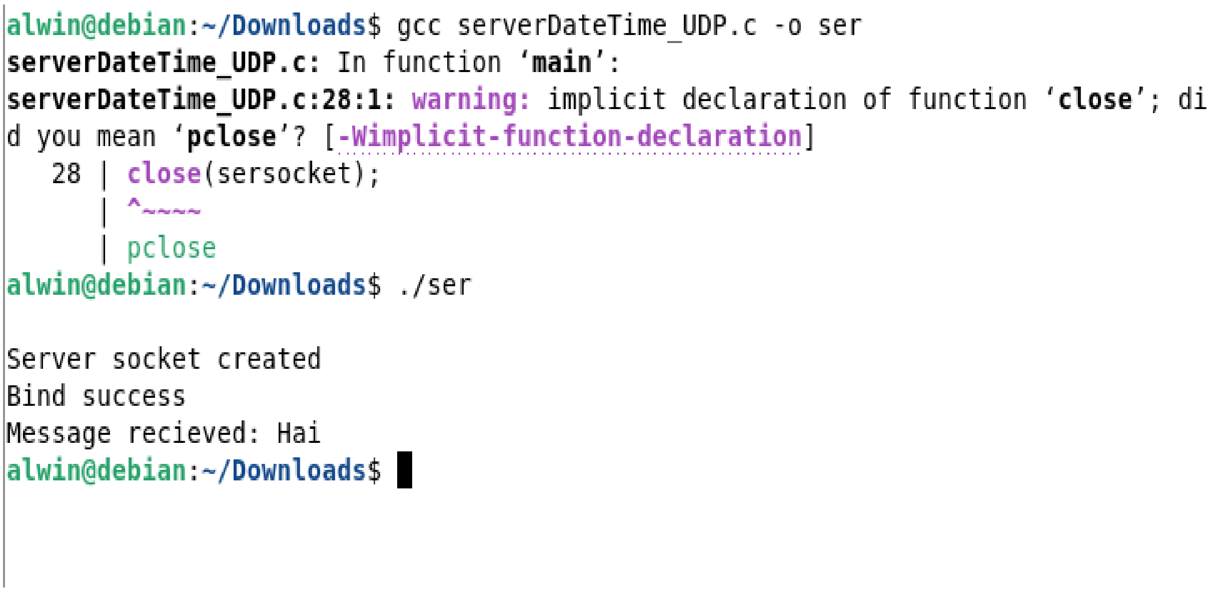
SOCKET PROGRAMMING IN C

Experiment-6: A simple UDP Server-Client program which displays the current calendar time.

serverDateTime_UDP.c

```
#include<stdio.h>
#include<string.h>
#include<time.h>
#include<arpa/inet.h>
#define port 4000
void main(){
time_t ticks;
struct sockaddr_in serveraddr,newaddr;
int sersocket,s,size;
char buffer[100],str[100];
sersocket=socket(AF_INET,SOCK_DGRAM,0);
if(sersocket>0)
printf("\nServer socket created");
serveraddr.sin_family=AF_INET;
serveraddr.sin_port=htons(port);
serveraddr.sin_addr.s_addr=htonl(INADDR_ANY);
s=bind(sersocket,(struct sockaddr *)&serveraddr,sizeof(serveraddr));
if(s==0)
printf("\nBind success");
size=sizeof(newaddr);
recvfrom(sersocket,buffer,sizeof(buffer),0,(struct sockaddr *)&newaddr,&size);
printf("\nMessage recieved: %s",buffer);
ticks=time(NULL);
strcpy(str,ctime(&ticks));
snprintf(buffer,sizeof(buffer),"%s",str);
sendto(sersocket,buffer,sizeof(buffer),0,(struct sockaddr
*)&newaddr,sizeof(newaddr));
printf("\n");
close(sersocket);
}
```

