CSE 5306 Lab 1 - A Spelling Checker Client & Server Summer– 2017

Objectives:

1. Exposure to distributed applications
2. Understanding one communication protocol for distributed processing - Sockets
3. Programming GUI processes
4. Using an IDE

Due: Monday July 3, 2017, 11:59:59 pm

Project Specification:

Read this **entire** assignment before you start any work. These will be individual projects. You may write the program in any language or environment under any Integrated Development Environment (IDE) that you desire. Keep in mind that more help may be available to you in some languages than in others. Furthermore, available controls, objects, libraries etc. may make some of these tasks easier in one language than in another. Finally, for an IDE other than Visual Studio.net you will have to have that IDE available to demo to the TA. For example, you might bring a laptop to demo the program. Socket programming is so universal that you can probably find major portions of this part of the program with Google. Using code you find on the Internet is fine, but be sure to document the source in the writeup and in the program source code! Otherwise you may use code also found by another student and we will have no choice but to identify this as collusion.

You will write a client/server system that will check the spelling in a block of text. Your client process will connect to the server over a socket connection and send a block of text that the user has input. Your server will have a lexicon (list of words) that it will read from a file that will be supplied on the class web site. It will scan the text and look up each word in the text in the lexicon. If it does not find the word then it will insert a string in the text that will indicate the error - something like "<ERROR>." When it has finished checking the input text block the server will return it to the client and break the connection. The client and server will each have an Exit option that will close the program.

The server process should also have a GUI interface. It should include a text box which should display incoming client requests so that the TA can see that the processes are communicating. The server should run at a known port above 1024 and less than 32768. You should be able to run two clients and a single server without any problems.

Helpful References:

1. http://tangentsoft.net/wskfaq/ Winsock Programmer's FAQ. (may be an overkill.)
2. http://www.sockets.com/sample.htm
3. http://www.eggheadcafe.com/articles/20020323.asp VB.net Single thread Telnet client & server. This uses the sockets class from .net.

Writeup:

Your write-up should include instructions on how to compile and run your program. Ideally it should be complete enough that the TA can test your program without your being there. Your writeup should include any known bugs and limitations in your programs. If you made any assumptions you should document what you decided and why. This writeup should be in text format and should be submitted along with your code.

Submission Guidelines:

Submit your code via Blackboard. You should zip your source files and other necessary items like project definitions, classes, special controls, DLLs, etc. and your writeup into a name_number.zip. Be sure that you include everything necessary to unzip this file on another machine and compile and run it. This might include forms, modules, classes, config. files, etc.

Make sure your name and your student ID are listed in your writeup, and in comments in your source code. You may resubmit the project at any time. Late submissions will be accepted at a penalty of 10 points per day. This penalty will apply regardless of whether you have other excuses. In other words, it is better to submit this project early or on time. If the TA can not run your program based on the information in your writeup then he will email you to schedule a demo.

If your program is not working by the deadline, send it anyway and review it with the TA for partial credit. Do not take a zero or excessive late penalties just because it isn't working yet. We will make an effort to grade you on the work you have done.

Grading:
Points - element
20 - Client and server processes connect via sockets
20 - Client Process and GUI work correctly
20 - Server Process works correctly
10 - Sever GUI shows incoming client messages
10 - Sever works correctly with messages from multiple clients
05 - TA discretion for overlooked specification problems
05 - Server disconnects after each response
05 - Correct port numbers used
05 - Comments in code

To receive full credit for comments in the code you should have **brief** headers at the start of every module/ subroutine/ function explaining the inputs, outputs and function of the module. To a large extent VB modules are standardized and self explanatory so that more elaborate headers are not necessary. Other languages may have similar characteristics. You should have a comment on every data item explaining what it is about. (Almost) every line of code should have a comment explaining what is going on. A comment such as /* Add 1 to counter */ will not be sufficient. The comment should explain what is being counted.

Extra credit: If you do any of these extra functions then your email should alert the TA to this fact so he will not have to search for things that are not there. If you do not include such a message then the extra credit work will not be counted. Some of these things will require that the TA check your source code. Please make it easy for him to find them.

05 extra points- Make the server multithreaded.

Deductions for failing to follow directions:

-5 Including absolute/ binary/ executable module in submission

-2 Submitting writeup in other than plain ASCII text format
-2 Submitted file has a name other than student's campus Login_Id.zip
-10 Submitting a complete installation of the java virtual machine.

Important Note:

You may discuss the problem definition and tools with other students. You may discuss the lab requirements. You may discuss or share project designs. All coding work must be your own. You may use any book, WWW reference or other people's programs (but not those of other students in the class) as a reference as long as you cite that reference in the comments. If you use parts of other programs or code from web sites or books YOU MUST CITE THOSE REFERENCES. If we detect that portions of your program match portions of any other student's program it will be presumed that you have collaborated unless you both cite some other source for the code. You must not violate UTA, state of Texas or US laws or professional ethics. Any violations, however small, will not be tolerated: you will be referred to the Office of Student Conduct and will receive a grade of -100 on the assignment. Students who do not submit anything get a grade of 0.

DO NOT POST YOUR CODE ON THE INTERNET UNTIL THE END OF THE SEMESTER. IF YOUR CODE IS FOUND ON THE INTERNET PRIOR TO THE END OF SEMESTER, THIS WILL BE TREATED AS COLLUSION, AND YOU WILL BE REFERRED TO THE OFFICE OF STUDENT CONDUCT.