24. Online Calculator based on the Client Server Socket Programming

Harsh Sahay, Assistant Professor ,Department of Computer Science and Engineering

DAV Institute of Engineering and Technology,Betla Road Palamau,Daltonganj,

822126,India sahayharsh53@gmail.com

ABSTRACT

In client server computing, the clients request a resource and the server provides that resource. A server may serve multiple clients at the same time while a client is in contact with only one server. Both the client and server usually communicate via a computer network but sometimes they may reside in the same system. Here in this work, we have developed online calculator by socket programming. Basic requests those are sent by clients are serviced by remote server.

Keywords— *c,c++,tcp-ip,Linux.*

INTRODUCTION

Client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Client/Server is a term used to describe a computing model for the development of computerized systems. This model is based on the distribution of functions between two types of independent and autonomous processors: servers and clients. A client is any process that requests specific services from server processes. A server is a process that provides requested services for clients. Client and server processes can reside in the same computer or in different computers connected by a network.[1] According to MIS terminology, Client/Server computing is new technology that yields solutions to many data management problems faced by modern organizations. The term Client/Server is used to describe a computing model for the development of computerized systems. This model is based on distribution of functions between two types of independent and autonomous processes: Server and Client. The client/Server architecture is based on hardware and software components that interacts to form a system. This system includes three main components: Clients, Servers, Communication middleware [2]

The client is any computer process that requests services from the server. The client is also known as the front-end- application, reflecting the fact that the end user usually interacts with the client process. The server is any computer process providing services to the clients. The server is also known as the back-end application, reflecting the fact that the server process provides the background services for the client process.

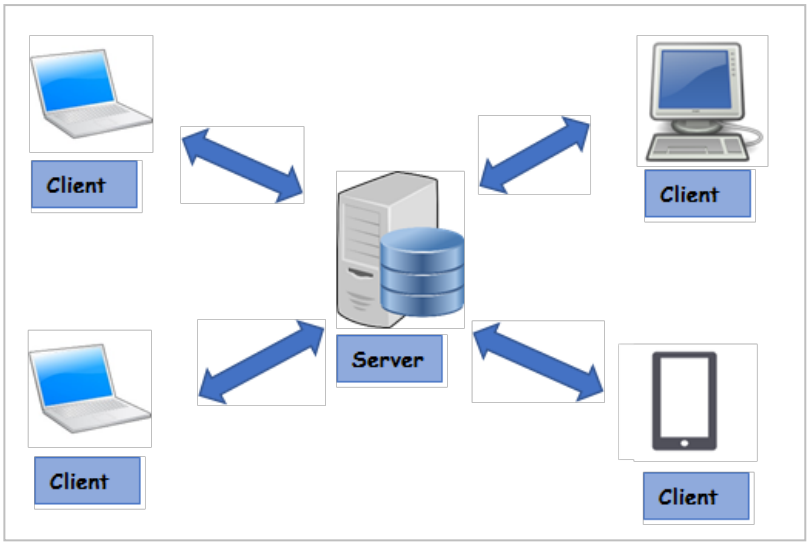
It is any computer process (es) through which clients and servers communicate. The communication middleware, also known as middleware or the communications layers, is made up of several layers of software that aid the transmission of data and control information between clients and servers.[2].

METHODOLOGY

In this methodology we have used programming languages such as c, c++, Tcp-ip, Linux Unix, socket programming. We have developed clients, also servers. The basic requests those are sent by clients such as addition, subtraction, multiplication, division etc. are responded by server to the client.

