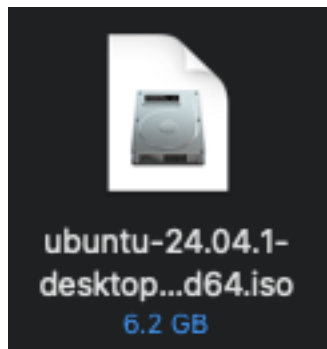


EE533 Lab1

Environment setup

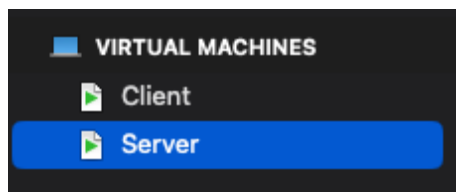
1. Download Ubuntu iso.



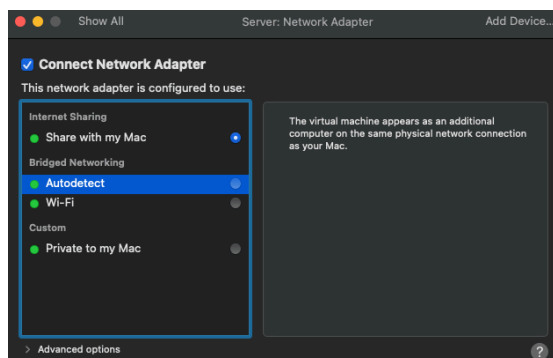
2. Download VMWare



3. Create Virtual machine in VMWare



4. Connecting network between Client and Server



Communication Between Server and Client with Sockets

1. Compile Server.c and get the executable file.

```
ubuntu@ubuntu:~/Downloads$ gcc Server.c -o server.c
```

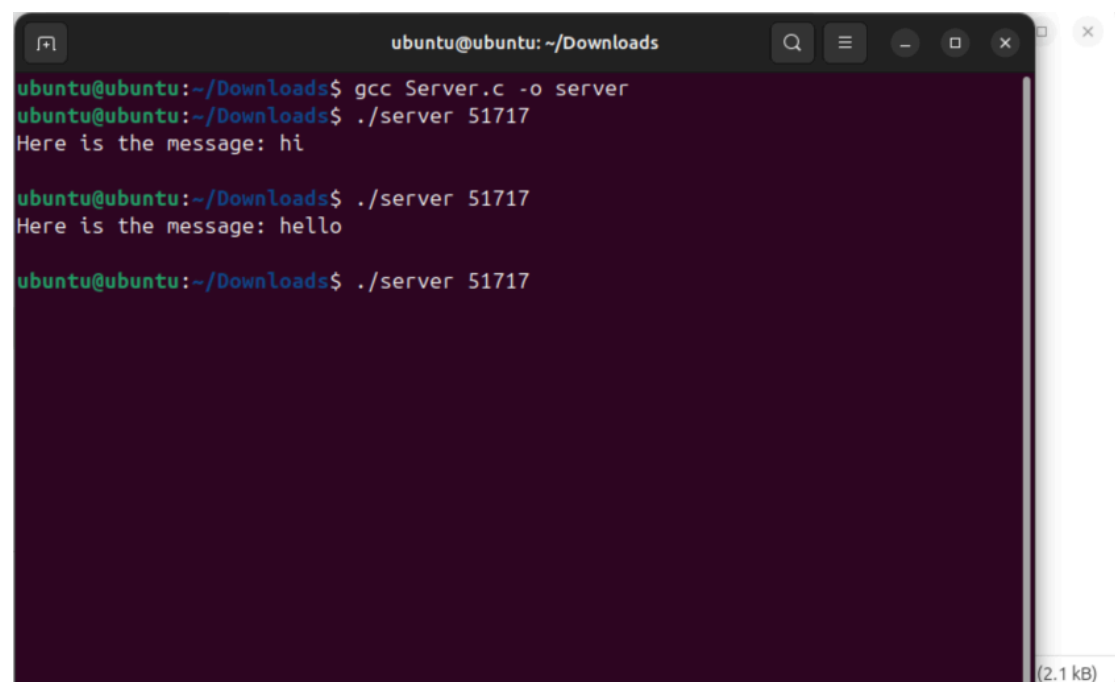
2. Get the ip address of Server.

```
ubuntu@ubuntu:~/Downloads$ ip addr  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue  
    link local :::::0:0:0:0:0:0:0:0 link-netnsid 0  
    inet 172.16.171.130/24 b
```

