

**ASSIGNMENT 6**

|  |  |
| --- | --- |
| Name: | Wilson Vidyut Doloy |
| Reg No: | 19BCE1603 |
| Subject: | CSE1004 Network and Communication Lab |
| Slot: | L52+L53 (Prof. Ganesan R) |
| Date: | 26 August 2020 |

1. Execute the code - both in client and server

2. Then implement the same in the form of chat application. Whatever message sent should be sent along the current time.

**Server Side Code:**

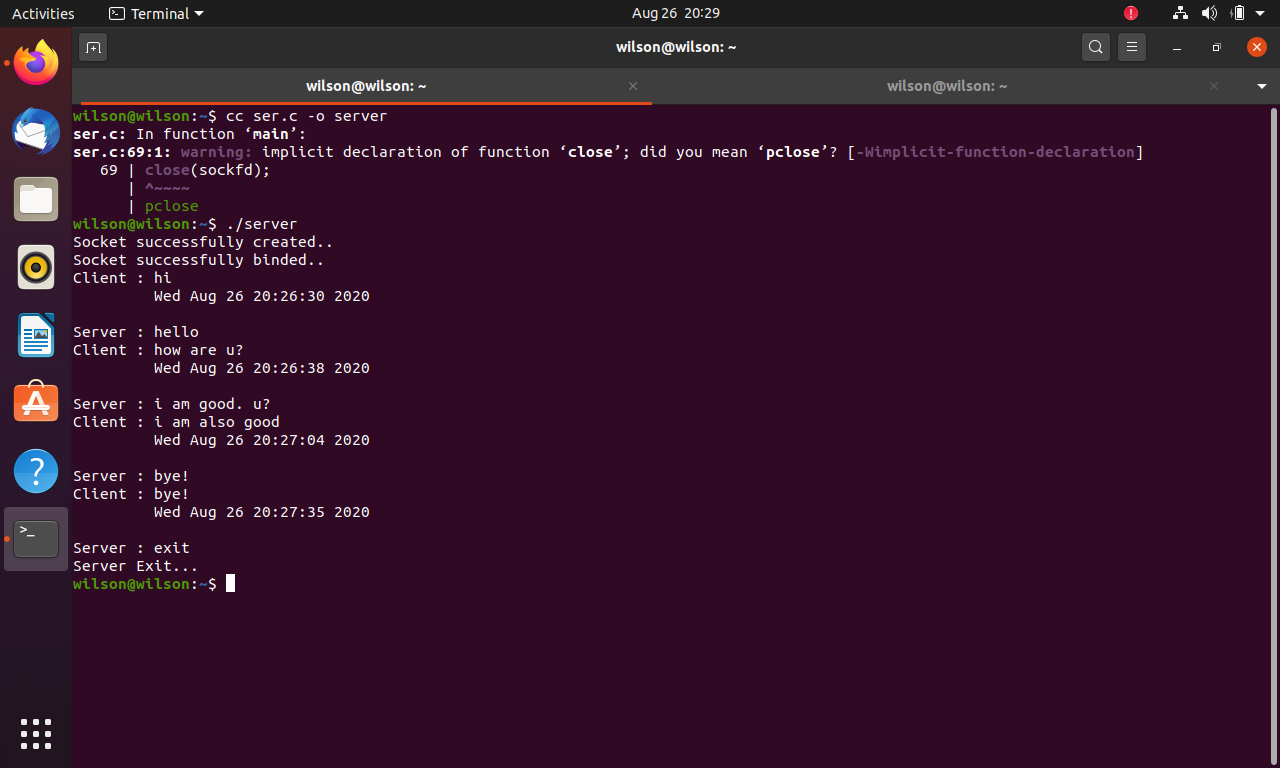
#include<stdio.h>  
#include<netinet/in.h>  
#include<sys/types.h>  
#include<sys/socket.h>  
#include<netdb.h>  
#include<string.h>  
#include<stdlib.h>  
#include<time.h>  
#define MAX 80  
#define PORT 43454  
#define SA struct sockaddr  
void func(int sockfd)  
{  
char buff[MAX],buf[MAX];  
int n,clen;  
time\_t tick;  
struct sockaddr\_in cli;  
clen=sizeof(cli);  
for(;;)  
{  
bzero(buff,MAX);  
bzero(buf,MAX);  
recvfrom(sockfd,buff,sizeof(buff),0,(SA \*)&cli,&clen);  
recvfrom(sockfd,buf,sizeof(buf),0,(SA \*)&cli,&clen);  
printf("Client : %s \t %s\n",buff,buf);  
printf("Server : ");  
bzero(buff,MAX);  
bzero(buf,MAX);  
  
n=0;  
while((buff[n++]=getchar())!='\n');  
sendto(sockfd,buff,sizeof(buff),0,(SA \*)&cli,clen);  
  
tick=time(NULL);  
snprintf(buf,sizeof(buf),"%s",ctime(&tick));  
sendto(sockfd,buf,sizeof(buf),0,(SA \*)&cli,clen);  
bzero(buf,sizeof(buf));  
if(strncmp("exit",buff,4)==0)  
{  
printf("Server Exit...\n");  
break;  
}  
}  
}  
int main()  
{  
int sockfd;  
struct sockaddr\_in servaddr;  
sockfd=socket(AF\_INET,SOCK\_DGRAM,0);  
if(sockfd==-1)  
{  
printf("socket creation failed...\n");  
exit(0);  
}  
else  
printf("Socket successfully created..\n");  
bzero(&servaddr,sizeof(servaddr));  
servaddr.sin\_family=AF\_INET;  
servaddr.sin\_addr.s\_addr=htonl(INADDR\_ANY);  
servaddr.sin\_port=htons(PORT);  
if((bind(sockfd,(SA \*)&servaddr,sizeof(servaddr)))!=0)  
{  
printf("socket bind failed...\n");  
exit(0);  
}  
else  
printf("Socket successfully binded..\n");  
func(sockfd);  
close(sockfd);  
}

