

CS 428/528

Instructor: Yan Wang

Individual Project 1: Socket Programming

Due: March 20th, 2019 23:59:59pm

In this project, you'll need to implement an IRC-like chat server, focusing first on ensuring familiarity with socket programming. There are two computers, User1 (**client**) and User2 (**server**) that will communicate with each other. User2 will create the socket and wait for an incoming connection from User1. After User1 connects to the User2's socket, then they can send/receive the messages back and forth (e.g., User1 sends "How are you?" to User2, then User2 can receive the message and reply "Good" back to User1 and so on, as shown in the following figure.) The communication should be maintained until any of them closes the connection.



What programming language to choose?

You must work by yourself on this project using C.

Here Are A Few Tips for Programming in C

You will need to learn the use of system calls such as `socket()`, `bind()`, `listen()`, `accept()`, `close()`. Here is a url to get you familiar with socket programming:

<https://www.cs.rutgers.edu/~pxk/rutgers/notes/sockets/index.html>

What requirements this project has?

1. You still need a terminal window to show the received text messages and inputs (No special requirement on the interface).
2. The client and server need to be run on different computers (you can use computers in the lab G7 or Q22 or your own computers).
3. Must use the socket in C language to implement the client and server.

4. The server needs to be able to accept multiple clients connecting to it at the same time.

What to hand in?

1. Submit your code (e.g., yourname_client.c and yourname_server.c) with in-line documentation file to blackboard (*deduct 10% points per day after the deadline*)
2. Submit a Readme file with your code to describe your code
 - Instructions about how to run your client and server
 - Completion status of the project (e.g., what has been implemented and tested, what has not)
 - Anything else you want the TA to be aware of while grading your code
3. Show the demo of your project to TA (*your code must run correctly using two computers in the lab G7 or Q22. When do the demo, TA and you will open two computers in the lab as server and client respectively. It is your responsibility to make sure that your code compiles and runs correctly on those computers.*)

Grading Criteria

1. In-line documentation - 10 points
2. Readme file – 10 points
3. Correctness - 80 points
 - **NO credit** if your project does not compile
 - **NO credit** if you copy codes from any resources other than yours (*your project must be your original work. we will use MOSS2 to detect plagiarism in the projects.*)