

CS 5200
Assignment 7
Due November 21, 2008
25 Points

The Letter Game

Overview

This is the fifth part in our multi-part programming assignment associated with the *license plate games*, which is an educational way to pass the time during long car trips. In this assignment, you'll develop a more sophisticated game that will reuse portions of your other solutions. In addition, you must use Linux RPCs (Remote Procedure Calls) to implementing the client-server application.

Program Description

The initial assignment involved writing a simple C++ program that automated the letter game by supporting the following commands:

- **Dictionary** – Allow players to designate a dictionary of valid words.
- **Set** – Set the current letter set to the desired three letter string of letters from a license plate.
- **Guess** – Allow players to query whether a particular word contains the current letter set in order in the word and also exists in the dictionary
- **List** – List all the words in the dictionary that match the current letter set.
- **Score** – Show the current score for the player.
- **Quit** – Stop playing the game and exit the program.

In this part of the assignment, you enhance your letter game program to use a client/server architecture. As before, the client reads commands from standard input and displays the results on its standard output. Instead of running the commands in the client process, however, the commands are sent to the server process where the commands are run and their results returned to the client for display.

Below, the general behavior of the client and the server processes are discussed.

Client Application

The letter game client program should perform the following activities:

1. Read commands from the user's standard input and convert these into the appropriate type of encoded messages.
2. Send the encoded messages to the server and wait for the reply.
3. Read the reply from the server and display it to the standard output of the client.

Server Application

The letter game server program should perform the following activities:

1. The servers runs as a function and is accessed as an RPC.
2. When a client calls the server function, the server performs the appropriate operation designated by the type of command received from the client.
3. Assuming the operation succeeds, the function returns the reply to the client.

Requirements

- You must submit a makefile with you code so make it easy for the grader to compile the program. If you don't know about makefiles, then it is time to learn.
- You must submit a readme file that tells the grader how to run your program.
- In order for you to do RPC, the portmap daemon must be running.