

# CIS527 Programming Assignment 2 (Extra Credit)

## Internet Multiperson Chat Service

Professor Jinhua Guo

Fall 2003

*Due Tuesday, November 11, 2003*

### 1. Introduction

In this assignment, you will use the socket interface to implement a simple, Internet chat room services. Your implementation will consist of both a client and a server application. The client can allow users to login and logout, check who is active, send messages to other active users, receive messages from other active users, and quit the chat room. The server verifies the identity of the users, forwards the message to the destination, and print out all clients activities on the screen.

### 2. The Assignment 2

For this assignment, the server must allow multiple clients connect the server at the same time. You may use Select Server (Synchronous I/O Multiplexing) or multithread Server. You will need to implement the following command on the client side and the corresponding function on the server side.

#### 1. login [UserID] [Password]

Identify yourself to the remote talk server. If the UserID is not specified, the client will prompt the user for it. If the Password is not specified, the client will prompt the user for it. If the UserID and the Password match, the server will send a confirmation message to the client; otherwise, the system will prompt the user to try again. For a successful login, the server should inform all other active users that a new user joins the chat room.

#### 2. logout

Logout from the remote server. The server should inform all other active users that the user exits from the chat room.

#### 3. who

List all active users, including the **UserID** and the users **IP addresses**.

#### 4. send UserID

Send a message to an active user. If the UserID is invalid or the receiver is not active, the server replies the client with an error message. If the UserID is “*all*”, the server broadcasts the message to all other active users, otherwise it forwards the message to the designated user. And, the receiving client should process the message immediately.

#### 5. quit

Terminates the talk session with the remote server. The client exits. If the user has not logout yet, it should automatically logout the user and inform all other active users.

### 3. Programming Environment

You can use either C/C++ or Java to implement the assignments. The assignments will be tested on the Sun workstations in 1180EC. For easy grading, please don't use any GUI interface.

### 4. Requirements

The following items are required for full-credit:

- must allow at least 5 clients connect to the server at the same time.
- implement all five commands: login, logout, who, send, and quit
- the users information should be maintained by the server. You must have the following users in your system:

UserID	Password
john	john
david	david
jennifer	jennifer
mary	mary

- a user can only be able to send the message to the server after its identity has been verified by the server.
- a user cannot have two or more active sessions.
- both the server and client should be able to run on any Sun workstations.
- the client should be able to connect to the server running on any machine. Therefore, the server IP address should be a command line parameter.
- the server should output all client activities on the screen.
- your source codes must be commented.
- include a README file in your submission.
- include a Makefile in your submission.

## 5. Submission Instruction

(1) Copy all your files (only source codes, data file, Readme and Makefile, no object file and executable file) in a directory. The directory should be named like lastname\_firstnameinitial\_p2. For example, if your name is John Smith, your directory name should **smith\_j\_p2**.

(2) Generate a tar file of the directory using the following command.

Enter the parent directory of your current directory and type

```
% tar cvf lastname_firstnameinitial_p2.tar lastname_firstnameinitial_p2
```

For example

```
%tar cvf smith_j_p2.tar smith_j_p2
```

(3) Email the tar file before 11:59PM, November 11th, 2003 to me [jinhua@engin.umd.umich.edu](mailto:jinhua@engin.umd.umich.edu)