**CS 513 Class Project**

**A Client-Server Chat Program**

**Introduction**

For this project, you must write a chat program based on the client-server model using C or C++ using Linux socket commands or Java and Windows socket commands. The learning objective of this project is provide you with insight into practical computer networks and the problems faced when implementing them, as well as extending your knowledge of computer networks beyond the material covered in class.

# General Information

This is an individual project assignment for each student. Each student must develop and submit a project report. Please see the *Software Programming Project Sample Outline* in the Assignment section on our class website for more detailed information.

All project sources and reports will be checked for plagiarism, including being checked against solutions from online sources. If you are found guilty there will be serious consequences, including receiving a mark of zero for the project. Please review the WPI policies on Academic Honesty and Avoiding Plagiarism at <http://www.wpi.edu/Images/CMS/Bio/academic.pdf>

# Project Specifications

The model used for this project is the single server - multiple client model, and you must use C or C++ and Linux socket commands, or Java and Windows socket commands for this project. The following general specifications must be implemented:

1. Multiple clients
2. No GUI is needed for the server
3. A simple GUI can be implemented for the server
4. Clients must be able to choose a nickname
5. Clients must be able to “whisper” to each other without having messages displayed to other users

## The Server

A single server program should handle all requests from the clients. Your client will have to implement a multi-service solution for your server. The following must be implemented in your server application:

1. Server operations (such as connect requests and disconnect requests) should be printed out by the server.
2. The server must handle connections / disconnections without disruption of other services.
3. Clients must have unique nicknames, duplicates must be resolved before allowing a client to be connected.
4. All clients must be informed of changes in the list of connected users.

## The Client(s)

The following must be implemented in your client application:

1. A list of online users must be displayed (via GUI or command) to all users.
2. Connection / disconnection actions of users must be displayed to all users.
3. Messages from the originating user and other users must be displayed (in other words the messages you send must also be displayed).
4. Must still be able to receive messages / actions while typing a message.
5. Clients must be able to disconnect without disrupting the server.

**Report Preparation and Submission**

This is an individual project assignment for each student. Each student must develop and submit a project report. Please see the *Software Programming Project Sample Outline* in the Assignments section on our class website for more detailed information.

You report should include annotated screen shots which demonstrate that your program has successfully implemented all of the above general project specifications, as well as the specifications for the client(s) and server, and all the functions listed below in the Project Grade Allocation rubric. Your grade for the class project will be based on the demonstrations **(i.e.: annotated screen shots) of successfully implementing all of the functions listed below in the Project Grade Allocation rubric**

The submission deadline given on the course schedule in the syllabus section of our class website, and your project report must be uploaded to our class website **no later than Midnight EDT/EST on that date.**

All project sources and reports will be checked for plagiarism, including being checked against solutions from online sources. If you are found guilty there will be serious consequences, including receiving a mark of zero for the project. Please review the WPI policies on Academic Honesty and Avoiding Plagiarism at <http://www.wpi.edu/Images/CMS/Bio/academic.pdf>

# Helpful Hints

Start early, do not leave this project to the last few days and think you will finish on time.

Look at the project grade allocation and see what needs to be done to help you gauge your progress.

Do not think you are almost done after you have successfully had your client connect to your server.

Handling multiple clients is a big part of this project so make sure you do not wait until the end before you start implementing multiple client handlers.

Disconnecting clients can cause a lot of problems, so please make sure you do not leave the issues for later.

**Project Grade Allocation**

|  |  |
| --- | --- |
| Client GUI | **10** |
| Sending / Receiving Messages | **10** |
| Sending / Receiving Messages (concurrency) | **10** |
| List of currently connected users | **15** |
| Program Stability and Error Reporting | **15** |
| *- Updating user list after user connection / disconnection* | 3 |
| *- Client Stable after Server Termination* | 3 |
| *- Server Stable after Client Termination* | 3 |
| *- Whispering to a Non-Existing Client* | 3 |
| *- Trying to Connect to Non-Existing Server* | 3 |
| All Opened Sockets and Streams Closed on Client after Termination | **10** |
| Server Notification System (notifying clients of activities) | **10** |
| Report | **20** |
| *- Language, spelling and grammar* | 5 |
| *- Selection of tests and experiments discussed* | 5 |
| *- Description of the tests and experiments conducted with copies of test results* | 5 |
| *- Conclusions drawn from the results* | 5 |
| **TOTAL** | **100** |