

CS 5200
Assignment 6
Due November 5, 2008
25 Points

The Letter Game

Overview

This is the fourth part in our multi-part programming assignment associated with the *letter game*, which is an educational way to pass the time during long car trips. In this assignment, you'll enhance the performance and scalability of the game so that it supports multiple players efficiently using threads.

Program Description

In the fourth part of the assignment, you'll enhance your letter game program to use threads instead of fork. Each client has its own session, in which it reads commands from standard input and displays the results from the server on its standard output. The commands are sent to the server process, which uses threads to run commands and return results to each game session.

The letter game server supports the following commands:

- **Session** – Allows a player to create or join a session, which is used to keep track of the scores for all players in a particular game. This operation is passed a *session name*, that is a string that identifies the game. If the session isn't already opened, then allocate the resources (e.g., dictionary, score table, etc.) associated with this particular instance of the game. Each player is identified by her unique host/port address. This command returns a *session handle*, which is an integer that uniquely identifies the game session and must be passed to all other commands to identify the session.
- **Dictionary** – Allow a player to designate a dictionary of valid words for use with a particular session.
- **Set** – Set the current letter set for a session to the desired three letter string of letters from a license plate.
- **Guess** – Allow a player in a particular session to query whether the designated word contains the current letter set *in order* in the word and also exists in the dictionary. Whenever a query “matches” a word in the dictionary, the player receives 1 point.
- **List** – List all the words in the session's dictionary that match the current letter set.
- **Score** – Return the current results for all players in a particular game session.
- **Quit** – Stop playing the game for this session and return the results. Only when all players in a game session have called **Quit** should the resources for that game be reclaimed.
- **Shutdown** – Shutdown the server process ultimately leading to exiting the process. Only when all players and sessions have shutdown, however, should the server process exit. Once **Shutdown** has been called, no further connections should be allowed.

The **Set**, **Guess**, **List**, **Score**, and **Quit** commands can be invoked multiple times by players of the same game, but can only run when no other commands are executing concurrently on a game. Below, we discuss the general behavior of the client and the server programs.