Introduction to Computer Networks

Lab (Deadline: will be announced on the eLearn system)

1. Description

Implement a simple number guessing game. The server should allow clients to guess a number between 0 to 999, and reply a guessing result. The client should be able to access the server, guess a number, and read the result replied by the server. Please use TCP socket, and the program should be written in C or C++. You are also required to use Wireshark to capture the packets transmitted by the server and the client, observe the result, and answer the questions.

2. Socket programming

2.1 Server Program

- File name of the server program should be named "(studentID)_ser.c" or "(studentID)_ser.cpp".
- The server program is executed using the command line by typing "(studentID)_ser (port)", where "(port)" is a port number.
- Functions of the server program:
  - Generate a random number for clients to guess.
  - Allow clients to access the server.
  - Show a prompt, and let the client guess a number.
  - Accept a client's request and receive the guessing number from the client.
  - Respond the result to the clients.

2.2 Client Program

- File name of the client program should be named "(studentID)_cli.c" or "(studentID)_cli.cpp".
- The client program is executed using the command line by typing "(studentID)_cli (ip) (port)", where "(ip)" is the IP address of the server, and "(port)" is a port number.

- Functions of the client program:
  - Should be able to connect to the server.
  - Should be able to send a guessing number to the server.
  - Should be able to receive a response from the server.

2.3 Note

- For simplicity, we assume that there is only one client at a time.
- All data must be transmitted via TCP socket.
- Use fixed port numbers on both the client side and server side.
- When the server receives the correct answer, it should response “Answer Correct” to the client and restart the game.
- Use “winsock2.h”, “socket.h” or any other socket library you can find.
- Plagiarism is not allowed.

3. Wireshark

- Capture the packets transmitted by the server and the client.
- Observe the result and answer the following questions.
- The report should include:
  - Show the packets used for TCP hand shaking.
  - Show the server’s IP and the client’s IP.
  - Show the server’s port and the client’s port.
  - Show the size of the packet transmitted by the client. (in bytes)
  - How many routers does each of the transmitted packets goes through. (Hint: TTL)

4. Evaluation

- (60%) Server program and client program.
- (20%) The answers to the Wireshark observation.
- (20%) Report.
  - File name of the report should be “(studentID)_report.pdf”. The report should include:

- ✦ Details of your implementation, including server-side and client-side.
- ✦ Step-by-step screenshots and explanations of the execution of each function.
- ✦ The answers to the Wireshark observation.
- ✦ Descriptions of difficulties you encountered and your solutions.

5. Submission

- Please upload the following files to the eLearn system.

  (1) "(studentID)_ser.c" or "(studentID)_ser.cpp"

  (2) "(studentID)_cli.c" or "(studentID)_cli.cpp"

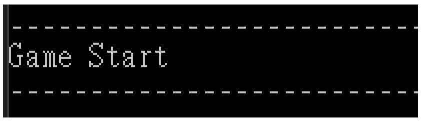
  (3) "(studentID)_report.pdf"

- All the files should be included, otherwise no grade will be given for the programming lab.
- Deadline: will be announced on the eLearn system.

6. Sample Screenshots

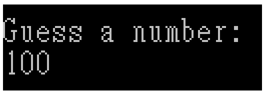
You don't need to follow the sample exactly.

- **Game Start:**



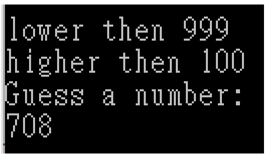
```
-------------------------
Game Start
-------------------------
```

- **Guess a number:**



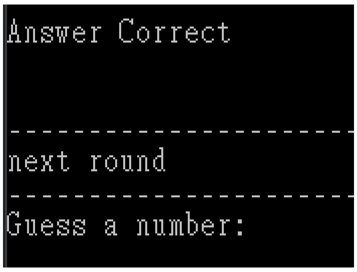
```
Guess a number:
100
```

- **If it is wrong, output the new range and guess next number:**



```
lower then 999
higher then 100
Guess a number:
708
```

- **If it is correct, output "Answer Correct", and create a new game.**



```
Answer Correct


----------------------
next round
----------------------
Guess a number:
```