**Client Side Code:**

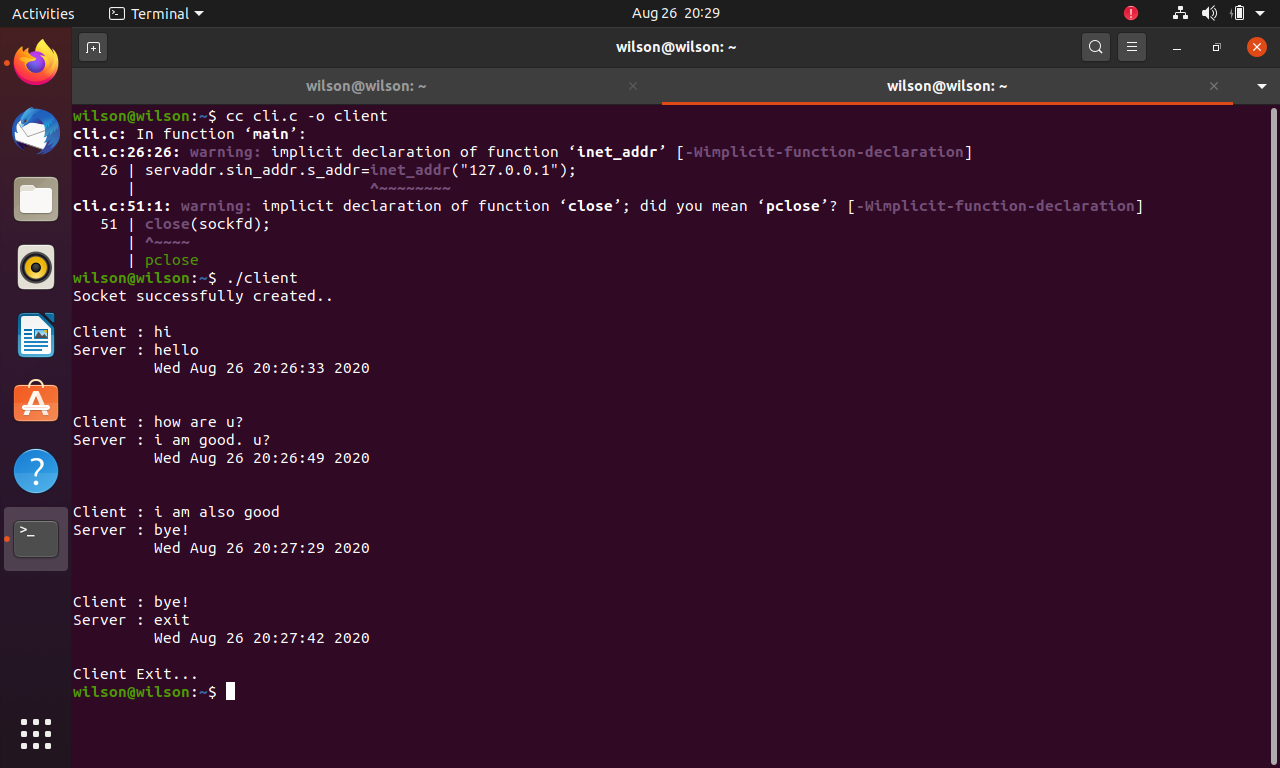
#include<sys/socket.h>  
#include<netdb.h>  
#include<string.h>  
#include<stdlib.h>  
#include<stdio.h>  
#include<time.h>  
#define MAX 80  
#define PORT 43454  
#define SA struct sockaddr  
int main()  
{  
char buff[MAX],buf[MAX];  
int sockfd,len,n;  
struct sockaddr\_in servaddr;  
time\_t tick;  
sockfd=socket(AF\_INET,SOCK\_DGRAM,0);  
if(sockfd==-1)  
{  
printf("socket creation failed...\n");  
exit(0);  
}  
else  
printf("Socket successfully created..\n");  
bzero(&servaddr,sizeof(len));  
servaddr.sin\_family=AF\_INET;  
servaddr.sin\_addr.s\_addr=inet\_addr("127.0.0.1");  
servaddr.sin\_port=htons(PORT);  
len=sizeof(servaddr);  
for(;;)  
{  
printf("\nClient : ");  
n=0;  
while((buff[n++]=getchar())!='\n');  
sendto(sockfd,buff,sizeof(buff),0,(SA \*)&servaddr,len);  
bzero(buff,sizeof(buff));  
tick=time(NULL);  
snprintf(buf,sizeof(buf),"%s",ctime(&tick));  
sendto(sockfd,buf,sizeof(buf),0,(SA \*)&servaddr,len);  
bzero(buf,sizeof(buf));  
  
recvfrom(sockfd,buff,sizeof(buff),0,(SA \*)&servaddr,&len);  
recvfrom(sockfd,buf,sizeof(buf),0,(SA \*)&servaddr,&len);  
  
printf("Server : %s \t %s\n",buff,buf);  
if(strncmp("exit",buff,4)==0)  
{  
printf("Client Exit...\n");  
break;  
}  
}  
close(sockfd);  
}

**Output:**

**Server:**

****

**Client:**

****