Project 2: A simple application level protocol

Please upload your files to **CSNS before 23:59:59pm, 12/13/14.**

Your submissions should include:

1. *The source code of the client and server programs*
2. *Documentation*
    a. Introduction. What do your programs do? How the client and the server interact with each other (what is the protocol)?
    b. Implementation. Though java is preferred, you can choose to use c/c++. Specify the language and the platform you use.
    c. How to compile/run the client and the server?
    d. Some screen shots.

One submission/group.
Total points: 6

In this project you need to write a client and a server program to implement a very simple application level protocol. Your client and server should work as follows:

1. Your **server runs first** and then waits for connections from your client on a port number you choose.
2. You client talks to the server by sending commands to the server:
    a. UPPERCASE
    b. LOWERCASE
    c. ~~GET~~ **REVERSE**
    d. EXIT

Initially, when the server accepts the connection from a client, the server should first reply with the message "server: got connection from clientx.x.x.x" where x.x.x.x is the IP address of the client. Then, It should reply with the message "Server is ready...". Both messages are displayed at the client side.

The following illustrates the protocol in detail.

UPPERCASE
This command requires the server to transform the ASCII text sent from the client to all upper-letter text. The ASCII text sent from the client is ended with "." in the last line. The server should reply "200 OK" and then return the text in uppercase to the client. A client-server interaction looks like (shown at the client side):

client: UPPERCASE
server: 200 OK
client: Hello.
client: Nice to meet you!
client: **.**
server: HELLO.
server: NICE TO MEET YOU!

At the server side, the server should display "x.x.x.x sends UPPERCASE" when receiving this command. For all other commands, the server should also do this kind of display.

LOWERCASE
This command is similar to the previous command except that the server sends lower case text back to client.

client: LOWERCASE
server: 200 OK
client: HELLO.
client: NICE TO MEET YOU!
client: **.**
server: hello.
server: nice to meet you!

REVERSE
When the server receives this command, it sends to the client the received message in the reverse order.
client: REVERSE
server: 200 OK
client: HELLO
server: OLLEH

EXIT
This command closes the connection. After receiving this command, the server should respond the acknowledgement "200 OK" to the client, display the "x.x.x.x sends EXIT", then close itself.

A client-server interaction thus looks like:
client: EXIT
server: 200 OK

If the *server* receives an invalid command, it should sends an error message to the client "400: Not a valid command!".