3. Compile Client.c and get the executable file.

```
ubuntu@ubuntu:~/Downloads$ gcc Client.c -o client.c
```

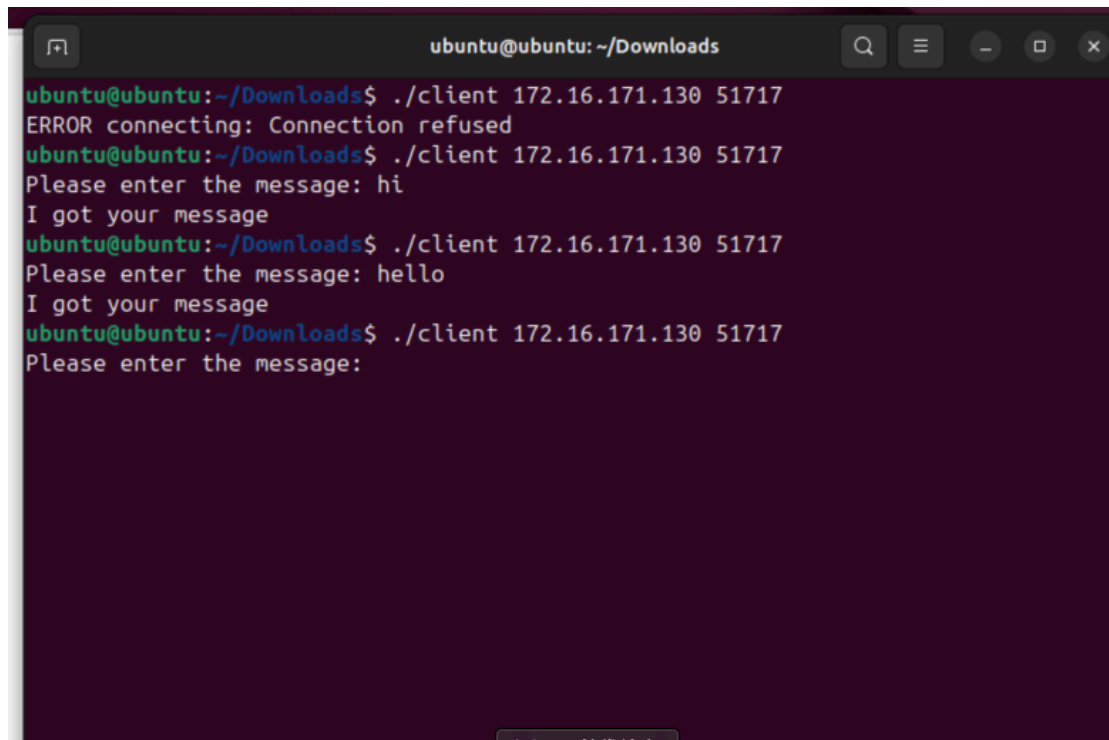
4. Server Waiting for Client.



```
ubuntu@ubuntu: ~/Downloads  
ubuntu@ubuntu:~/Downloads$ gcc Server.c -o server  
ubuntu@ubuntu:~/Downloads$ ./server 51717  
Here is the message: hi  
  
ubuntu@ubuntu:~/Downloads$ ./server 51717  
Here is the message: hello  
  
ubuntu@ubuntu:~/Downloads$ ./server 51717
```