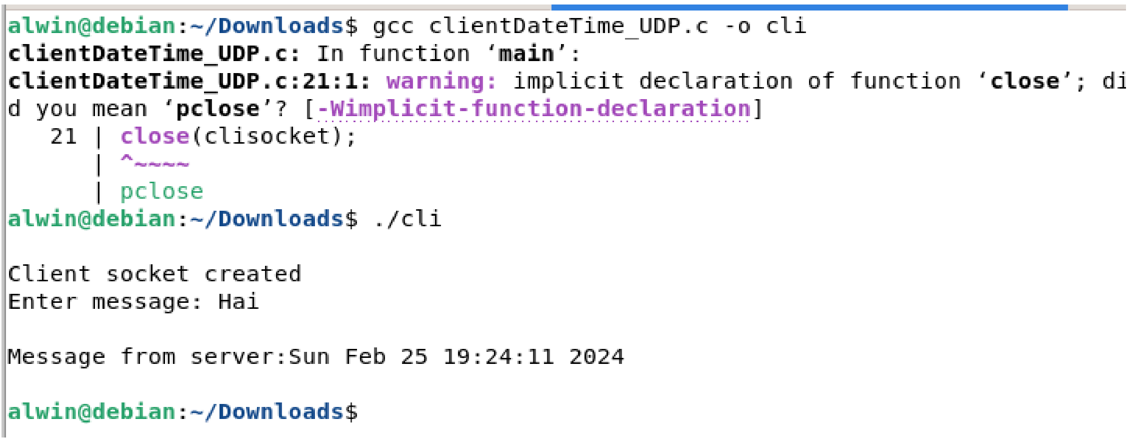


```
alwin@debian:~/Downloads$ gcc serverDateTime_UDP.c -o ser
serverDateTime_UDP.c: In function ‘main’:
serverDateTime_UDP.c:28:1: warning: implicit declaration of function ‘close’; di
d you mean ‘pclose’? [-Wimplicit-function-declaration]
   28 | close(sersocket);
      | ^~~~~
      | pclose
alwin@debian:~/Downloads$ ./ser

Server socket created
Bind success
Message recieved: Hai
alwin@debian:~/Downloads$ 
```

clientDateTime_UDP.c

```
#include<stdio.h>
#include<arpa/inet.h>
#define port 4000
void main()
{
struct sockaddr_in serveraddr,newaddr;
int clisocket,size;
char buffer[100];
clisocket=socket(AF_INET,SOCK_DGRAM,0);
if(clisocket>0)
printf("\nClient socket created");
```

```
serveraddr.sin_family=AF_INET;
serveraddr.sin_port=htons(port);
serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");
printf("\nEnter message: ");
scanf("%s",buffer);
size=sizeof(newaddr);
sendto(clisocket,buffer,sizeof(buffer),0,(struct
sockaddr*)&serveraddr,sizeof(serveraddr));
recvfrom(clisocket,buffer,sizeof(buffer),0,(struct sockaddr*)&newaddr,&size);
printf("\nMessage from server:%s\n",buffer);
close(clisocket);
}
```



```
alwin@debian:~/Downloads$ gcc clientDateTime_UDP.c -o cli
clientDateTime_UDP.c: In function ‘main’:
clientDateTime_UDP.c:21:1: warning: implicit declaration of function ‘close’; di
d you mean ‘pclose’? [-Wimplicit-function-declaration]
   21 | close(clisocket);
      | ^~~~~
      | pclose
alwin@debian:~/Downloads$ ./cli

Client socket created
Enter message: Hai

Message from server:Sun Feb 25 19:24:11 2024

alwin@debian:~/Downloads$
```

##Implementation

Compile

```
Server : gcc serverDateTime_UDP.c -o ser
Client : gcc clientDateTime_UDP.c -o cli
```

