

Project0

통신네트워크

■ 기본적인 TCP, UDP 소켓 프로그래밍 코드를 분석하고 실행해보기

- Client
 - src.jpg 파일을 4096 Byte 단위로 나누어 Server로 전송
- Server
 - Client로부터 수신 받은 파일들을 합쳐 dst.jpg 생성

■ 제출 관련

- 제출해야 할 것
 - UDP: 터미널 결과 화면(2장) & dst.jpg 사진 화면 캡처 후 pdf에 첨부
 - Client가 Server port번호를 잘못 입력했을 때 1장
 - Client가 Server port번호를 제대로 입력했을 때 1장
 - TCP: 터미널 결과 화면 (2장) & dst.jpg 사진 화면 캡처 후 pdf에 첨부
 - Client가 Server port번호를 잘못 입력했을 때 1장
 - Client가 Server port번호를 제대로 입력했을 때 1장
 - 결과 화면을 기반으로 알 수 있는 UDP와 TCP의 차이점을 간략하게 서술할 것
- 제출 기한
 - 10/12 (수) 23시 59분까지
- 제출 파일명
 - Project0_학번_이름.pdf

참고

1. 주어진 TCP, UDP 코드는 가장 기본적인 동작들만 구현된 코드로 많은 기능들이 제외되어 있습니다.
2. 코드 분석 결과에 대해선 따로 제출하지 않아도 되지만, 코드 분석을 직접 해보시면 Project1을 할 때 도움이 많이 되니 해보시기 바랍니다.

