**To Do:**

* **Design doc (page 3)**
  + **Submit to canvas**
* **Push correct files to github**
* **Provide github name to TA for grading**
* **Add error messages to socket system calls. Like if socket fails to connect. Display an error message like socket failed to connect. That sort of thing**

**crsd**:

Chat Room Server

Infiinite loop with steps in any order:

* Listen for commands from client
  + CREATE
    - Check if chat room exists
      * If no,
        + create new master socket with post number diff from “usedr”
        + create new entry for local chat room in local db

store name and port number of chat room

* + - * + return command result to client
      * If yes,
        + Nothing
  + JOIN
    - Check if room exists
      * no
        + nothing
      * yes
        + return port number of master socket for room
        + return current number of room members
        + connect client to chat room
  + DELETE
    - Check if room exists
      * No
        + Nothing
      * Yes
        + Send a warning message to connected clients before terminating their connections
        + Close master socket
        + Delete entry
        + Inform client with results

Incoming messages:

* Handled on slave socket derived from master socket of a chat room
* When a message comes in, forward message to all folks in that room

Client exit:

* Handled by terminating connection to server
* Server must handle this

Invoke crsd:

$ crsd <port\_number> &

**Crc:**

chat room client

Invoked with:

$ crc <host\_name\_or\_ip\_address> <port\_number>

How it works:

* CREATE
  + Connect to port of chat server
  + Send creation message
    - $ CREATE <name>
  + Display reply and close connection
* DELETE
  + Connect to port of chat server
  + Send delete message
    - $ DELETE <name>
  + Display reply and close connection
* JOIN
  + Connect to port of chat server
  + Send join message
    - $ JOIN <name>
  + Display info
  + On success, connect to port returned by number and begin message exchange
* Leaving a room
  + Terminate the connection on the client side
  + Server will handle this accordingly

**Complication**

* You need to read command line input and connection data at the same time
  + 2 ways to handle
    - Separate threads to handle each input source
    - Use select() to handle the sources on one thread