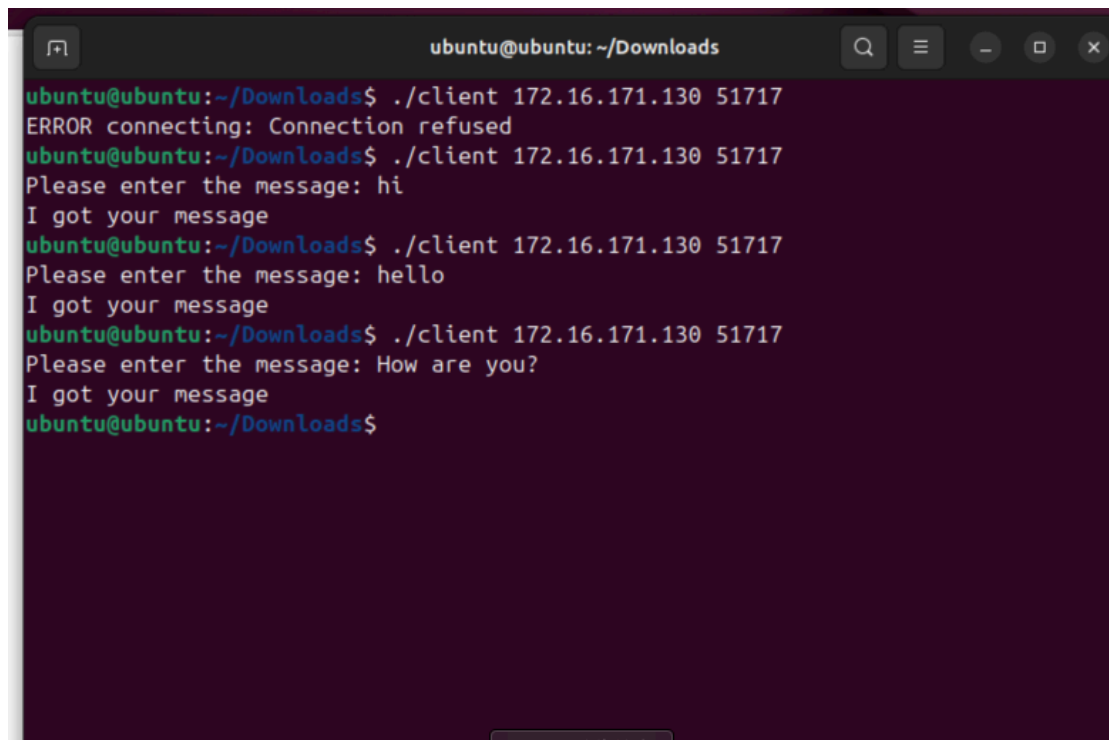
(2.1 kB)

5. Client connecting to Server.



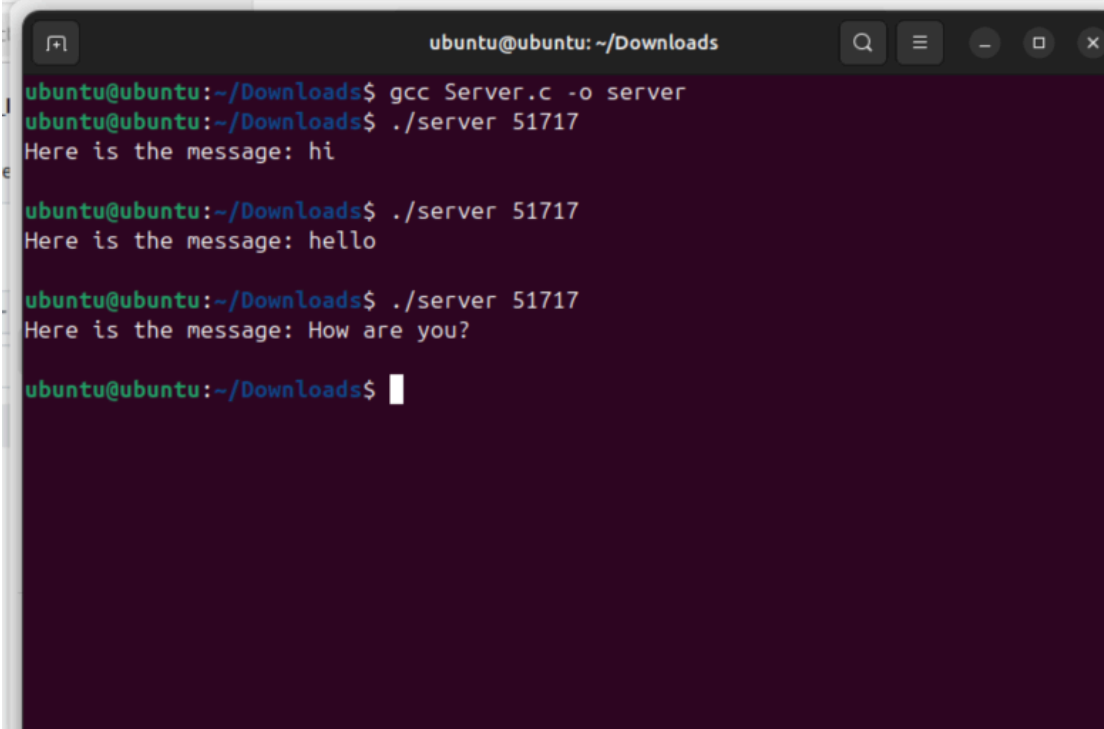
```
ubuntu@ubuntu: ~/Downloads
ubuntu@ubuntu:~/Downloads$ ./client 172.16.171.130 51717
ERROR connecting: Connection refused
ubuntu@ubuntu:~/Downloads$ ./client 172.16.171.130 51717
Please enter the message: hi
I got your message
ubuntu@ubuntu:~/Downloads$ ./client 172.16.171.130 51717
Please enter the message: hello
I got your message
ubuntu@ubuntu:~/Downloads$ ./client 172.16.171.130 51717
Please enter the message:
```