■ UDP/IP

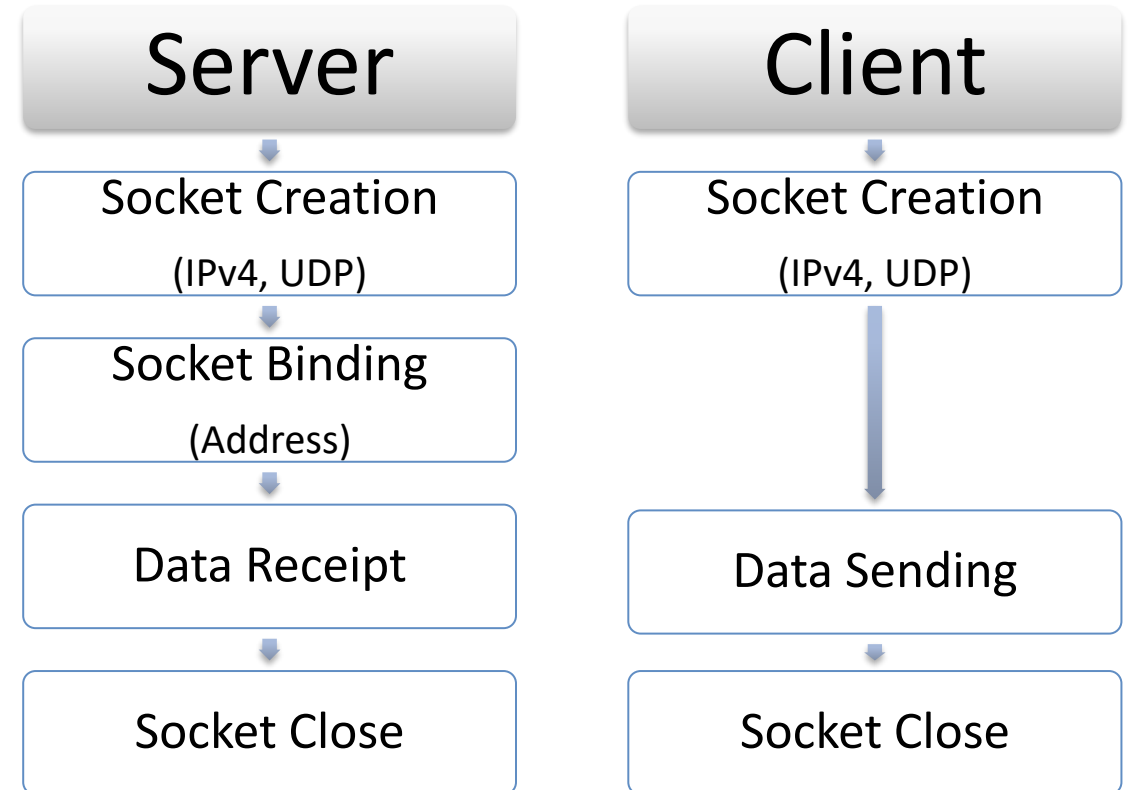
– Server 사용 함수

- socket
- bind
- recvfrom
- close

– Client 사용 함수

- socket
- sendto
- close

– 상세 함수 정보는 터미널 창에서 man 명령어 이용



■ TDP/IP

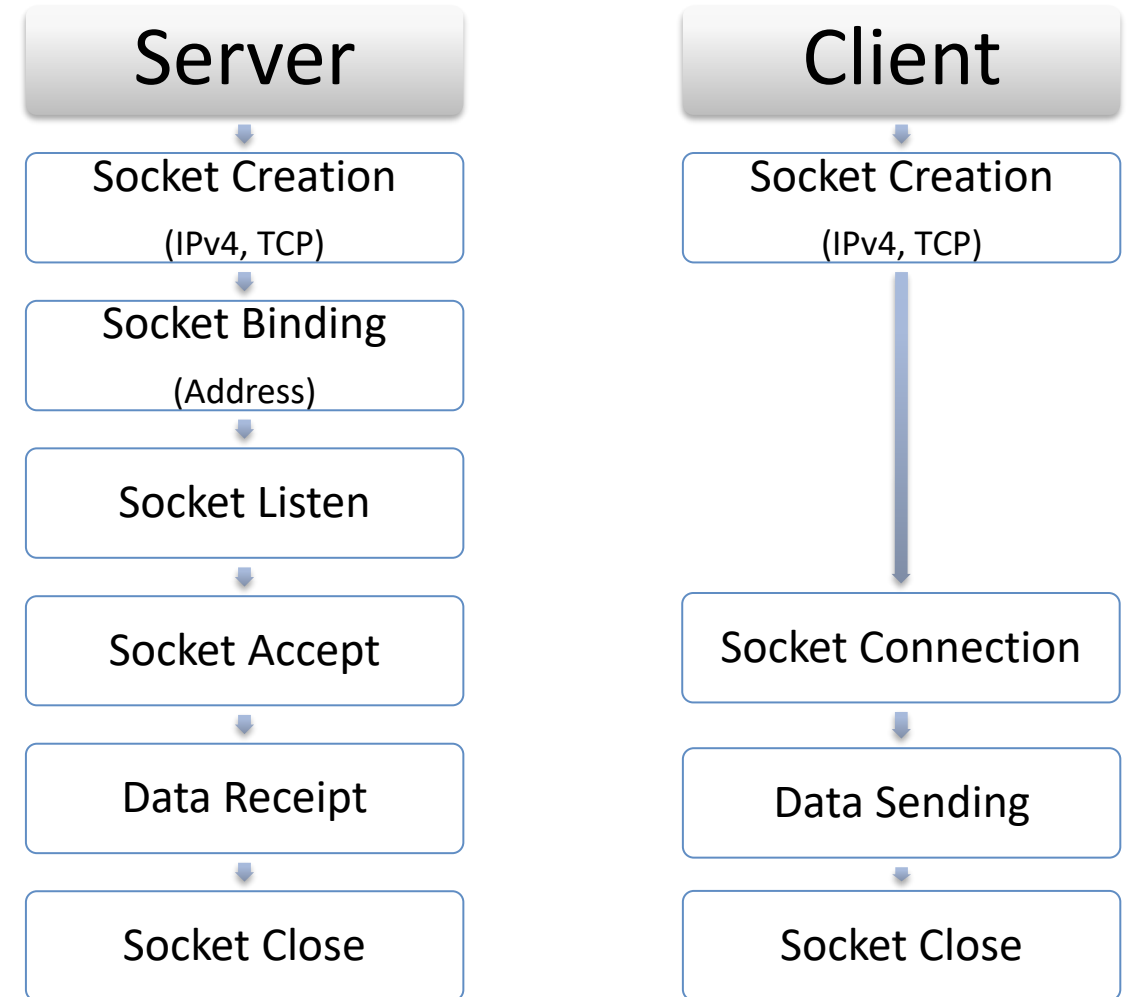
