CS349/449

Recitation # 1
September 23, 2004
Project 1

• Instant Messaging Client and Server
  • Similar to AIM, IRC

• Purpose
  • to become familiar with client/server socket programming
Server

• create a server socket
• listen for connection requests
• on accept( ), spawn a new thread with connection socket
• now each thread is responsible for handling the communication between server and the client that initiated the connection
Client

- Connect to server IP address on port \(<x>\)
- Login with username and password
- Send messages to server (and server passes messages to other clients)
- Publishes messages received from server
Client Commands

• The following are entered by the user in the chat client
  • /login <username> <password>
  • /who
  • /logout
  • /knock <username>
  • /msg <message>
Client Commands

- /accept
- /deny
- /endprivate
Protocol Commands

- Take the general form
  - COMMAND\n  - TAGNAME: value\n  - TAGNAME: value\n
- Commands are case sensitive (all caps)!
Protocol Commands

- CONNECT
- OK
- ERROR
- SENDALL
- PUBLISH
- WHO
Protocol Commands

- ULIST
- BYE
- TALKTO
- TALKREQ
- TALKRESP
- SEND
Protocol Command Tags

- UID
- PASSWD
- SEQ
- ERRMSG
- MSG
- PORT
- IP
User Database

- Simple text file kept on server machine
- Each line of the form “username:password”
- Passwords are stored in plain text and sent across the network “in the clear”
- Don’t use passwords you use in real life!
- A note about threads and shared variables
Figure 1: Example of Client Server communications
Figure 2: Example of Peer communications
Details

• Assignment specifications will be available online tomorrow (September 24th) afternoon

• Project due date is October 15th
  • This is three weeks from tomorrow

• A 1-2 page writeup will be due along with code and makefiles
  • plaintext documents only!
More Details

• A test server will be provided to test your clients and compare the functionality of your servers (soon, but TBA)

• Java vs. C/C++

• NO late submissions (sorry guys)