6. Client sending message: (How are you?)



```
ubuntu@ubuntu: ~/Downloads
ubuntu@ubuntu:~/Downloads$ ./client 172.16.171.130 51717
ERROR connecting: Connection refused
ubuntu@ubuntu:~/Downloads$ ./client 172.16.171.130 51717
Please enter the message: hi
I got your message
ubuntu@ubuntu:~/Downloads$ ./client 172.16.171.130 51717
Please enter the message: hello
I got your message
ubuntu@ubuntu:~/Downloads$ ./client 172.16.171.130 51717
Please enter the message: How are you?
I got your message
ubuntu@ubuntu:~/Downloads$
```

7. Server received message.

A terminal window titled 'ubuntu@ubuntu: ~/Downloads' with standard window controls. It shows the compilation of 'Server.c' into 'server' and three subsequent runs of './server 51717' which receive messages 'hi', 'hello', and 'How are you?' respectively. The prompt is currently at the fourth run.

```
ubuntu@ubuntu:~/Downloads$ gcc Server.c -o server
ubuntu@ubuntu:~/Downloads$ ./server 51717
Here is the message: hi

ubuntu@ubuntu:~/Downloads$ ./server 51717
Here is the message: hello

ubuntu@ubuntu:~/Downloads$ ./server 51717
Here is the message: How are you?

ubuntu@ubuntu:~/Downloads$
```

Enhancements to the server code

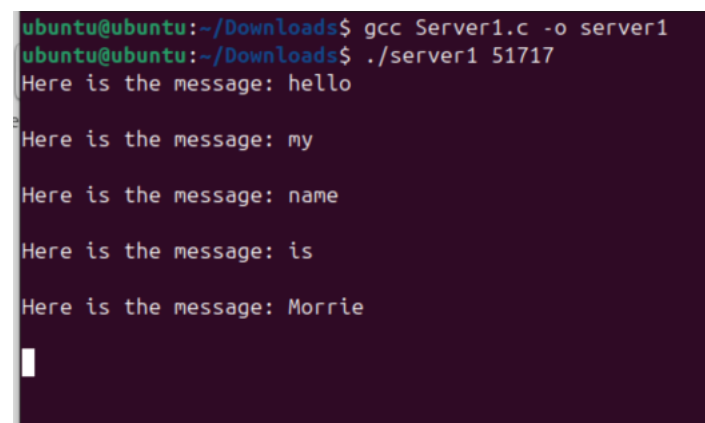
Allowing the server to handle multiple simultaneous connections.

Changes to the code:

```
while (1)
{
    newsockfd = accept(sockfd,
        (struct sockaddr *) &cli_addr, &clilen);
    if (newsockfd < 0)
        error("ERROR on accept");
    pid = fork();
    if (pid < 0)
        error("ERROR on fork");
    if (pid == 0)
    {
        close(sockfd);
        dostuff(newsockfd);
        exit(0);
    }
    else
        close(newsockfd);
} /* end of while */
```