– Server 사용 함수

- socket
- bind
- listen
- accept
- read
- close

– Client 사용 함수

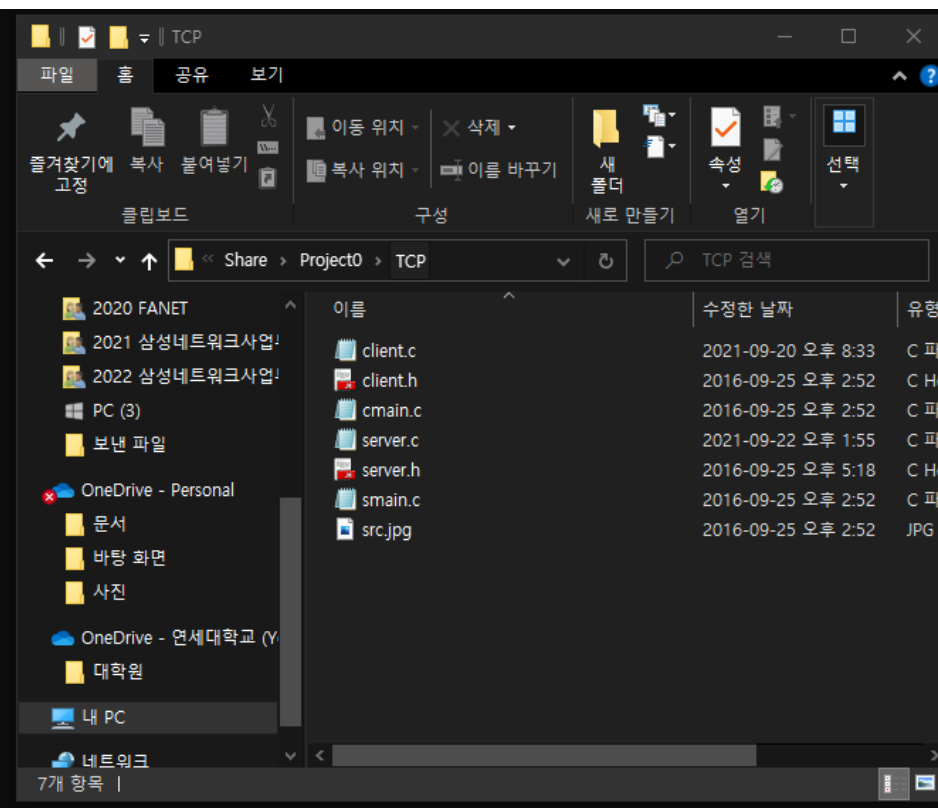
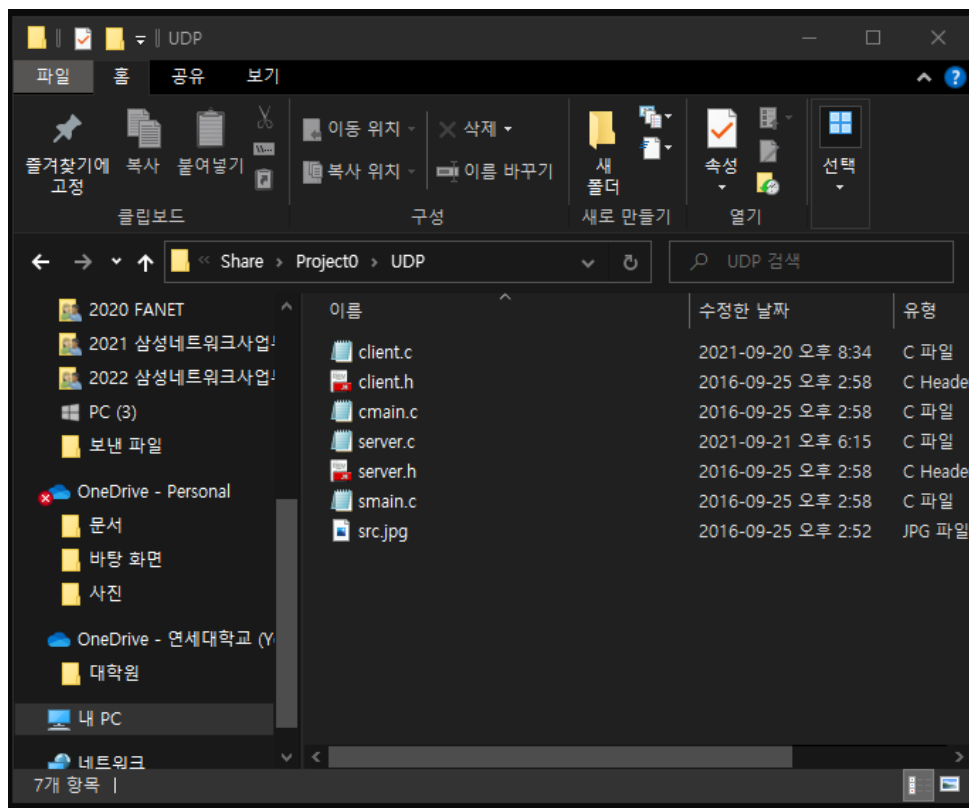
- socket
- connect
- write
- close