A. PROGRAMMING

As we see the massages are sent in the form of bits, after going through digital circuit, it generates cipher text. In the receiver side by the reverse process of these digital circuits generates plain text to the receiver side. The cipher text can be hided by the technique of Steganography.




```
#include<stdio.h>
#include<netinet/in.h>    // Internet address family
#include<sys/socket.h>    //Internet Protocol family
#include<fcntl.h>         //file control options
#include<sys/types.h>     //data types
#include<unistd.h>        //standard symbolic constants and types
int choice(int);
int main()
{
int sd,newsd;   // File Descriptor
//char wrbuf[100];
//char rdbuf[100];
struct sockaddr_in s_addr,c_addr;
sd=socket(PF_INET,SOCK_STREAM,0);
if(sd<0)
{
perror("socket");
}
s_addr.sin_family=PF_INET;
s_addr.sin_port=htons(2012);  s_addr.sin_addr.s_addr=inet_addr("0.0.0.0");

if(bind(sd,(struct sockaddr*)&s_addr,sizeof(s_addr))==0
printf("bind success\n");
else
{
perror("bind");
return;
}
listen(sd,1);
int len;
len=sizeof(c_addr);
printf("waiting for connectioni\n");
newsd=accept(sd,(struct sockaddr*)&c_addr,&len);
if(newsd==-1)
{
perror("accept");
      return;
      }
//if(fork())
//{
int wrbuf;
int rdbuf;
while(1)
{
bzero(rdbuf,100);
read(newsd,rdbuf,20);
write(newsd,wrbuf,20);
```

```
calculate(wrbuf);

//printf("read for data\n");
//printf("rdbuf=%s\n",rdbuf);
//printf("enter data\n");
//gets(wrbuf);
read(newsd,wrbuf,20);
}
return 0;
}

int calculate(int choice)
{
                clrscr();
                float a, b, res;
                char choice, ch;
                do
                {
                                printf("1.Addition\n");
                                printf("2.Subtraction\n");
                                printf("3.Multiplication\n");
                                printf("4.Division\n");
                                printf("5.Exit\n\n");
                                //printf("Enter Your Choice");
                                //scanf("%d",&choice);
                                switch(choice)
                                {
                                                case '1' : printf("Enter two number : ");
                                                                scanf("%f%f",&a,&b);
                                                                res=a+b;
                                                                //printf("Result = %f",res);
                        return(res);
                                                                break;
                                                case '2' : printf("Enter two number : ");
                                                                scanf("%f%f",&a,&b);
                                                                res=a-b;
                                                                //printf("Result = %f",res);
                         return(res);
                                                                break;
                                                case '3' : printf("Enter two number : ");
                                                                scanf("%f%f",&a,&b);
                                                                res=a*b;
                                                                //printf("Result = %f",res);
                         return(res);
                                                                break;
                        case '4' : printf("Enter two number : ");
                                                                scanf("%f%f",&a,&b);
                                                                res=a/b;
                                                                //printf("Result = %f",res);
                            return(res);
                                                                break;
                                                case '5' : exit(0);
                                                                break;
                                                default : printf("Wrong Choice..!!");
                                                                break;
                                }
                                printf("\n-------------------------------------\n");
                 }while(choice!=5 && choice!=getchar());
                 getch();
}

Bshop School near bahu bazar
sunday     11.00-1.00
Tuesday,Thirsday holiday
Rest of all 5.00-7.00

//client
#include<stdio.h>
#include<sys/socket.h>
#include<unistd.h>
main()
{
int sd,newsd,len;
//char wrbuf[128],rdbuf[128];
```

```
struct sockaddr_in s_addr,c_addr;
sd=socket(PF_INET,SOCK_STREAM,0);
if(sd<0)
{
perror("socket");
return;
      }
s_addr.sin_family=PF_INET;
s_addr.sin_port=htons(2012);
s_addr.sin_addr.s_addr=inet_addr("127.0.0.1");
connect(sd,(struct sockaddr*)&s_addr,sizeof(s_addr));
printf("connect success");
//if(fork())
//{
int choice;
int result;
while(1)
{
//bzero(wrbuf,128);
printf("enter choice\n");
gets(choice);
read(sd,choice,20);
//}
//}
//else
//{
//char rdbuf[128];
//while(1)
write(sd,result,20);
printf("%s\n",result);
}
}
```

CONCLUSION

It seems the excellent way to design client server architecture by socket programming. We can design it excellently by languages c, c + +.linux,unix,tcp-ip. Client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. On line calculator is designed to the server side, and requests are sent by the clients to the remote server i.e. server side.

REFERENCES:

[1] Copyright @ www.bcanotes.com

[2] An introduction to client server computing subhash Chandra yadav,Sanjay kumar singh