Assignment #1

Date Due: October 21, 2019

Total: 100 marks

Inter Process Communication

Write there suites of C programs to compute the next course you should take.

If the mark in the current course is more than 70% you cant take courses in list1.txt otherwise courses in list2.txt. After the list is presented to the user, the client program sends data to the server, which will confirm registration, send it to the client, which will print the information to the standard output.

The program should be written as a client server setup, where the server process is doing all the computations and client processes are interacting with the users.

Requirements:

- The program accepting the input and displaying the output, i.e., the interface program should not do any other computation then communicating with a server program. Your client program should accept input from the programmer and send it to the server program. It receives the result from the server and prints it.
- The server program will perform all the computations and send the results back to the interface program. Your server program will accept connections and compute final mark. If the mark is greater than 70% it reads list1.txt and sent this information to the client, otherwise reads list2.txt and sent this information to the client. The client will then select one course and sent it to the server. The server will create the file NEXT-course.txt where it stores the name of the selected course. It will send this information back to the client.
- The type of communication between the programs will constitute a complete suite.
- A default marking scheme for computing your mark is stored on the server.

  You have the freedom to chose the syntax of your files, but it must be in a human readable format. The exact syntax for each file should be well documented and included in a Readme file. Compiling should be done using an appropriate make files.

  Each suite is marked as follows:

  1. client program 5 marks
  2. server program 5 marks
  3. communication protocol (explained and properly implemented) 10 marks

- Communication protocol is one of the following:
  1. named pipes (fixed names one for client to server the other one for server to client)
  2. unnamed pipes
  3. named sockets (one for each connection)
  4. internet sockets on localhost.
  5. named shared memory
  6. private shared memory

For each program(20 marks): the Readme file, Makefile and proper documentation will worth 5 marks (if not present the mark cannot be more than 15). Testing programs worth another 2 marks(for each suite). These are additional marks on top of the 20. Showing an execution of your programs should be done to get these points (text files only, no png/gif/jpeg/pdf pictures).

All programs should work on our Virtual machines(provided).