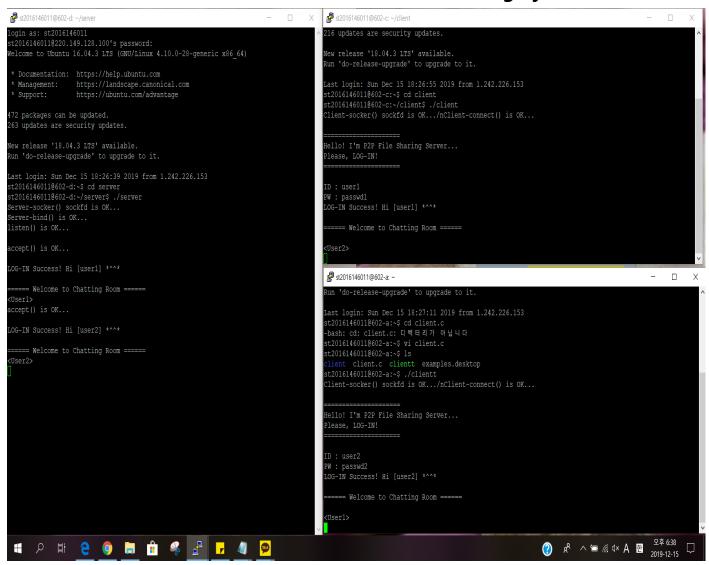
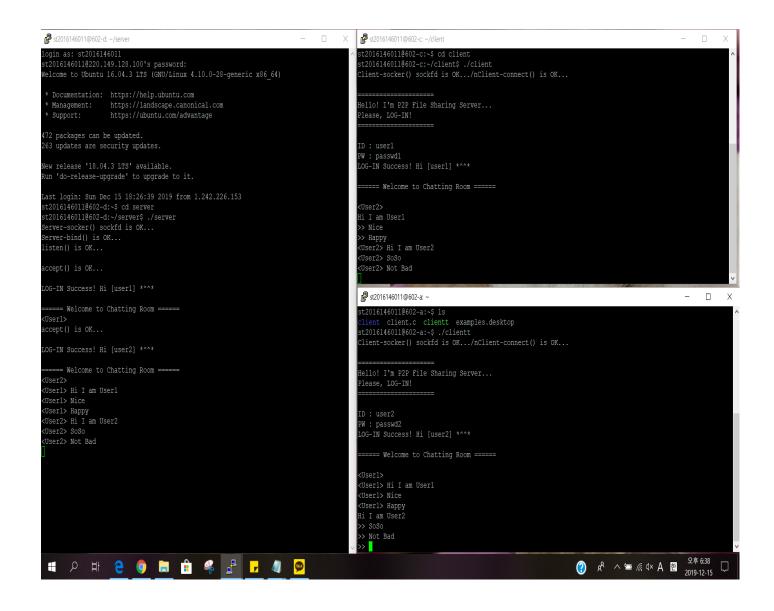
임베디드운영체제 P2P File Transfer & Chatting System 결과



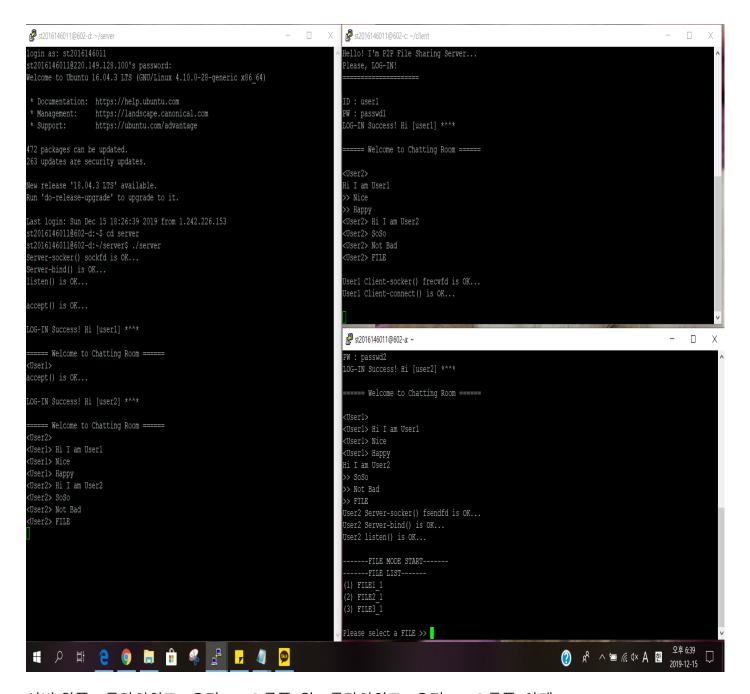
서버(왼쪽), 클라이언트1(오른쪽위-101), 클라이언트2(오른쪽아래-102)