– 상세 함수 정보는 터미널 창에서 man 명령어 이용



■ 공유 폴더에 파일 추가

- 내 OS에서 virtual box와 공유해둔 폴더에 다운로드 받은 파일들 추가
 - UDP, TCP 폴더로 나누어 추가할 것



■ 컴파일 및 실행

- 터미널 창에 `sudo apt-get install gcc` 입력
 - gcc 컴파일러 다운로드

```
myubuntu@myubuntu:~$ sudo apt-get install gcc
[sudo] password for myubuntu:
Sorry, try again.
[sudo] password for myubuntu:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu gcc-9 libasan5 libatomic1
  libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0
  libgcc-9-dev libitm1 liblsan0 libquadmath0 libtsan0 libubsan1 linux-libc-dev
  manpages-dev
Suggested packages:
  binutils-doc gcc-multilib make autoconf automake libtool flex bison gcc-doc
  gcc-9-multilib gcc-9-doc gcc-9-locales glibc-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu gcc gcc-9 libasan5
  libatomic1 libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0
  libctf0 libgcc-9-dev libitm1 liblsan0 libquadmath0 libtsan0 libubsan1
  linux-libc-dev manpages-dev
0 upgraded, 21 newly installed, 0 to remove and 51 not upgraded.
Need to get 23.3 MB of archives.
After this operation, 107 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://kr.archive.ubuntu.com/ubuntu focal-updates/main amd64 binutils-comm
on amd64 2.34-6ubuntu1.1 [207 kB]
Get:2 http://kr.archive.ubuntu.com/ubuntu focal-updates/main amd64 libbinutils
```

■ 컴파일 및 실행 (UDP 기준)

- 터미널 2개 실행 후 각각 공유 폴더내의 UDP 파일들 저장 위치로 이동
 - cd 활용

```
myubuntu@myubuntu: ~/my_share/Project0/UDP
myubuntu@myubuntu:~$ cd my_share
myubuntu@myubuntu:~/my_share$ ls
Project0 Project1 Project2 Project3
myubuntu@myubuntu:~/my_share$ cd Project0
myubuntu@myubuntu:~/my_share/Project0$ ls
TCP UDP
myubuntu@myubuntu:~/my_share/Project0$ cd UDP
myubuntu@myubuntu:~/my_share/Project0/UDP$
```

```
myubuntu@myubuntu: ~/my_share/Project0/UDP
myubuntu@myubuntu:~$ cd my_share
myubuntu@myubuntu:~/my_share$ cd Project0
myubuntu@myubuntu:~/my_share/Project0$ ls
TCP UDP
myubuntu@myubuntu:~/my_share/Project0$ cd UDP
myubuntu@myubuntu:~/my_share/Project0/UDP$ ls
client.c client.h cmain.c server.c server.h smain.c
myubuntu@myubuntu:~/my_share/Project0/UDP$
```

