

**David Alummoottil**  
**Anirudh Balachandran**

## **Assignment 3 – Remote Files**

### **Program:**

This was one of the more challenging assignments during this course. We were asked to create a server and library that would communicate with a remote client and access and modify files on the computer. We were asked to receive a connection from the client to our library c file which contained the open, read, write and close functions which would then send the flags and the name of the path to our server. Our server which is connected to the library file would then perform the necessary function and return the appropriate value back to the library which would in turn return this value to the client. If an error is encountered -1 is returned and the errno value is also set accordingly.

### **Algorithm- Base Program:**

Upon receiving the path name from the client, the library file sends the pathname and other required variables to the server file. All the files must be opened before other functions can be performed on it and if the file is not open for a particular client, an error message is outputted and the program comes to an end. The open function in the server file receives the buffer along with the appropriate flag as the two parameters. This function then takes these parameters and performs the open function. The file descriptor is returned if it is a success otherwise it returns -1. The appropriate values are sent back to the library in the form of an int array. The errno value is then set in the libnet file and an appropriate error message gets printed out if necessary. This libnet file returns either the file descriptor or -1 back to the client file. The netread function receives the file descriptor, the buffer and the number of bytes that are actually supposed to be read. If it is a success, it returns the number of bytes that are actually read otherwise it returns -1 and errno value gets set accordingly. The server then returns an int array containing the number of bytes read or -1 along with the errno value. The libnet file then sets the errno value accordingly and prints an error message if required or returns the number of bytes read back to the client. The net write function works in similar way. It receives buffer to be written, file descriptor and also the number of bytes to write to the server. The server does the write function and appends the appropriate message to the file and returns the number of bytes actually written or -1 along with the errno value back to the libnet file. Errno value is then set and an appropriate error message gets printed out. Otherwise it return the number of bytes written back to the client. The net close function sends the file descriptor to the server. This function return a 0 on success or an errno value along with -1 back to the libnet where an appropriate error messages get printed out.

### **Notes about our program:**

Each time we try to access a particular port, the port number has to be manually changed in the header file (libnetfiles.h) and the main of netfileserver.c where getaddrinfo is called. Our makefile is meant to be used on our client side, and assumes that the client program is called "client.c" and creates an executable called client. However, you can change this/compile it any other way by editing or not using the make. The client is required to send the file descriptor value to the other functions after it is returned from the netopen function.

Extension A:

Our program implements extension A. Our program creates a separate struct for each client with the required variables for our program,