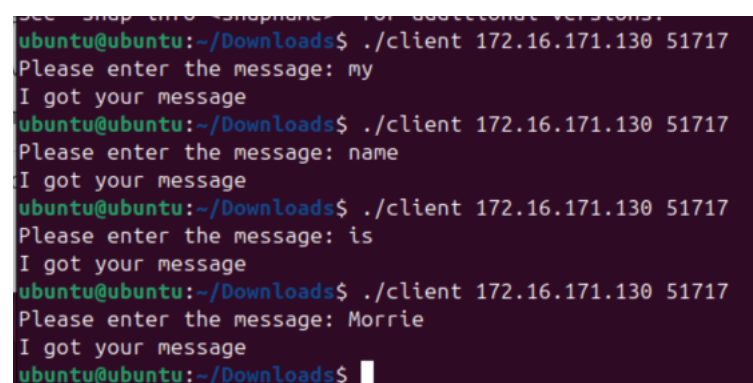
Result:

1. Real World Server (Server now can receive multiple message from the client)



```
ubuntu@ubuntu:~/Downloads$ gcc Server1.c -o server1
ubuntu@ubuntu:~/Downloads$ ./server1 51717
Here is the message: hello
Here is the message: my
Here is the message: name
Here is the message: is
Here is the message: Morrie
```

2. Client



```
ubuntu@ubuntu:~/Downloads$ ./client 172.16.171.130 51717
Please enter the message: my
I got your message
ubuntu@ubuntu:~/Downloads$ ./client 172.16.171.130 51717
Please enter the message: name
I got your message
ubuntu@ubuntu:~/Downloads$ ./client 172.16.171.130 51717
Please enter the message: is
I got your message
ubuntu@ubuntu:~/Downloads$ ./client 172.16.171.130 51717
Please enter the message: Morrie
I got your message
ubuntu@ubuntu:~/Downloads$
```

Dealing with the zombie problem

A zombie is a process which has terminated but cannot be permitted to fully die because at some point in the future, the parent of the process might execute a wait and would want information about the death of the child.

Changes to the code:

```
signal(SIGCHLD, SIG_IGN);
```

Result:

⇒ Now the zombie can be eliminated.

Datagram socket

Servers using datagram sockets do not use the listen() or the accept() system calls. After a socket has been bound to an address, the program calls recvfrom() to read a message.

```
ubuntu@ubuntu:~/Downloads$ gcc ClientUDP.c -o clientudp
ubuntu@ubuntu:~/Downloads$ ./clientudp 172.16.171.130 51717
Please enter the message: hi
Server reply: Got your message
```

```
ubuntu@ubuntu:~/Downloads$
```

```
ubuntu@ubuntu:~/Downloads$ gcc ServerUDP -o serverudp
/usr/bin/ld: cannot find ServerUDP: No such file or directory
collect2: error: ld returned 1 exit status
ubuntu@ubuntu:~/Downloads$ gcc ServerUDP.c -o serverudp
ubuntu@ubuntu:~/Downloads$ ./serverudp 51717
Received a datagram: hi
```

Changes to the code:

```
sock=socket(AF_INET, SOCK_DGRAM, 0);
```

Note that when the socket is created, the second argument is the symbolic constant SOCK_DGRAM instead of SOCK_STREAM. The protocol will be UDP, not TCP. ----

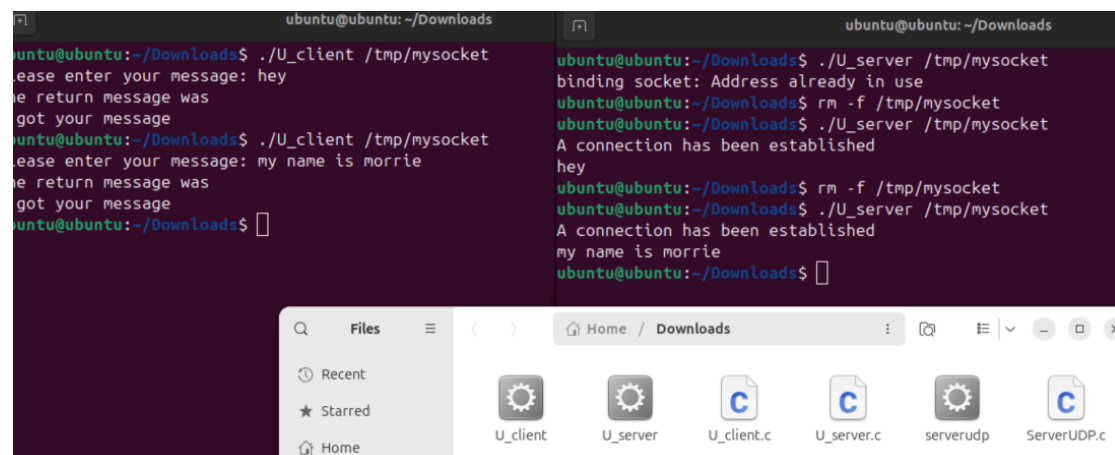
```
fromlen = sizeof(struct sockaddr_in);
while (1)
{
    n = recvfrom(sock,buf,1024,0,(struct sockaddr *)&from,&fromlen);
    if (n < 0) error("recvfrom");
```

