**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 |

1. Have the server require a new argument, a filename. The filename must be first, then the optional port number is second. This file will store questions and answers for a quiz. It may be read using normal C library functions, such as fgets. You may assume the file, if it exists and is readable, is well-formatted. The format is simple and doesn’t really require complex parsing. Each question has an id and is on its own line, followed by the possible answers, one per line, also identified somehow. The last answer line is followed by a blank line. Then the id of the correct answer appears on its own line, followed a blank line. The total number of characters in a question with its answers, fully formatted, with not exceed 2048 bytes.

Quiz file format:

***id question*\n**

***id* *answer*\n**

***id answer*\n**

***id answer*\n\n**

***id*\n\n**

Example:

**1 What is a system call?**

**A. A call to the system**

**B. A function call that invokes the operating system to take some action**

**C. A call to the operating system**

**B**

**2 Another question**

*(and so on)*

1. As long as at least one client remains in the group, keep asking questions, until questions are exhausted. When they are all finished, announce the winner (and standings of all players), and disconnect from the clients.

Protocol for sending questions and announcing a winner.

|  |  |
| --- | --- |
| **Server sends:** | **Client answers:** |
| QUES|size|full-question-text | ANS|answer |
| WIN|name | ANSOK |

1. More to come on the final standing protocol details …