CSCI5134 Concurrent Programming and Software Modeling

Programming Assignment No. 2

DUE: Tuesday October 20th, 2015

Objective: The objective of this assignment is to give you some experience on using the Sockets interface to TCP/IP to develop a simple multiservice server and a client application and to allow you to work with multiprocessing application development.

The multiservice service should provide two services:

You have the option of implementing your multiservice using two threads, one for each service or as multiple processes, one for each server (i.e., fork a process initially and parent process provides one service and child process provides the other).

Service 1: Fortune Cookie Service. This service should be provided as an iterative server using predefined UDP port number to provide fortune cookie(s) to clients who request it/them. You server should, after receiving a request form a client, handle that request, identifying the number of cookies requested by the client, look up the appropriate number of fortune cookies from either an array, a file or a small database, and return them to the client. Each fortune cookie should be drawn at random from a predefined list of at least 50 fortune cookies. Each client should receive different cookie(s). It is not allowed for multiple clients to receive the same cookie the first time they requests one, it should be truly a random cookie from the set. (A fortune cookie is a phrase like the ones you get in the fortune cookie at the end of your meal in Chinese restaurants).

Service 2: ENCRYPT/DECRYPT Service. This service should be provided as a concurrent server using multiprocessing, the server should receive connection requests from client(s) and hands them off to a new process. The original process should go back to accept connections from other clients. The new process should interact with the client until the client decides to terminate the connection. Each request from the client should be a) A request to encrypt a line of text b) a request to decrypt a previously encrypted line of text c) a terminate command that indicates the client is about to close the connection and won’t be sending any more requests.

For a) the server should encrypt the line of text using which ever encryption algorithm you choose, it could be as simple as you want to make it, as long as the returned line looks like garbage to the naked eye.

For b) the server should decrypt the line of text by running the reverse algorithm and return the original line that was previously encrypted.

For c) the thread/process must close the connection and exit.

Client: The client should:

Interact with the user (you) and asks what the user wants to do? a) interact with the Fortune Cookie Server b) interact with the encryption/decryption server or c) quit.

For a) the client should ask the user how many Fortune Cookies he/she wants, send the request to the appropriate server, receive the reply and display the fortune cookies on the screen. Then go back to ask again what the user wants to do.

For b) the client should ask the user if he/she wants to encrypt, decrypt or stop, then ask for a line of text to be encrypted or decrypted depending on the previous answer and send the appropriate request to the server, receive the replay and display it in the screen. Repeat until the user chooses to stop, in which case it should go back to ask the user which server to interact with or quit. i.e. a client should be able to send multiple lines to the server to be encrypted/decrypted until the user decides to stop and go back to original menu.

Application must be developed in a unix platform.

Upload to blackboard a zip file that includes the source code of **well documented programs** (both server and client) and schedule a demo with the TA to show your program working the week after the assignment is due. The demo should include the servers running in one machine and multiple clients running in several other machines.