**Code for Server**

**server.c**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <netinet/in.h>

#include <netinet/ip.h>

#include <arpa/inet.h>

#include <unistd.h>

#include <time.h>

#define STDIN 0

int main()

{

system("clear");

int serverfd=socket(AF\_INET,SOCK\_STREAM,IPPROTO\_TCP);

if(serverfd==-1)

{

perror("Socket creation failed try again\n");

exit(0);

}

struct sockaddr\_in server,client;

server.sin\_family=AF\_INET;

server.sin\_port=htons(1110);

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

int opt=1;

setsockopt(serverfd,SOL\_SOCKET,SO\_REUSEADDR,&opt,sizeof(opt));

int b=bind(serverfd,(struct sockaddr\*)&server,sizeof(server));

if(b==-1)

{

perror("Error Bind failure\n");

exit(0);

}

listen(serverfd,5);

fd\_set readfd;

FD\_ZERO(&readfd);

int size = sizeof(server);

int clientfd=accept(serverfd,(struct sockaddr\*)&client,&size);

if(clientfd==-1)

{

perror("Accept failed try again!\n");

exit(0);

}

printf("I am sever %s \n",inet\_ntoa(client.sin\_addr));

struct timeval tv;

char snd[50],rcv[50];

while(1)

{

FD\_SET(STDIN,&readfd);

FD\_SET(clientfd,&readfd);

tv.tv\_sec=15;

tv.tv\_usec=5000000;

int s=select(clientfd+1,&readfd,NULL,NULL,&tv);

if(s==-1)

perror("select error\n");

else if(s==0)

printf("Timeout occured!!\n");

else

{

if(FD\_ISSET(STDIN,&readfd))

{

scanf("%s",snd);

send(clientfd,snd,strlen(snd),0);

}

if(FD\_ISSET(clientfd,&readfd))

{

int r=recv(clientfd,rcv,sizeof(rcv),0);

rcv[r]='\0';

if(r==0)

{

printf("Client is now closed\n");

break;

}

time\_t t;

int h,m,s;

time(&t);

struct tm \*local = localtime(&t);

h=local->tm\_hour;

m=local->tm\_min;

s=local->tm\_sec;

system("notify-send Unread \"Message from client\"");

printf("CILENT MESSAGE: %02d:%02d:%02d : %s\a\n",h,m,s,rcv);

}

}

}

}