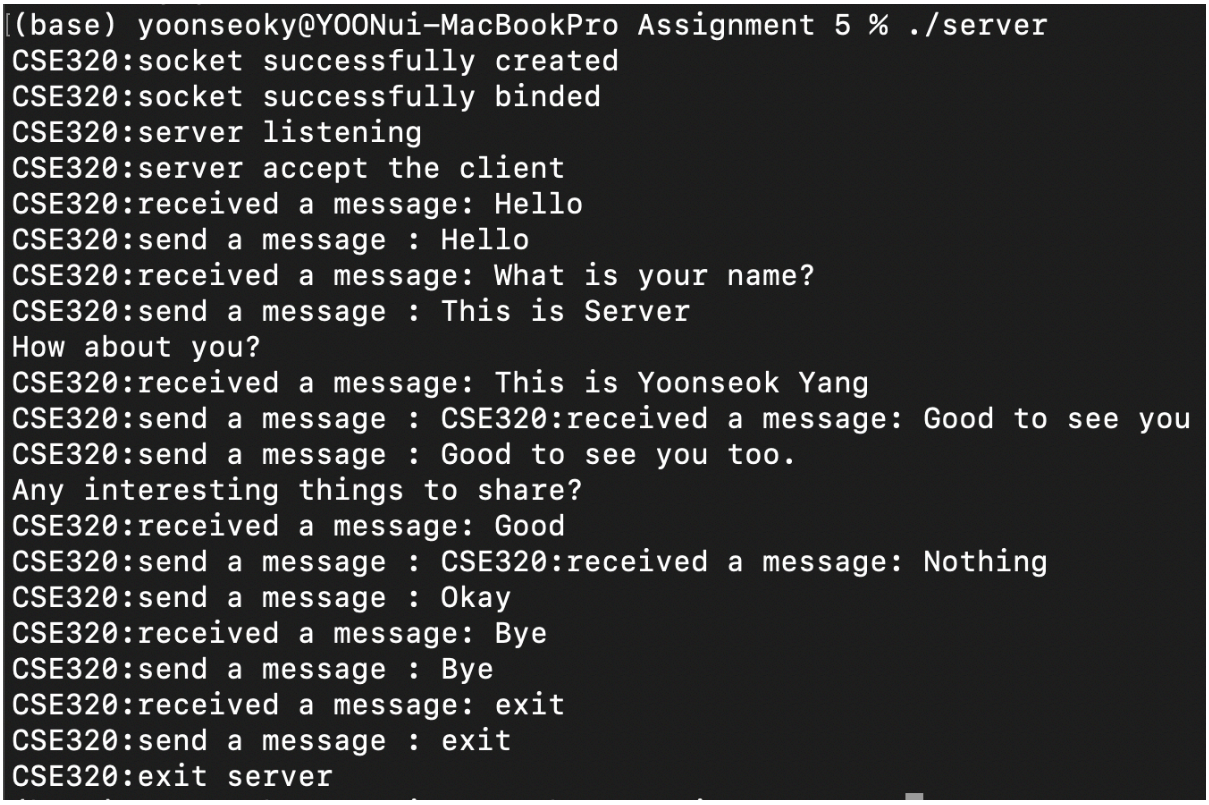
<u>Example result</u>

Client:



```
(base) yoonseoky@YOONui-MacBookPro Assignment 5 % ./client
CSE320:socket successfully created
CSE320:connected to the server..
CSE320:send a message : Hello
What is your name?
CSE320:received a message : Hello
CSE320:send a message : CSE320:received a message : This is Server
CSE320:send a message : This is Yoonseok Yang
CSE320:received a message : How about you?
CSE320:send a message : Good to see you
CSE320:received a message : Good to see you too.
CSE320:send a message : Good
CSE320:received a message : Any interesting things to share?
CSE320:send a message : Nothing
Bye
CSE320:received a message : Okay
CSE320:send a message : CSE320:received a message : Bye
CSE320:send a message : exit
CSE320:received a message : exit
CSE320:exit client
```

Server:



```
[(base) yoonseoky@YOONui-MacBookPro Assignment 5 % ./server
CSE320:socket successfully created
CSE320:socket successfully binded
CSE320:server listening
CSE320:server accept the client
CSE320:received a message: Hello
CSE320:send a message : Hello
CSE320:received a message: What is your name?
CSE320:send a message : This is Server
How about you?
CSE320:received a message: This is Yoonseok Yang
CSE320:send a message : CSE320:received a message: Good to see you
CSE320:send a message : Good to see you too.
Any interesting things to share?
CSE320:received a message: Good
CSE320:send a message : CSE320:received a message: Nothing
CSE320:send a message : Okay
CSE320:received a message: Bye
CSE320:send a message : Bye
CSE320:received a message: exit
CSE320:send a message : exit
CSE320:exit server
```

Documentation

- Document your code thoroughly with comments explaining each section.
- Prepare a report PDF document summarizing everything such as each step or functions that you implemented, challenges faced, and what you've learned during this assignment.

Submission Guidelines:

- Submit the consolidated source code (a zip file) for the entire assignment.
- Include a README file with instructions for compiling and running your program.
- Ensure your code is well-documented with comments.

Assessment Criteria:

Your assignment will be assessed based on the following criteria:

- Submit your own work (80% in points) even though it doesn't execute as expected.
- Proper use of C data types, pointers, arrays, structures, and dynamic memory allocation.
- Efficiency and readability of the code.
- Effective use of memory, functions, and function pointers.
- Proper error handling and validation.
- Utilization of macros for code optimization.
- Utilization of C standard library functions.
- Quality and completeness of documentation.

Important Note: **Plagiarism will not be tolerated, and students are expected to produce their work independently.** Please perform this assignment by yourself and don't copy any solutions from any Generative AI applications like ChatGPT or Bard, your friends or online. If I find any things that imply plagiarism, you will lose the whole points and be reported.