Project0

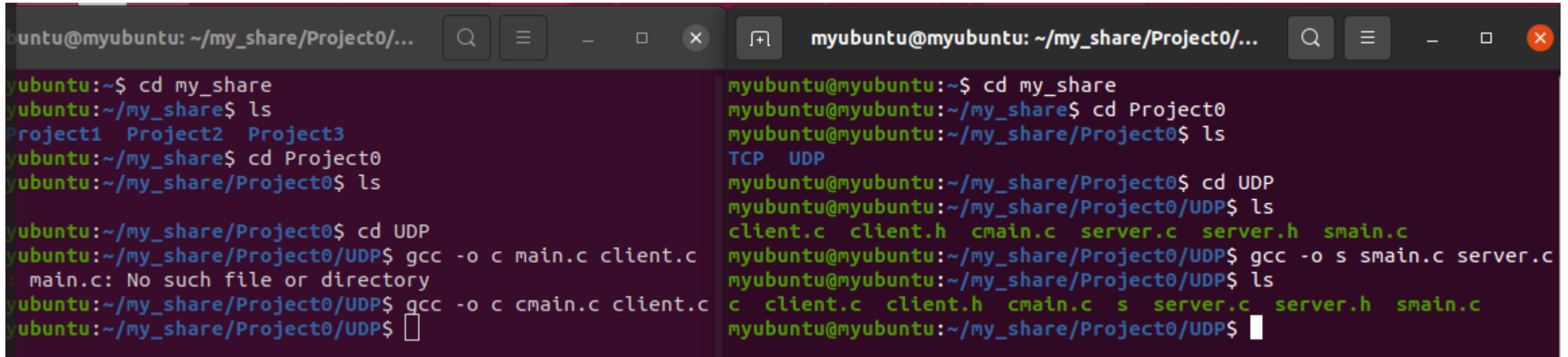
■ 컴파일 및 실행 (UDP 기준)

– 터미널 1

- gcc -o s smain.c server.c 입력
 - Server 소스파일(smain.c, server.c) 컴파일해서 s라는 이름의 실행파일 만들기

– 터미널 2

- gcc -o c cmain.c client.c 입력
 - Client 소스파일(cmain.c, client.c) 컴파일해서 c라는 이름의 실행파일 만들기



```
myubuntu@myubuntu: ~/my_share/Project0/...  
myubuntu:~$ cd my_share  
myubuntu:~/my_share$ ls  
Project1 Project2 Project3  
myubuntu:~/my_share$ cd Project0  
myubuntu:~/my_share/Project0$ ls  
TCP UDP  
myubuntu:~/my_share/Project0$ cd UDP  
myubuntu:~/my_share/Project0/UDP$ ls  
client.c client.h cmain.c server.c server.h smain.c  
myubuntu:~/my_share/Project0/UDP$ gcc -o s smain.c server.c  
myubuntu:~/my_share/Project0/UDP$ ls  
c client.c client.h cmain.c s server.c server.h smain.c  
myubuntu:~/my_share/Project0/UDP$  
  
myubuntu@myubuntu: ~/my_share/Project0/...  
myubuntu:~$ cd my_share  
myubuntu:~/my_share$ cd Project0  
myubuntu:~/my_share/Project0$ ls  
TCP UDP  
myubuntu:~/my_share/Project0$ cd UDP  
myubuntu:~/my_share/Project0/UDP$ ls  
client.c client.h cmain.c server.c server.h smain.c  
myubuntu:~/my_share/Project0/UDP$ gcc -o c main.c client.c  
main.c: No such file or directory  
myubuntu:~/my_share/Project0/UDP$ gcc -o c cmain.c client.c  
myubuntu:~/my_share/Project0/UDP$
```


■ 컴파일 및 실행 (UDP 기준)

– 터미널 1

- ./s 99999 입력
 - ./실행파일명 Port 번호(임의의 번호 가능)
 - Server side에서는 specific한 port number를 할당한다.

```
myubuntu@myubuntu:~/my_share/Project0/UDP$ ./s 99999
```

