**Introduction**

The purpose of this document is to outline the behavior and protocol of the “QuizServer” C program. The QuizServer application is able to provide predefined synchronized quiz to groups populated by clients joining into a local server. The Quiz information is stored in a .txt file with a specified format (See Appendix).

**The protocol**

**Joining the Server**

Whenever a new client accesses the server, the server sends a message about the groups currently awaiting members: OPENGROUPS, followed by the list of groups, each consisting of the quiz topic the group name, the desired group size, and the current size. This message is terminated by a CRLF.  At any time a client is not involved in the group activity (either not in a group or in a group that has not started yet), the client may request the current list of open groups, by sending the message GETOPENGROUPS.  ENDGROUP is used when a group has ended, to inform the group members that their activity is over.

|  |  |  |
| --- | --- | --- |
| **Server sends:** | **Client answers:** | **Notes** |
| OPENGROUPS|topic|name|size|curr … | nothing | The group information repeats for all open groups |
| WAIT | nothing | Optional command – client may ignore |
| ENDGROUP|groupname | nothing | Group is ended naturally or unnaturally, but waiting clients are not dropped |

**Creating a Group**

The client may start a new group, or join an existing group: either one or the other, but not both.  To create a group, the client sends the GROUP command, which contains the quiz topic, the group name, and the user’s name.  The server will respond with SENDQUIZ, which invites the client to send the quiz file as a stream of questions in exactly the same format as in the previous homework.  The client QUIZ message is the type of message where the size precedes the data, and the CRLF terminator is not used.This is so the server can read the data in a loop. The server may answer BAD if the group name is already in use (or any other error occurs).

BAD messages are always followed by an error description, and terminated by CRLF.  For example:

BAD|The group name is not uniqueCRLF

**Joining a Group**

The client may join an existing group with the JOIN message by supplying the group name and the user name.  The server responds OK or BAD (with some message, as in the example above).

The client may LEAVE a group before or during the group activity.  After leaving their current group cleanly, the client may join another group, or create one.

A group creator may cancel a group, only if the group activity has not started yet.  If the activity has started, it will continue, and even if the creator exits the program it will still continue, but the creator cannot cancel it.  The LEAVE command is not allowed for the group creator.

There are a maximum of 32 groups allowed at one time.

|  |  |  |
| --- | --- | --- |
| **Client sends:** | **Server answers** | **Notes** |
| GROUP|topic|groupname|groupsize | SENDQUIZ or BAD | After server responds with SENDQUIZ, the client can send the quiz message |
| QUIZ|quizsize|quiztext | OK or BAD | This message is not terminated with CRLF – the size gives the needed information |
| CANCEL|groupname | OK or BAD | Server closes the group |
| JOIN|groupname|username | OK or BAD |  |
| LEAVE | OK or BAD |  |
| GETOPENGROUPS | OPENGROUPS | Server responds with the entire list of open groups |

**The Quiz**

For each question, the first person to answer the question correctly is recorded as the winner, and gets 2 points.  Other users who get the correct answer get 1 point, no answer gets 0 points, and the wrong answer yields -1 points.

The server assumes that clients are well-behaved and answer either a real answer, NOANS, LEAVE or close their socket in a reasonable amount of time (no handling of badly-behaved clients that remain connected but do not send messages during an activity is required).

The group creator is sent each question, but is not allowed to answer.  The creator is also sent the message about each question’s winner, and about the results.

The server sends the size of the entire question plus answers string, and then the string.  A terminating CRLF is not used or required because the size precedes the message, and the message contains newlines within it.  If the client does not know the answer it sends the special text “NOANS” in place of the answer ID.

**Finishing the Quiz**

After all the users have answered a quiz question, the server sends all the client the winner’s name (the name of the client who scored 2 points for the fastest right answer).  If no one answered correctly, the winner’s name that is sent will be the empty string “”.

|  |  |
| --- | --- |
| **Server sends:** | **Client answers:** |
| QUES|size|full-question-text | ANS|answerID |

When a group activity is completed, the clients return to being able to create a group or join a group. As long as at least one client remains in the group, the server will continue with the group activity, until the activity ends normally.  When the activity is finished, the server should announce the winner (and the standings of all players).

After all the questions are finished, the server sends the quiz standings as a series of names and scores in descending order by score, terminated by CRLF.

|  |  |
| --- | --- |
| **Server sends:** | **Client answers:** |
| RESULT|name|score|name|score … | Nothing |
| RESULT|name|history|name|history … | Nothing |

Appendix

Quiz file format and example

**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)