ASSIGNMENT NO:1 **DATE:17/01/2017**

PROGRAM TITLE:Develop a Client Server Application using TCP/IP where the client will send temperature in degree Fahrenheit using commandline argument, the server will convert it to degree Centigrade and send the result back to the client. The Client will display the result.

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

main()
{
	int servSock, clientAddrLen, clientSock, recvBufSize;
	float f, c;
	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;
            }
            f=atof(recvBuf);
            c=(f-32)*(5/9.0);
            sprintf(sendBuf,"%f",c);
            if(write(clientSock,sendBuf,sizeof(sendBuf))<0)
            {
                printf("\n\tSend Error.\n");
                close(clientSock);
                continue;
            }
            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!=2)
        {
            printf("\n\tNo argument or more than one argument.\n");
            exit(1);
        }
        strcpy(sendBuf,argv[1]);
        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");
        if(recv(clientSock,recvBuf,BUFSIZE,0)<0)
        {
            printf("\n\tReceive Failed.\n");
            close(clientSock);
        }
        printf("\n\tCLIENT: Received.\n");
        printf("\t%sdeg F = %sdeg C\n",sendBuf,recvBuf);
        close(clientSock);
}
```

OUTPUT:

Server

```
[student@localhost 1]$ ./server

     SERVER: Socket Created.

     SERVER: Binded Successfully.

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

Client

```
Output 1:
[student@localhost 1]$ ./client

     No argument or more than one argument.

Output 2:
[student@localhost 1]$ ./client 40

     CLIENT: Socket Created.

     CLIENT: Connected.

     CLIENT: Sent.

     CLIENT: Received.
     40deg F = 4.444445deg C

Output 3:
[student@localhost 1]$ ./client -40

     CLIENT: Socket Created.

     CLIENT: Connected.

     CLIENT: Sent.

     CLIENT: Received.
     -40deg F = -40.000000deg C
```