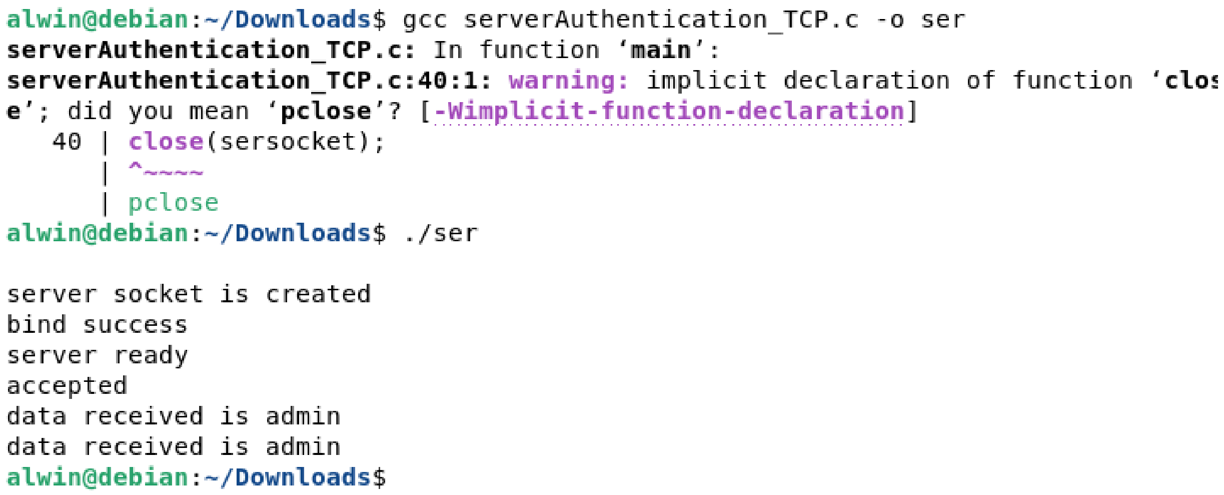
Run

```
Server : ./ser
Client : ./cli
```

Experiment-7: A simple TCP Server-Client program where the client provides the username and password as request and the server authenticates the request and returns the result.

serverAuthentication_TCP.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include<string.h>
# define port 5000
void main()
{
struct sockaddr_in serveraddr,newaddr;
int sersocket,newsocket,s,size;
char buffer1[100],buffer2[100],buffer[100];
sersocket=socket(PF_INET,SOCK_STREAM,0);
if(sersocket>0)
printf("\nserver socket is created");
serveraddr.sin_family= PF_INET;
serveraddr.sin_port= htons(port);
serveraddr.sin_addr.s_addr=htonl(INADDR_ANY);
s=bind(sersocket,(struct sockaddr *)&serveraddr,sizeof(serveraddr));
if(s==0)
printf("\nbind success");
listen(sersocket,1);
size=sizeof(newaddr);
printf("\nserver ready");
newsocket=accept(sersocket,(struct sockaddr *)&newaddr,&size);
if(newsocket>0)
printf("\naccepted");
recv(newsocket,buffer1,1024,0);
printf("\ndata received is %s\n",buffer1);
if(strcmp(buffer1,"admin")==0)
{
recv(newsocket,buffer2,1024,0);
printf("data received is %s\n",buffer2);
if(strcmp(buffer2,"admin")==0)
strcpy(buffer,"\nCONFIRMED\n");
else
strcpy(buffer,"\nEnter valid password\n");
send(newsocket,buffer,sizeof(buffer),0);
}
else
strcpy(buffer,"\nEnter valid username\n");
send(newsocket,buffer,sizeof(buffer),0);
close(sersocket);
}
```

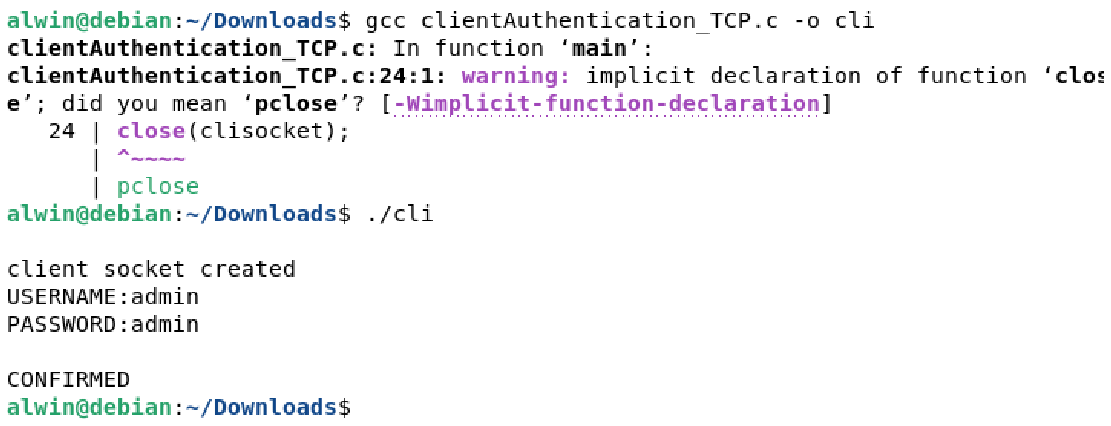


