Term Project

Create an internet-based online chat system, consisting of one server and *m* clients, where *m* can be a reasonably predefined number.

After the server has been started, up to *m* clients can communicate with each other via the server at the same time. Each client is identified at the user level by a user selected ID, which can be of any combination of letters and numbers. Messages sent from one client will be displayed on all other clients. Each message is prefixed by a symbol that consists of its user's chosen ID, followed by a colon (**:**). It is the server's responsibility to make sure that each message be sent in its entirety before sending another message. Let's define a message be a sequence of characters terminated by a carriage return.

When a client is invoked, it will prompt the user to select a user ID. This ID is used by other users to identify from whom a message is generated. A client can join or leave a chat session at anytime. A user announces his/her intent to join a chat session by sending the chosen ID to the chat server. The server, in turn, will broadcast the new comer's ID to all current users. A <ctrl-d> by itself on a line indicates a user's intent to terminate a chat session. A message showing a client has left must be displayed on all other clients' screens. When there is only one user online, he/she must be told by the server to wait until at least one more user has joined the chat session.

The server can be terminated by the system administrator via the SIGUSR1 signal. Before the server terminates, it must send a message to all clients indicating its upcoming shutdown.

C is the preferred language for implementing this project. However, you may choose other general-purpose programming language, such as C++, Java, etc., if you feel more comfortable with that language. Also, you may use either threads or processes to implement your current server.

Name your files based on the functionality of the programs: i.e., **projSer.*x***, **projCli.*x***, etc., where ***.x*** stands for the appropriate filename extension of the programming language you use.

The due date of this term project is November 28, 2010, before 11:30 pm for all students in this course.

All term projects will be run-tested. The run-tests will start on November 29, 2010. You need to be present for your run-test. We will discuss more about your run-test schedule in class as we go along. All run tests will be conducted on the CSE student server: **lamp.cse.fau.edu**.

A final project report is required for your project. It is free format. However, it should be composed of two parts: general introduction and technical details. The first part is for general public to read. You should focus on what you system is about, and how to use it.

The second part is for programmers to read. You need to put emphasis on the design logic of your project. Don’t include your source listing in your term project report. Make them separate files and submit them via the **HWSAM** program before or on the due date.

The term project will be graded based on: implementation, documentation, and presentation (during run-test). It is worth at least 20% of your final grade.

All term projects are individual projects. You are the **sole creator** of your project. Due credit must be given to the author whose idea you have used in your project. Permission must be sought from the original author whenever applicable. Verbatim copy of other’s work with/without modifications is in no case permissible.

Don’t procrastinate! An early start will help you do a timely and quality project.