**CIS 620**

**Spring 2014**

**Assignment #2**

**Due date: April 2, 2014 (6pm)**

**Email Server**

In this assignment you will use Internet-domain stream sockets to implement a simple email server system. The email server should implement the following functions:

1. **Register** a system. The name of a machine (e.g. ‘camelot’) which will be using the email server is registered with the server. In order to use the server, the machine must first register with the server. Attempts to send/receive email by unregistered clients will be rejected.
2. **Deregister** a system. A client (e.g. ‘camelot’) deregisters when it will no longer be using the server. The server removes the client from its list of registered clients and deletes the buffer of messages for that client (if any exist).
3. **Send** a message. A client sends a message to a user at a registered machine. If the destination or source machine is not registered, the server sends an error code to the client. Otherwise, the server appends the message to its queue of messages for the destination machine. The buffer space for each registered machine should be limited to 1024 bytes.
4. **Receive** all messages for the client. The server sends all of the queued messages to the client, who displays them, nicely formatted.

Email messages contain several fields: **sender** (e.g. ‘joe@merlin’ – the server must verify that the message sender field matches the actual sender, otherwise an error message is generated); **destination** (e.g. ‘jack@camelot’ – the server verifies that ‘camelot’ is registered); **subject** (up to 80 characters); **message body** (variable length, max. 512 characters).

You should write a client program which has a menu allowing the user to register the system, deregister the system, compose/send a message, receive all messages for the machine the client is running on, or exit (5 menu choices). The compose/send command allows the user to fill in each of the message fields, verifies the format of each field, and then sends the message. A message indicating success or failure of the command is printed. The receive message prints out each of the received messages, nicely formatted, or an error message if there are no messages on the server or something else has gone wrong. All communication between clients and server is via Internet-domain stream sockets.

Note that the server does not check for validity of individual users, only validity of the source/destination machines (whether they are registered). Any number of clients should be able to run simultaneously and your server runs forever (infinite loop). The name of the machine the server is running on is passed in on the command line to the client program. The client program keeps running (displaying the menu) until the user enters an ‘exit’ command.

Your program must be submitted as described in the syllabus. Make sure that you document the procedure to be used to test your program (i.e. the sequence of commands to be entered). Any deviation from the description given here or from the description of the submission process given in the syllabus will cause you to lose points. Programs will be examined for copying.