EEP-703 Computer Network Lab

Assignment3-Client-Server stubs in C/C++ in UNIX environment

Harshit Kumar Gupta
2013EET2369

Computer Technology
Department Of Electrical Engineering
IIT DELHI

January 30, 2014

Contents

Chapter 1

PROBLEM STATEMENT

Basics of Network Programming: An introduction to the basics of writing client-server stubs in C/C++ in UNIX environment.

1. Problem Statement (a) Using C-C++ develop a clientserver scenario in which server acts as a data provider to clients. The server holds a student directory which contains Name, Entry Number and Email ID for each student. A client can request for a student's email ID by providing either his-her name or entry number. In response server should return the corresponding student record (all fields). Use TCP sockets and a text file for database which has been mailed to the group.

2. Problem Statement (b) Implement a query based system as follows: RETURN <year> (e.g. RETURN 2012 should return records of all students with entry in 2012) RETURN * (return all records) ADD <name> <entry no> <email id> (e.g. add a new student record for given arguments)

3. Problem Statement (c) Upgrade your server code to serve multiple clients simultaneously. Please note that server must be able to handle several write requests at the same time.

Chapter 2

ABSTRACT

The Intention of the C Code is to make use of Socket Programming to create a Server and a Socket Client on a TCP approach where we are supposed to send the Search string from the Client side and send it to the Server. The Server then in return send backs the entire record for that student and displays it on the Client terminal.

Chapter 3

INTRODUCTION

In all computer networks, one of the computers acts as a server (for applications, data, services) to client computers. In this assignment, we will learn how to develop the C/C++ code which is used to program this functionality on the client and on the server.

C is one of the most widely used programming languages of all time, and C compilers are available for the majority of available computer architectures and operating systems. , C has facilities for structured programming and allows lexical variable scope and recursion, while a static type system prevents many unintended operations. Its design provides constructs that map efficiently to typical machine instructions, and therefore it has found lasting use in applications that had formerly been coded in assembly language, most notably system software like the Unix computer operating system.

Chapter 4

SPECIFICATIONS AND ASSUMPTIONS

Specifications

1. bind() function has been used to bind the hostaddress
2. buffer size used is 256 as it is sufficient.
3. The input file database.txt is stored in the same path.
4. Socket Programming has been implemented in C language.

Assumptions

1. There are as such no multiple entries for the same student.
2. The search String accept character Values only.
3. Error handling has been done accordingly for non existent records or as such.

Chapter 5

LOGIC USED/METHODOLOGY

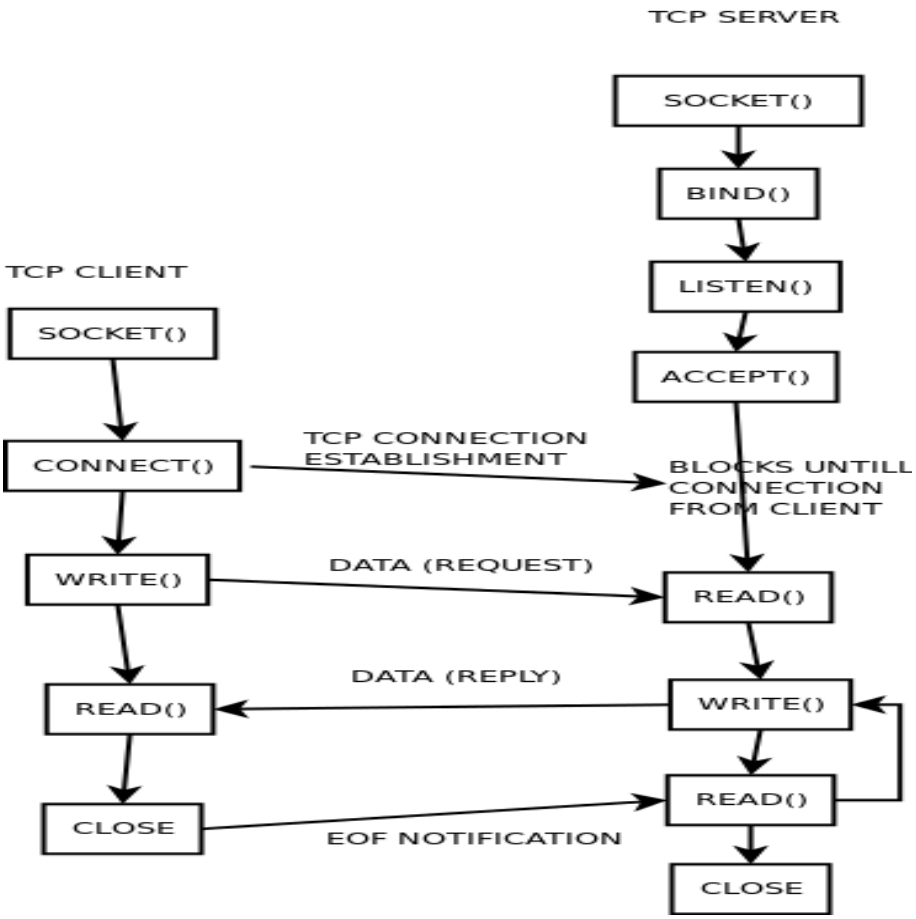
The methodology that is used for developing this project work is defined below:

1. First of all a Server is To be Created and the port no is defined.
2. The Server Code also connects to the Database File where Records are Saved.
3. A buffer is defined so as to save the Input file.
4. Now the Socket function is called and structure Intialised.
5. Host address is bind using the bind() call.
6. Now the input Search string say name is sent to the server.
7. Server searches for the String match and returns the Corresponding Record value.
8. If the User eneters some Invalid value , the error message is displayed.

Chapter 6

FLOWCHART

Flowchart



Chapter 7

Execution Directives

In the terminal of LINUX system the following commands are executed in order to create a network topology and its analysis.

1. Simply Compile the file writing gcc -o QueryServer server.c service.c -lpthread
2. Set the Server host to telnet <hostaddress> <Portno>
3. Ensure that there exists a file named database.txt
4. Now simply run the file using ./QueryServer Command
5. Pass the Commands to the Telnet Client-
   name <NAME>
   entry <ENTRYNO>
   return *
   return <YEAR>
   add <NAME> <ENTRYNO> <EMAILID
   quit

Chapter 8

RESULTS AND CONCLUSIONS

We find that for the given search values the code seems to work perfectly fine and returns the value exactly as the record of that student which is supposed to be accessed.