**CLIENT SIDE SCRIPTING:-**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<netdb.h>

void error(char \*msg)

{

perror(msg);

exit(0);

}

int main(int argc, char \*argv[])

{

int sockfd,portno,n;

struct sockaddr\_in serv\_addr;

struct hostent \*server;

char buffer[256];

if (argc < 3)

{

fprintf(stderr,"usage %s hostname port\n", argv[0]);

exit(0);

}

portno = atoi(argv[2]);

sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

if (sockfd < 0)

{

error("Error opening socket");

}

server = gethostbyname(argv[1]);

if(server == NULL)

{

fprintf(stderr,"Error,no such host\n");

exit(0);

}

bzero((char \*) &serv\_addr,sizeof(serv\_addr));

serv\_addr.sin\_family = AF\_INET;

bcopy((char \*)server->h\_addr,(char \*)&serv\_addr.sin\_addr.s\_addr,server->h\_length);

serv\_addr.sin\_port=htons(portno);

if(connect(sockfd,&serv\_addr,sizeof(serv\_addr)) < 0)

{

error("Error connecting");

}

printf("Enter The Message:\ns");

bzero(buffer,256);

fgets(buffer,255,stdin);

n = write(sockfd,buffer,strlen(buffer));

if(n<0)

{

error("Error writing to socket");

}

bzero(buffer,256);

n = read(sockfd,buffer,255);

if(n < 0)

{

error("Error reading");

}

printf("%s\n",buffer);

return 0;

}

**SERVER SIDE SCRIPTING:-**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

void error(char \*msg)

{

perror(msg);

exit(1);

}

int main(int argc, char \*argv[])

{

int sockfd,newsockfd,portno,clilen,n;

struct sockaddr\_in serv\_addr;

struct sockaddr\_in cli\_addr;

char buffer[256];

if (argc < 2)

{

fprintf(stderr, "ERROR, no port provided\n");

exit(1);

}

sockfd=socket(AF\_INET, SOCK\_STREAM, 0);

if (sockfd < 0)

{

error("Error Opening socket");

}

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

portno = atoi(argv[1]);

serv\_addr.sin\_family=AF\_INET;

serv\_addr.sin\_addr.s\_addr=INADDR\_ANY;

serv\_addr.sin\_port=htons(portno);

if (bind(sockfd,(struct sockaddr \*) &serv\_addr,sizeof(serv\_addr))<0)

{

error("error binding");

}

listen(sockfd,5);

clilen = sizeof(cli\_addr);

newsockfd = accept(sockfd,(struct sockaddr \*) &cli\_addr, &clilen);

if(newsockfd < 0)

{

error("Error on accept");

}

bzero(buffer,256);

n = read(newsockfd,buffer,255);

if(n<0)

{

error("Reading from socket");

}

printf("Here is message: %s\n",buffer);

n = write(newsockfd,"I got your message", 18);

if(n<0)

{

error("Error writing");

}

return 0;

}