

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<stdio.h>#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<errno.h>

#include<string.h>

#include<unistd.h>

#include<sys/types.h>

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

// define server socket

int server\_socket;

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

// define server address

struct sockaddr\_in server\_address;

server\_address.sin\_family= AF\_INET;

server\_address.sin\_addr.s\_addr= INADDR\_ANY;

server\_address.sin\_port= htons(64000);

// bind

bind(server\_socket,(struct sockaddr\*)&server\_address,sizeof(server\_address));

listen(server\_socket,5);

printf("\n server started listening\n");

while(1){

int client\_socket;

client\_socket=accept(server\_socket,NULL,NULL);

char\* msg="Welcome to the server of Mohit";

write(client\_socket,msg,strlen(msg));

printf("message sent ....\n");

char buff[256];

read(client\_socket,buff,sizeof(buff));

printf("Message received from client: %s\n",buff);

char\* p=strstr(buff,"END");

if(p!=NULL){

printf("Serevr is going to off\n");

break;

}

}

}

#include<stdlib.h>

#include<unistd.h>

#include<errno.h>

#include<string.h>

#include<unistd.h>

#include<sys/types.h>

#include<fcntl.h>

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

// define client socket

int client\_socket;

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

// define client address

struct sockaddr\_in server\_address;

server\_address.sin\_family= AF\_INET;

server\_address.sin\_addr.s\_addr= INADDR\_ANY;

server\_address.sin\_port= htons(64000);

int connection\_status=connect(client\_socket,(struct sockaddr\*)&server\_address,sizeof(server\_address));

if( connection\_status ==-1){

printf("Error happen\n");

exit(1);

}

char buff[256];

read(client\_socket,buff,sizeof(buff));

printf("Message received: %s\n",buff);

char \*msg;

msg=(char\*)malloc(sizeof(char)\*50);

printf("Enter message for server:");

fgets(msg,50,stdin);

write(client\_socket,msg,strlen(msg));

close(client\_socket);

return 0;

}