ASSIGNMENT NO:2 **DATE:31/01/2017**

PROGRAM TITLE:Develop a Client Server Application using TCP/IP where the client will send 2 operands and a operator to the server using commandline argument in "operand1 operator operand2" format and the server will calculate the result and display it. Allowed operators are '+,-,*,/'.

PROGRAM CODE:

server.c

```
#include<stdio.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<stdlib.h>
#include<string.h>

#define MAXPENDING 5
#define RECVBUFSIZE 10

void calc(char s[])
{
	int i,j;
	float a,b,c;
	char op1[RECVBUFSIZE],op2[RECVBUFSIZE],op;
	for(i=0;s[i] != ' ';i++)
	{
		op1[i]=s[i];
	}
	op1[i]='\0';
	op=s[++i];
	if(op == '\\')
		op=s[++i];
	for(i=i+1,j=0;s[i] != '\0';i++,j++)
	{
		op2[j]=s[i];
	}
	op2[j]='\0';
	a=atof(op1);
	b=atof(op2);
	switch(op)
	{
		case '+': c=a+b;
			  break;
		case '-': c=a-b;
			  break;
		case '*': c=a*b;
			  break;
		case '/': c=a/b;
			  break;
		default: printf("\n\tWrong Input.\n");
	}
	printf("\n\t%.2f %c %.2f = %.2f\n",a,op,b,c);
}
main()
{
	int servSock, clientAddrLen, clientSock, recvBufSize;
	float res;
	struct sockaddr_in clientAddr,serverAddr;
	char server_ip[] = "127.0.0.1";
	unsigned short server_port=25051;
	char recvBuf[RECVBUFSIZE],sendBuf[RECVBUFSIZE];
```

```
      bzero(&serverAddr,sizeof(serverAddr));
      serverAddr.sin_family = AF_INET;//Internet Address family
      serverAddr.sin_port = htons(server_port);//Local Port address
      inet_aton(server_ip,(&serverAddr.sin_addr));
      if((servSock=socket(AF_INET,SOCK_STREAM,0))<0)
      {
            printf("\n\tSocket Error.\n");
            exit(1);
      }
      printf("\n\tSERVER: Socket Created.\n");
      if((bind(servSock,(struct sockaddr*)&serverAddr, sizeof(serverAddr)))<0)//-1
indicates failure
      {
            printf("\n\tBind Error.\n");
            close(servSock);//Closing the socket
            exit(1);
      }
      printf("\n\tSERVER: Binded Successfully.\n");
      if(listen(servSock,MAXPENDING)<0)//-1 indicates failure
      {
            printf("\n\tListen Error.\n");
            close(servSock);//Closing the socket
            exit(1);
      }
      printf("\n\tSERVER: Listening to Clients..\n\tPress Ctrl+C to stop the
server.\n");
      while(1)//Run forever
      {
            clientAddrLen = sizeof(clientAddr);
            if((clientSock=accept(servSock,(struct sockaddr
*)&clientAddr,&clientAddrLen))<0)
            {
                  printf("\n\tAccept Error.\n");
                  close(servSock);
                  exit(1);
            }
            if(recvBufSize=recv(clientSock,recvBuf,RECVBUFSIZE,0)<0)
            {
                  printf("\n\tReceive Error.\n");
                  close(clientSock);
                  continue;
            }
            calc(recvBuf);
            close(clientSock);
      }
      close(servSock);
}
```

client.c

```
#include<stdio.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<stdlib.h>
#include<string.h>

#define BUFSIZE 10

main(int argc,char **argv)
{
      int clientSock;
      struct sockaddr_in serverAddr;
      char server_ip[] = "127.0.0.1";
      unsigned short server_port=25051;
```

```
	char sendBuf[BUFSIZE],recvBuf[BUFSIZE];
	if(argc!=4)
	{
		if(argc==1)
		{
			printf("\n\tNo argument\n");
			exit(1);
		}
		else
		{
			printf("\n\tInput must be in this format:\"op1 op op2\" \n");
			exit(1);
		}
	}
	strcpy(sendBuf,argv[1]);
	strcat(sendBuf," ");
	strcat(sendBuf,argv[2]);
	strcat(sendBuf," ");
	strcat(sendBuf,argv[3]);
	bzero(&serverAddr,sizeof(serverAddr));
	serverAddr.sin_family = AF_INET;//Internet Address family
	serverAddr.sin_port = htons(server_port);//Local Port address
	inet_aton(server_ip,(&serverAddr.sin_addr));
	if((clientSock=socket(PF_INET,SOCK_STREAM,0))<0)
	{
		printf("\n\tSocket Error.\n");
		exit(1);
	}
	printf("\n\tCLIENT: Socket Created.\n");
	if((connect(clientSock,(struct sockaddr*)&serverAddr,sizeof(serverAddr)))<0)
	{
		printf("\nConnect Error\n");
		close(clientSock);
		exit(1);
	}
	printf("\n\tCLIENT: Connected.\n");
	if(write(clientSock,sendBuf,sizeof(sendBuf))<0)
	{
		printf("\n\tSend Error.\n");
		exit(1);
	}
	printf("\n\tCLIENT: Sent.\n");
	close(clientSock);
}
```

OUTPUT:

Server

```
[student@localhost 2]$ ./server

	SERVER: Socket Created.

	SERVER: Binded Successfully.

	SERVER: Listening to Clients..
	Press Ctrl+C to stop the server.

	2.00 + 6.00 = 8.00

	2.00 - 6.00 = -4.00
```

```
      Wrong Input.

      2.00 x 6.00 = -4.00

      2.00 / 6.00 = 0.33
^C
```

<u>Client</u>

```
[student@localhost 2]$ ./client 2 + 6

      CLIENT: Socket Created.

      CLIENT: Connected.

      CLIENT: Sent.
[student@localhost 2]$ ./client 2 - 6

      CLIENT: Socket Created.

      CLIENT: Connected.

      CLIENT: Sent.
[student@localhost 2]$ ./client 2 * 6

      Input must be in this format:"op1 op op2"
[student@localhost 2]$ ./client 2 x 6

      CLIENT: Socket Created.

      CLIENT: Connected.

      CLIENT: Sent.
[student@localhost 2]$ ./client 2 / 6

      CLIENT: Socket Created.

      CLIENT: Connected.

      CLIENT: Sent.
```