```
alwin@debian:~/Downloads$ gcc serverAuthentication_TCP.c -o ser
serverAuthentication_TCP.c: In function ‘main’:
serverAuthentication_TCP.c:40:1: warning: implicit declaration of function ‘clos
e’; did you mean ‘pclose’? [-Wimplicit-function-declaration]
   40 | close(sersocket);
      | ^~~~~
      | pclose
alwin@debian:~/Downloads$ ./ser

server socket is created
bind success
server ready
accepted
data received is admin
data received is admin
alwin@debian:~/Downloads$
```

clientAuthentication_TCP.c

```
#include<stdio.h>
#include<arpa/inet.h>
# define port 5000
void main()
{
struct sockaddr_in serveraddr;
int clisocket;
char buffer[100];
clisocket=socket(PF_INET,SOCK_STREAM,0);
if(clisocket>0)
```

```
printf("\nclient socket created");
serveraddr.sin_family= PF_INET;
serveraddr.sin_port= htons(port);
serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");
connect(clisocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));
printf("\nUSERNAME:");
scanf("%s",buffer);
send(clisocket,buffer,sizeof(buffer),0);
printf("PASSWORD:");
scanf("%s",buffer);
send(clisocket,buffer,sizeof(buffer),0);
recv(clisocket,buffer,sizeof(buffer),0);
printf("%s",buffer);
close(clisocket);
}
```



```
alwin@debian:~/Downloads$ gcc clientAuthentication_TCP.c -o cli
clientAuthentication_TCP.c: In function ‘main’:
clientAuthentication_TCP.c:24:1: warning: implicit declaration of function ‘clos
e’; did you mean ‘pclose’? [-Wimplicit-function-declaration]
   24 | close(clisocket);
      | ^~~~~
      | pclose
alwin@debian:~/Downloads$ ./cli

client socket created
USERNAME:admin
PASSWORD:admin

CONFIRMED
alwin@debian:~/Downloads$
```

## Implementation

Compile

```
Server : gcc serverAuthentication_TCP.c -o ser
Client : gcc clientAuthentication_TCP.c -o cli
```

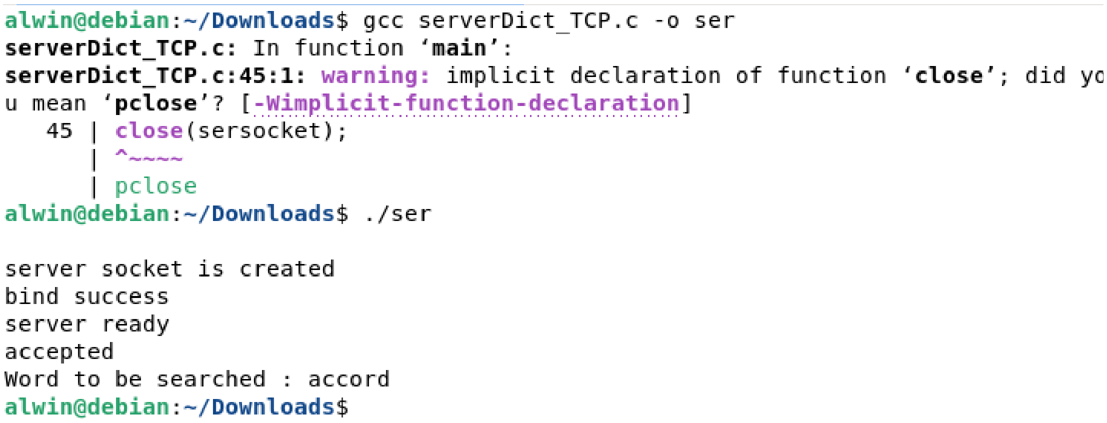
Run

```
Server : ./ser
Client : ./cli
```

Experiment-8: A simple TCP Server-Client program implementing a dictionary with meanings and antonyms.

serverDict_TCP.c

