Timothy Kuo
tyk3
EECS 325
April 28, 2017

Project 4 Report

I. Application Description

My application will create a client/server architecture will also for string manipulation. The client will send over a string to the server, and once received, the server will ask for an operation to perform on the string and then return the final string. The server can handle multiple clients. This application will use TCP because it is important to have the correct values being sent back and forth between the client and server.

II. Usage Instructions

a. Client:
The client must connect to the same port that the server has established a connection as well as know the hostname of the server. To connect, type in "./proj4 hostname port", where hostname is the hostname of the server and port is the port that the server is connected to. Once connected, the server will ask you the string you would like to manipulate and when received, will ask for an operation. The only operations available are as follows separated by commas: -r, -lc, -uc, -m[any char x], -n[any char x]. The description of these operations are described in section III. If you would like to exit the connection, simply type exit and the program will exit.

b. Server:
The server must create a connection by typing in first the program name and then the port number it would like to open, as follows: "./proj4d port". Once created, it doesn't require any manual commands to be typed in as the program will do all the operations for you. Multiple connections to the server are allowed, as the program uses the fork() method to handle multiple clients. It will continue to accept incoming connections from client and can be excited by typing in "Ctrl + c" in the terminal.

III. Application Protocol

This section will describe the usage instructions of each command the client can give and how the server should respond. Once a connection has been established with the server, the client will be able to run a series of commands, that can be called through the following syntaxes and can be run as follows –r, -lc, -uc, -m[any char x], –n[any char x] in the command line:

- -r will reverse the given string
- -lc will change all the letters in the string into lowercase letters
- -uc will change all the letters in the string into uppercase letters

- -m[any char x] will remove all instances of char “x” in the string. For example, if the string is “string” and the char is “s”, the server will return the string “tring”. The space character is also allowed, by typing in a space as the char.
- -n[any char x] will count all the instances of char “x” in the string. For example, if the string is “string”, and the char is “s”, the server will return 1. The space character is also allowed, by typing in a space as the char.

IV. Justification for using TCP

I will be using TCP as the transportation protocol because it is important to have every bit of information received and communicated accurately since this program involves manipulating strings. Otherwise, if we were using UDP, it could cause some parts of the string to be lost and that would make the string manipulations inaccurate, thus destroying the whole point of the program.