

## **Assignment 6**

### ***Implementing a chat server***

**Due: November 1, 2009, 11-30 pm**

In this assignment, we will implement a simple chat server. Chat clients will connect to a central server  $S$  which will coordinate the chat.

The chat server  $S$  is a concurrent server handling multiple clients at the same time. It has one thread  $T_S$  which will wait on a TCP port  $X$ . A chat client  $C$  connects to  $T_S$  on port  $X$ , which then creates a separate thread  $T_C$  to handle that client. All communication between  $C$  and  $S$  will occur through  $T_C$ .

A client  $C$ , on a successful connection, first sends its name (a single word string) to  $S$ .  $S$  will announce the entry of  $C$  to all other connected clients by sending a message to all connected clients. After this, any message sent by the client is sent to all other connected clients by  $S$ . Similarly, any message sent by any other client is received by  $C$ . In essence, any message sent by any connected client is received by all other connected clients. Any message shown to a client will have the sender's name and the sending time before it. The sending time will be added by the server  $S$  before sending the message to all clients.

There is no security and any client can join the chat room. However, to avoid too much clutter, the server takes the following steps:

1. It blocks any message that is more than 200 characters long. If a client sends such a message to the server, the server sends an error message "Message too long" to the client and the message is not forwarded to any other client. The client will then show this message to the user and the user will need to type a shorter message.
2. The server allows at most 5 clients to connect at any one time. If all 5 clients are connected, any other client trying to chat will get a message from the server "Too many users, please try after some time". The server will send the message and close the connection with the client.
3. No client is allowed to be online for more than 10 minutes, to give others a chance to connect. The server should send a message "Online for too long" and close the connection of any client who is online for more than 10 minutes.

Use port  $X = 50000$ . Submit only the two C files for server  $S$  and client  $C$ .