```
#include<stdio.h>
#include<string.h>
#include<arpa/inet.h>
# define port 5000
void main()
{
struct sockaddr_in serveraddr,newaddr;
int sersocket,newsocket,s,size;
int i,c=0;
char buffer[100],word[100],antonym[100];
char dictionary[7][3][100]={"consider", "deem to be","disregard","minute" ,
"infinitely or immeasurably small","significant","accord" , "concurrence of
opinion","withhold","commit" ,"perform an act,usually with a negative
connotation","abstain","utter", "without qualification","partial","zealot", "a fervent
and even militant proponent of something","moderate","wanton","a lewd or lascivious
person","justifiable"};
sersocket=socket(PF_INET,SOCK_STREAM,0);
if(sersocket>0)
printf("\nserver socket is created");
serveraddr.sin_family= PF_INET;
serveraddr.sin_port= htons(port);
serveraddr.sin_addr.s_addr=htonl(INADDR_ANY);
s=bind(sersocket,(struct sockaddr *)&serveraddr,sizeof(serveraddr));
if(s==0)
printf("\nbind success");
listen(sersocket,1);
size=sizeof(newaddr);
printf("\nserver ready");
newsocket=accept(sersocket,(struct sockaddr *)&newaddr,&size);
if(newsocket>0)
printf("\naccepted");
recv(newsocket,buffer,1024,0);
printf("\nWord to be searched : %s\n",buffer);
for(i=0;i<7;i++)
{
if(strcmp(buffer,dictionary[i][0])==0)
{
strcpy(word,dictionary[i][1]);
send(newsocket,word,sizeof(word),0);
strcpy(antonym,dictionary[i][2]);
send(newsocket,antonym,sizeof(antonym),0);
}
else
c++;
}
if(c==7){
strcpy(word,"Word not found in dictionary");
send(newsocket,word,sizeof(word),0);
}
close(sersocket);
}
```

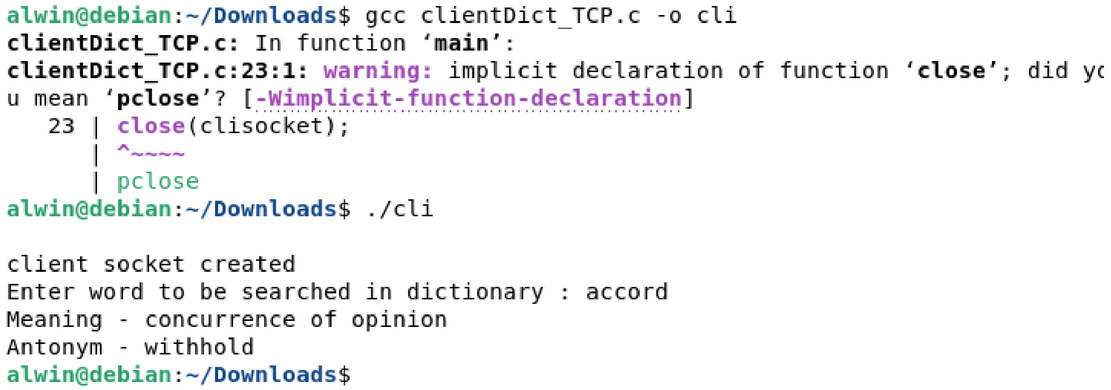


```
alwin@debian:~/Downloads$ gcc serverDict_TCP.c -o ser
serverDict_TCP.c: In function ‘main’:
serverDict_TCP.c:45:1: warning: implicit declaration of function ‘close’; did yo
u mean ‘pclose’? [-Wimplicit-function-declaration]
   45 | close(sersocket);
      | ^~~~~
      | pclose
alwin@debian:~/Downloads$ ./ser

server socket is created
bind success
server ready
accepted
Word to be searched : accord
alwin@debian:~/Downloads$
```

clientDict_TCP.c

```
#include<stdio.h>
#include<arpa/inet.h>
# define port 5000
void main()
{
struct sockaddr_in serveraddr;
int clisocket;
char buffer[100];
clisocket=socket(PF_INET,SOCK_STREAM,0);
if(clisocket>0)
printf("\nclient socket created");
serveraddr.sin_family= PF_INET;
serveraddr.sin_port= htons(port);
serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");
connect(clisocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));
printf("\nEnter word to be searched in dictionary : ");
scanf("%s",buffer);
send(clisocket,buffer,sizeof(buffer),0);
recv(clisocket,buffer,sizeof(buffer),0);
printf("Meaning - %s\n",buffer);
```

