CS 4850/7850 Computer Networks I

Fall 2004

Sample Codes: Echo Server and Client

The step-by-step instructions for java socket and window socket are provided below. You can pick one of them. *If you are familiar with compiling and running Java or C socket programs, you can skip this instruction.*

Step By Step Instructions for Executing the Java Code

1. Download Code. Save *EchoServer.java* and *EchoClient.java* on your computer.
2. Modify the server port number to avoid conflicting with other students' server program. Use 1 plus the last four digits of your student ID as the server port number. For example, if the last four digits of your student ID is 3456, then as the server port number is 13456. The original server port number is 9999 at *EchoServer.java*. Save the result.
3. Modify the server port number at *EchoClient.java* too.
4. Log into your Bengal account and create a directory for *project0* using **mkdir**.
5. **ftp** both the EchoServer.java and EchoClient.java to the directory named *project0*
6. Open two windows of Bengal and go to the directory *project0*.
7. Once you are in the directory *project0*, in one window compile the server side using the command **javac EchoServer.java** and on the other window compile the client side using the command **javac EchoClient.java**. You should be able to compile them without any error or warning.
8. On the window with the server side program, type in the command **hostname** and hit enter. You will be able to see the name of the host, say *xyz*.
9. On the window with the server side, type the command **java EchoServer** and hit enter. You will be able to see "Waiting for a client to connect..." that means the server has started and is waiting for the client to send messages.
10. On the other window, type in the command **java EchoClient xyz** where xyz is the name of the host the server is running on and hit enter. You will get a prompt "Type whatever you want: " and you can type in anything and hit enter. You will be able to see "Server says : " with the same thing you typed and it will exit you from the client.
11. Your server will always be running till you exit it, but the client will stop after every time you type a line to send to the server. Hence for again sending a message to the server, on the window with the client side simple repeat step # 10.
12. When you are done, go to the window with the server running and use **ctrl C** to exit from the server.

Note: if you run server and client at different machine, find out the IP address of the server machine, say, 128.34. 29. 1, then run **java EchoClient 128.34. 29. 1**

Instructions for Using the C code

1. Assume you are at Lab 142 (any PC with Microsoft Visual C++ 6.0 will be fine).
2. Great a folder called D:\project0
3. Download *client2.zip and server2.zip* and save in the folder *project0*. Unzip them. You will get two folders: *client2* for the client and *server2* for the server.
4. Open Microsoft Visual C++. Use menu *file->Open Workspace.*
5. Go to the folder *client2*. Open *client2.dsw*.
6. Before you build it, modify the server port number at *client2.cpp* to avoid conflicting with other students' server program. Use 1 plus the last four digit of your student ID as the server port number. For example, if the last four digit of your student ID is 3456, then as the server port number is 13456. The original server port number is 9999.
7. Use menu *Build->Build client2.exe* to compile and build the client program.
8. In case you don't see any error in compiling go to step # 9. In case you see some errors in the window down below after the Build has finished. In that case click on *Project* and then *Settings* and you will see a new window for Project Settings. Click on the *Link* tab and in the box for *object/library modules* add WS2_32.lib and then repeat step # 7.
9. Repeat steps 4-8 with the server program too.
10. After successfully finishing the Build, you will be able to see a folder called *Debug* inside both *client2* and *server2* folders which contain the executable codes.
11. Go to Start menu and click on Run then type ***cmd*** and hit ok. Repeat this one more time and then you will have two shells running.
12. In the shell for executing the server, type in *cd D:\project0\server2\debug* which will take you to the server folder. You should find *server2.exe*.
13. Type the command *server2* and hit enter. You will be able to see "Waiting for a client to connect..." that means the server has started and is waiting for the client to send messages.
14. In another shell for executing the client, type in *cd D:\project0\client2\debug* which will take you to the client folder. You should find *client2.exe*.
15. Type in the command **hostname** and hit enter. You will be able to see the name of the host, say *xyz*.
16. Type in *client2 xyz*, where hostname is the name of the host you saw after you typed in hostname. You should be able to see a command called "Type whatever you want:"
17. Type in anything and the server will respond.
18. When you are done, go to the window with the server running and use **ctrl C** to exit from the server.

Note: if you run server and client at different machine, find out the IP address of the server machine, say, 128.34. 29. 1, then run **client2 128.34. 29. 1**