Sockets in the Unix Domain

The only difference between a socket in the Unix domain and a socket in the Internet domain is the form of the address.

```
struct sockaddr_un
{
    short  sun_family; /* AF_UNIX */
    char   sun_path[108]; /* path name (gag) */
};
```

The field `sun_path` has the form of a path name in the Unix file system. This means that both client and server have to be running the same file system.



The image shows two terminal windows and a file manager. The left terminal window shows the execution of `U_client` with the following output:

```
ubuntu@ubuntu: ~/Downloads
ubuntu@ubuntu:~/Downloads$ ./U_client /tmp/mysocket
Please enter your message: hey
The return message was
I got your message
ubuntu@ubuntu:~/Downloads$ ./U_client /tmp/mysocket
Please enter your message: my name is morrie
The return message was
I got your message
ubuntu@ubuntu:~/Downloads$
```

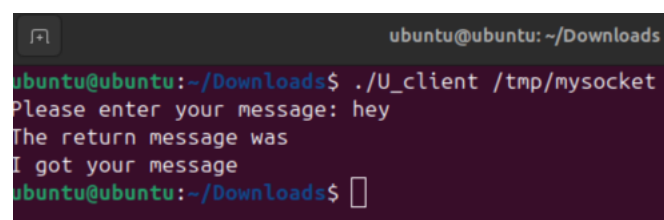
The right terminal window shows the execution of `U_server` with the following output:

```
ubuntu@ubuntu:~/Downloads$ ./U_server /tmp/mysocket
binding socket: Address already in use
ubuntu@ubuntu:~/Downloads$ rm -f /tmp/mysocket
ubuntu@ubuntu:~/Downloads$ ./U_server /tmp/mysocket
A connection has been established
hey
ubuntu@ubuntu:~/Downloads$ rm -f /tmp/mysocket
ubuntu@ubuntu:~/Downloads$ ./U_server /tmp/mysocket
A connection has been established
my name is morrie
ubuntu@ubuntu:~/Downloads$
```

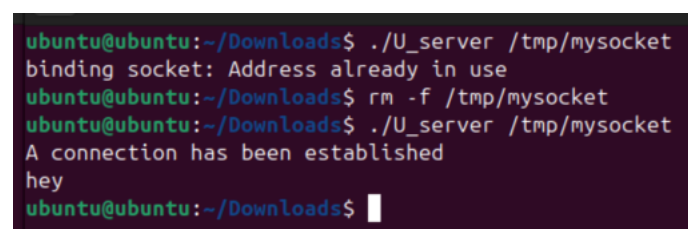
The file manager shows the following files in the `Downloads` directory:

- `U_client` (executable icon)
- `U_server` (executable icon)
- `U_client.c` (text file icon)
- `U_server.c` (text file icon)
- `serverudp` (executable icon)
- `ServerUDP.c` (text file icon)

Result:



```
ubuntu@ubuntu:~/Downloads$ ./U_client /tmp/mysocket
Please enter your message: hey
The return message was
I got your message
ubuntu@ubuntu:~/Downloads$
```



```
ubuntu@ubuntu:~/Downloads$ ./U_server /tmp/mysocket
binding socket: Address already in use
ubuntu@ubuntu:~/Downloads$ rm -f /tmp/mysocket
ubuntu@ubuntu:~/Downloads$ ./U_server /tmp/mysocket
A connection has been established
hey
ubuntu@ubuntu:~/Downloads$
```