```
recv(clisocket,buffer,sizeof(buffer),0);
printf("Antonym - %s\n",buffer);
close(clisocket);
}
```



```
alwin@debian:~/Downloads$ gcc clientDict_TCP.c -o cli
clientDict_TCP.c: In function ‘main’:
clientDict_TCP.c:23:1: warning: implicit declaration of function ‘close’; did yo
u mean ‘pclose’? [-Wimplicit-function-declaration]
   23 | close(clisocket);
      | ^~~~~
      | pclose
alwin@debian:~/Downloads$ ./cli

client socket created
Enter word to be searched in dictionary : accord
Meaning - concurrence of opinion
Antonym - withhold
alwin@debian:~/Downloads$
```

## Implementation

Compile

```
Server : gcc serverDict_TCP.c
Client : gcc clientDict_TCP.c
```

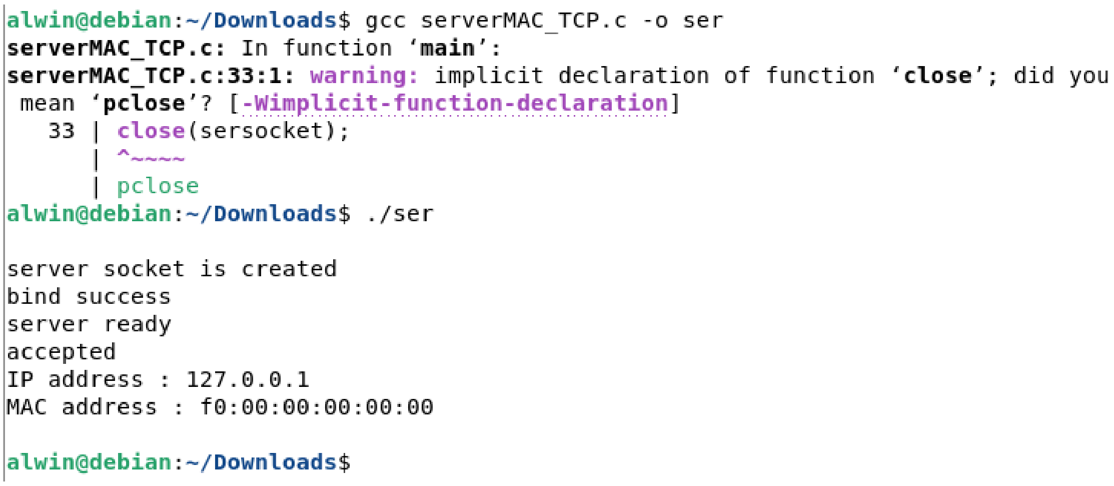
Run

```
Server : ./a.out
Client : ./a.out
```

Experiment-9: A simple TCP Server-Client program that gets the MAC address and IP address of the client connected.

serverMAC_TCP.c

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<arpa/inet.h>
# define port 5000
void main()
{
struct sockaddr_in serveraddr,newaddr;
int sersocket,newsocket,s,size;
char buffer[100];
sersocket=socket(PF_INET,SOCK_STREAM,0);
if(sersocket>0)
printf("\nserver socket is created");
serveraddr.sin_family= PF_INET;
serveraddr.sin_port= htons(port);
serveraddr.sin_addr.s_addr=htonl(INADDR_ANY);
s=bind(sersocket,(struct sockaddr *)&serveraddr,sizeof(serveraddr));
if(s==0)
printf("\nbind success");
listen(sersocket,1);
size=sizeof(newaddr);
printf("\nserver ready");
newsocket=accept(sersocket,(struct sockaddr *)&newaddr,&size);
if(newsocket>0)
printf("\naccepted");
recv(newsocket,buffer,1024,0);
printf("\nIP address : %s",buffer);
recv(newsocket,buffer,1024,0);
printf("\nMAC address : %s\n",buffer);
close(sersocket);
}
```

