ECEC 353: Systems Programming
Programming Project

Prof. Naga Kandasamy, ECE Department, Drexel University

February 24, 2014

The project is due March 23, 2014. You may work on the project in teams of up to two people.

Project Goal

Develop a Chat application consisting of a chat client and a chat server. The client and server must be written in C/C++, and if the applications are multi-threaded, you must use the pthread library. Also, since the Chat application is entirely housed on single machine, no network programming is required. The chat application allows multiple chat clients to connect to a chat server. Once connected, clients can exchange messages with users that are connected to the chat server. The server has to accept and maintain connections to all the clients and relay chat messages between them.

Specifications for the Chat Client

When the client program is executed, it attempts to connect to the server process. If the server is not running, the client application prints an error message to the user and exits. Once the client is connected to the server, it must: (1) transmit messages typed into the console to the server and (2) display messages that are sent to it by the server.

Specifications for the Chat Server

The server must allow up to a maximum of ten users to enter and leave the chat system at any time. When a client message is received by the server, the server must broadcast that message to the rest of the clients that are connected to it. Also, it should be clear to each client exactly who typed what. In other words, the broadcast message must also contain the identity of the originating client.

Enhance the chat server to also allow private one-on-one communication between any two clients at any time. For example, consider two clients A and B. The public messages sent by these clients are broadcast to the rest of the group. However, if client A (B) wants to exchange one-on-one private messages with client B (A), then your chat server must allow this.

The client and server processes are unrelated. You must use shared memory for inter-process communication as well as any appropriate synchronization mechanisms to control access to the shared memory.

Project Deliverables

E-mail your source code to the teaching assistant with complete instructions (in a separate README file) on how to compile and run your program on the `xunil` cluster. In addition, please submit the following documents to me as a hard copy.

- A brief description (up to five pages) of the software architecture of the client and server processes as well as the key data structures used in these processes. The discussion should also include the design challenges faced, and how you addressed them.

- A brief description (limit to one page) of how to test your code. Here, you must clearly describe how to set up a test environment in which multiple clients can connect to the server and exchange messages between one another.