**Operating Systems Project 3**

**University at Albany**

**Department of Computer Science**

**ICSI 500**

**Assigned: Wednesday, September 30th, 2020.**

**Due: Monday, October 12th, 2020 by 11:59 PM. Submissions with 20% penalty will be**

**accepted until Wednesday, October 14th, 2020 by 11:59 PM.**

**Student Name:**

## PURPOSE

Develop a practical understanding of task collaboration using socket programming.

## OBJECTIVES

## Develop a client/server application using Linux TCP sockets and the C programming language. Your solution will respond to service requests by clients. Such requests may be by either providing the IP address of the server or the name of the host where the server is executing.

## PROBLEM

You are to modify both the client and the server programs you have developed for project 2. You are also to use the solution you have developed for project 1 Milestone 3. In essence, your solution for project 3 will reuse previous code you have developed for both project 2 and project 1 Milestone 3.

You are to develop a data processing system, where the server will respond to different client requests by creating dedicated new processes. You are to use fork(), wait(), as well as the exec() family of Linux process system calls. Each new process created by the server will perform only one of the services defined in the project 2 document. All communications between both the client and the server will be encoded according to the format defined in the project 1 Milestone 3 document.

The services provided by the server will be almost the same ones defined in the project 2 document. However, the syntax used by the client will be such that both services are requested by means of a single command. The following syntax illustrates this requirement:

*service1, service2< char, file.txt>*

In the above example *file.txt* is a text file, *char* is a character, *service1* and *service2* are either *toUpper* or *count* as defined in the project 2 document. Note that the order the services are shared with the server is not relevant. Requests that have *service1* followed by *service2* should provide the same result as requests that have *service2* followed by *service1*. Also your server solution is to combine the data produced by each individual service into a single document, by merging them to a text file named *fileChar.txt* and share the *fileChar.txt* file with your client application.

**INPUT TEST FILE (*intext.txt*)**

You are to name your input file as *intext.txt* and populate it with the following contents:

*source code represents the part of process that contains the programming*

*language itself. you may use a text editor to write your source code file. a compiler will be used*

*to produce a machine representation of your source code. Such*

*representation my show your code as hexadecimal or*

*binary formats. the resulting hexadecimal code will contain combinations of numbers such 1 3 5 8 or*

*combinations of characters and numbers such as 1A3B4C0DEFA, for example.*

## DOCUMENTATION

Your program should be developed using GNU versions of the C/C++ compiler. It should be layered, modularized and well commented. The following is a tentative marking scheme and what is expected to be submitted for this assignment:

1. External Documentation (as many pages necessary to fulfill the requirements listed below.)

including the following:

* 1. Title page
  2. A table of contents
  3. [20%] System documentation
     1. A high-level data flow diagram for the system
     2. A list of routines and their brief descriptions
     3. Implementation details
  4. [5%] Test documentation
     1. How you tested your program
     2. Testing outputs
  5. [5%] User documentation
     1. Where is your source
     2. How to run your program
     3. Describe parameter (if any)

1. Source Code
   1. [65%] Correctness
   2. [5%] Programming style
      1. Layering
      2. Readability
      3. Comments
      4. Efficiency

## WHAT TO SUBMIT

The following are to be submitted through **Blackboard**:

1. Your documentation for project 3,
2. The source code of both your client.c and server.c well as
3. The file, *fileChar.txt,* according the specifications included in this exercise.
4. You are to place all files that are related to your solution in a .zip file. Your .zip file must follow the format: *500 Project3 Your Name*. Marks will be deducted if you do not follow this requirement.