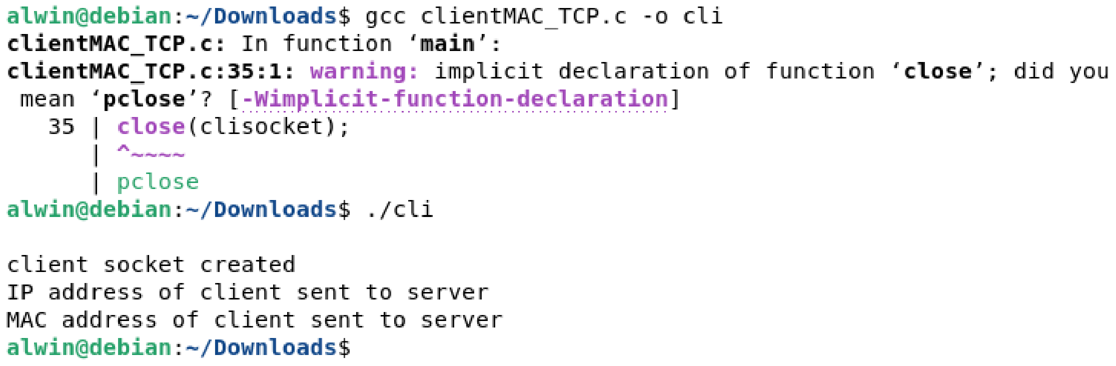


```
alwin@debian:~/Downloads$ gcc serverMAC_TCP.c -o ser
serverMAC_TCP.c: In function ‘main’:
serverMAC_TCP.c:33:1: warning: implicit declaration of function ‘close’; did you
 mean ‘pclose’? [-Wimplicit-function-declaration]
   33 | close(sersocket);
      | ^~~~~
      | pclose
alwin@debian:~/Downloads$ ./ser

server socket is created
bind success
server ready
accepted
IP address : 127.0.0.1
MAC address : f0:00:00:00:00:00

alwin@debian:~/Downloads$
```

clientMAC_TCP.c

```
#include<stdio.h>
#include<string.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include <sys/ioctl.h>
#include <net/if.h>
# define port 5000
void main()
{
```

```
struct sockaddr_in serveraddr;
struct ifreq ifr;
char *iface = "eth0";
char *mac;
int sersocket,clisocket;
char IPaddr[100],mac_addr[32]={0};
clisocket=socket(PF_INET,SOCK_STREAM,0);
if(clisocket>0)
printf("\nclient socket created");
serveraddr.sin_family= PF_INET;
serveraddr.sin_port= htons(port);
serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");
connect(clisocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));
strcpy(IPaddr,inet_ntoa(serveraddr.sin_addr));
send(clisocket,IPaddr,sizeof(IPaddr),0);
printf("\nIP address of client sent to server");
ifr.ifr_addr.sa_family = PF_INET;
strncpy((char *)ifr.ifr_name , (const char *)iface , IFNAMSIZ-1);
ioctl(clisocket, SIOCGIFHWADDR, &ifr);
mac = (char *)ifr.ifr_hwaddr.sa_data;
sprintf((char *)mac_addr,(const char
*)"%.2hhx:%.2hhx:%.2hhx:%.2hhx:%.2hhx:%.2hhx\n", mac[0], mac[1], mac[2],
mac[3], mac[4], mac[5]);
send(clisocket,mac_addr,sizeof(mac_addr),0);
printf("\nMAC address of client sent to server\n");
close(clisocket);
}
```



```
alwin@debian:~/Downloads$ gcc clientMAC_TCP.c -o cli
clientMAC_TCP.c: In function ‘main’:
clientMAC_TCP.c:35:1: warning: implicit declaration of function ‘close’; did you
 mean ‘pclose’? [-Wimplicit-function-declaration]
   35 | close(clisocket);
      | ^~~~~
      | pclose
alwin@debian:~/Downloads$ ./cli

client socket created
IP address of client sent to server
MAC address of client sent to server
alwin@debian:~/Downloads$
```

## Implementation

Compile

```

Server : gcc serverMAC_TCP.c -o ser

Client : gcc clientMAC_TCP.c -o cli

```

Run

```

Server : ./ser

Client : ./cli

```