– 터미널 2

- ./c 127.0.0.1 99999 입력
 - ./실행파일명 Server의IP Server의Port 번호
 - 127.0.0.1: localhost
 - » OS에서 자기 자신에게 접근하기 위한 IP
 - » 현재 같은 Host에서 Server와 Client가 동시에 구현되어 있으므로 사용 가능
 - 참고: localhost 대신 실제 ip주소를 이용하여 실행시키는 방법
 - » sudo apt-get install ifconfig 로 ifconfig 다운로드
 - » ifconfig 실행하여 ip주소 알아낸 뒤 그 ip주소를 대신 입력

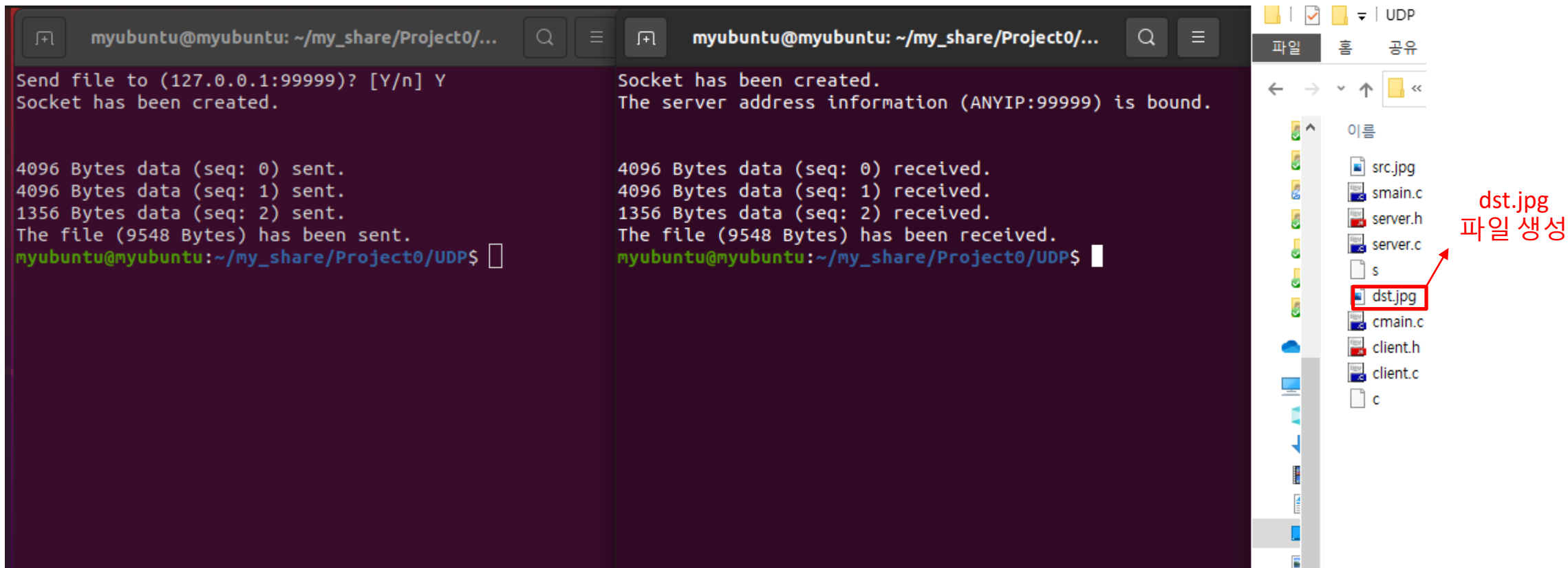
```
myubuntu@myubuntu:~/my_share/Project0/UDP$ ./c 127.0.0.1 99999
```

```
myubuntu@myubuntu: ~  
myubuntu@myubuntu:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.0.171 netmask 255.255.255.0 broadcast 192.168.0.255  
    inet6 fe80::dc4f:1567:9fa8:c7ec prefixlen 64  
    ether 08:00:27:e2:38:80 txqueuelen 1000 (Ethernet)  
    RX packets 141261 bytes 29599392 (29.5 MB)  
    RX errors 0 dropped 2185 overruns 0 frame 0  
    TX packets 2058 bytes 438117 (438.1 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0  
    collisions 0  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0
```

