**OS Lab: Threaded Server 2**

1. I tried to make the protocol human-friendly. I regret it. Please reduce it to the following protocol.

Assume the clients are well-behaved: names are unique, responses are appropriate and well-formed. The only problem that might occur is clients may disconnect unexpectedly. Each line must end with a CR (\r) and LF (\n). The combination helps to simply handle input from human users on various OSes.

* When the number of clients increases to 1:

|  |  |
| --- | --- |
| **Server sends:** | **Client answers:** |
| QS|ADMIN | GROUP|*Lname|Lnum* |
| WAIT |  |

* While the number of clients is less than Lnum:

|  |  |
| --- | --- |
| **Server sends:** | **Client answers:** |
| QS|JOIN | JOIN|*Lname* |
| WAIT |  |

* While the number of clients is equal to Lnum:

|  |  |
| --- | --- |
| **Server sends:** | **Client answers:** |
| QS|FULL | Nothing because server closes |

For the lab submission, on Friday at noon. Your server should do the following parts:

* Implement the greeting part of the protocol shown above.
* Handle the issue of waiting for a group of a specific size.
* Handle sending any single question in any format and determining the first client to answer without error handling

Some things you may need:

* strtok\_r: strtok is not thread-safe
* pthread condition variables
* write a threaded server to help you test simultaneous arrival of clients