Project0

■ UDP 실행 결과 (캡처 위해 작은 용량의 사진파일 전송)

- Port 번호 제대로 입력했을 때



The image shows a terminal window with two panes. The left pane shows the output of a file transfer from a client to a server. The right pane shows the output of a server receiving a file. To the right of the terminal is a file explorer window showing the contents of the directory `~/my_share/Project0/UDP`. The file `dst.jpg` is highlighted with a red box, and a red arrow points to it with the text `dst.jpg 파일 생성` (dst.jpg file created).

```
myubuntu@myubuntu: ~/my_share/Project0/...  
Send file to (127.0.0.1:99999)? [Y/n] Y  
Socket has been created.  
  
4096 Bytes data (seq: 0) sent.  
4096 Bytes data (seq: 1) sent.  
1356 Bytes data (seq: 2) sent.  
The file (9548 Bytes) has been sent.  
myubuntu@myubuntu:~/my_share/Project0/UDP$
```

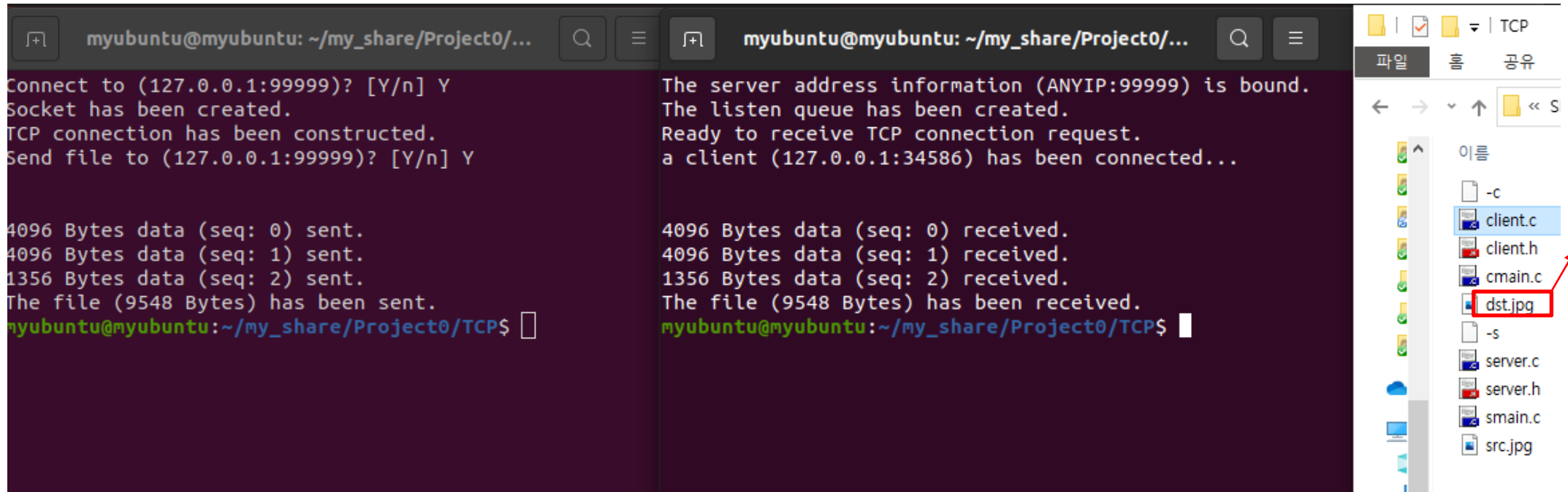
```
myubuntu@myubuntu: ~/my_share/Project0/...  
Socket has been created.  
The server address information (ANYIP:99999) is bound.  
  
4096 Bytes data (seq: 0) received.  
4096 Bytes data (seq: 1) received.  
1356 Bytes data (seq: 2) received.  
The file (9548 Bytes) has been received.  
myubuntu@myubuntu:~/my_share/Project0/UDP$
```

File Explorer: `~/my_share/Project0/UDP`
Files: `src.jpg`, `smain.c`, `server.h`, `server.c`, `s`, `dst.jpg` (highlighted), `cmain.c`, `client.h`, `client.c`, `c`

dst.jpg 파일 생성

Project0

- TCP 실행 결과 (캡처 위해 작은 용량의 사진파일 전송)
 - Port 번호 제대로 입력했을 때



The image shows a terminal window and a file explorer side-by-side. The terminal window is split into two panes, both showing the same text. The left pane shows the server's perspective: it prompts to connect to 127.0.0.1:99999, confirms the connection, and then shows data being sent in three chunks (4096, 4096, and 1356 bytes) and the full 9548-byte file being sent. The right pane shows the client's perspective: it confirms the server is bound and listening, shows a client connecting from 127.0.0.1:34586, and shows the data being received in three chunks and the full 9548-byte file being received. To the right of the terminal is a file explorer window titled 'TCP'. It shows a directory listing with files: -c, client.c, client.h, cmain.c, dst.jpg, -s, server.c, server.h, smain.c, and src.jpg. The file 'dst.jpg' is highlighted with a red box, and a red arrow points to it from the text 'dst.jpg 파일 생성' (dst.jpg file creation).

```
myubuntu@myubuntu: ~/my_share/Project0/...  
Connect to (127.0.0.1:99999)? [Y/n] Y  
Socket has been created.  
TCP connection has been constructed.  
Send file to (127.0.0.1:99999)? [Y/n] Y  
  
4096 Bytes data (seq: 0) sent.  
4096 Bytes data (seq: 1) sent.  
1356 Bytes data (seq: 2) sent.  
The file (9548 Bytes) has been sent.  
myubuntu@myubuntu:~/my_share/Project0/TCP$  
  
myubuntu@myubuntu: ~/my_share/Project0/...  
The server address information (ANYIP:99999) is bound.  
The listen queue has been created.  
Ready to receive TCP connection request.  
a client (127.0.0.1:34586) has been connected...  
  
4096 Bytes data (seq: 0) received.  
4096 Bytes data (seq: 1) received.  
1356 Bytes data (seq: 2) received.  
The file (9548 Bytes) has been received.  
myubuntu@myubuntu:~/my_share/Project0/TCP$
```

dst.jpg
파일 생성

■ 참고: src.jpg



- 참고: 가상머신 2대를 이용하여 서로 다른 Host간 TCP 사진 전송
 - (실제로 해보시려면 [머신] - [설정] - [네트워크] 에서 NAT를 어댑터에 브리지로 변경하시면 됩니다.)

```
ubuntu@ubuntu-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.98 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::bf62:a500:74d0:6b58 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:79:f6:ae txqueuelen 1000 (Ethernet)
    RX packets 9026 bytes 2378199 (2.3 MB)
    RX errors 0 dropped 141 overruns 0 frame 0
    TX packets 601 bytes 203482 (203.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1268 bytes 103776 (103.7 KB)

ubuntu@ubuntu-VirtualBox:~/pj1/TCP_Final$ nc -l 1118
The server address information (ANYIP:1118) is bound.
The listen queue has been created.
Ready to receive TCP connection request.
a client (192.168.0.171:33654) has been connected...

4096 Bytes data (seq: 0) received.
4096 Bytes data (seq: 1) received.
1356 Bytes data (seq: 2) received.
The file (9548 Bytes) has been received.
ubuntu@ubuntu-VirtualBox:~/pj1/TCP_Final$
```

```
myubuntu@myubuntu:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.171 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::dc4f:1567:9fa8:c7ec prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:e2:38:80 txqueuelen 1000 (Ethernet)
    RX packets 247789 bytes 51716326 (51.7 MB)
    RX errors 0 dropped 3881 overruns 0 frame 0
    TX packets 3100 bytes 590105 (590.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)

myubuntu@myubuntu:~/my_share/Project0$ nc 192.168.0.98 1118
Connect to (192.168.0.98:1118)? [Y/n] Y
Socket has been created.
TCP connection has been constructed.
Send file to (192.168.0.98:1118)? [Y/n] Y

4096 Bytes data (seq: 0) sent.
4096 Bytes data (seq: 1) sent.
1356 Bytes data (seq: 2) sent.
The file (9548 Bytes) has been sent.
myubuntu@myubuntu:~/my_share/Project0/TCP$
```

끝