두 유저 로그인 완료

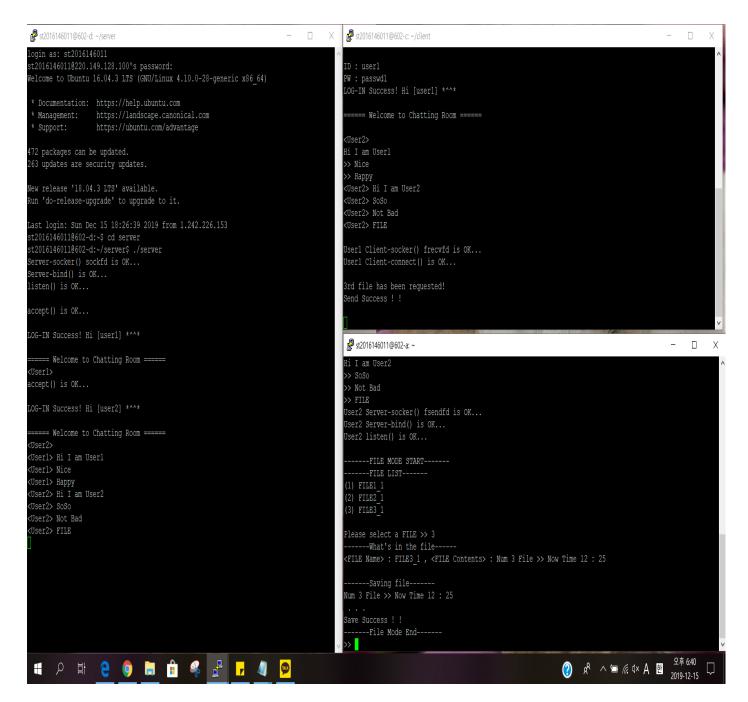


서버(왼쪽), 클라이언트1(오른쪽위-101), 클라이언트2(오른쪽아래-102)

클라이언트1이 보낸 메시지가 클라이언트2에도 잘 뜨고, 클라이언트2가 보낸 메시지가 클라이언트1에도 잘 뜨며 서버 쪽에도 잘 뜨는 것을 확인



채팅 중 유저2가 FILE을 요청하여 두 클라이언트는 연결되고 유저1이 보낸 파일(FILE1_1,FILE2_1,FILE3_1) 리스트를 출력하여 원하는 파일을 선택하도록 하는 실행화면



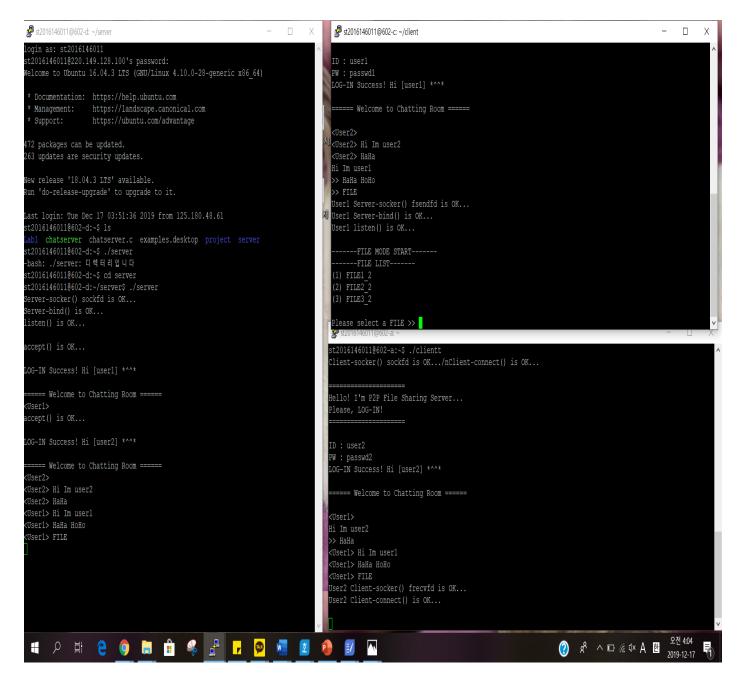
유저2가 파일을 선택(3번째 파일)하면 유저1에는 3번째파일을 요청받았음을 알리는 문구가 출력되고 Send Success!!문구와 함께 유저2에는 파일이름과 내용을 출력, 같은 내용/이름으로 파일을 저장한다.

```
st2016146011@602-a:~$ ls
client client.c clientt examples.desktop
```

유저2가 파일을 전송 받기 전

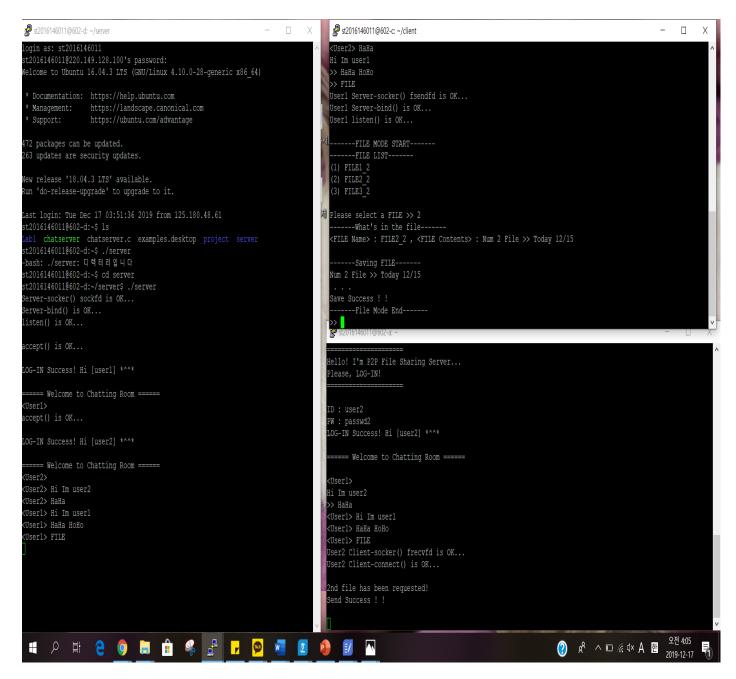
```
st2016146011@602-a:~$ ls
FILE3_1 client client.c clientt examples.desktop
```

유저2가 파일을 전송 받은 후



같은 방법으로 유저1이 유저2에게 요청

채팅 중 유저1이 FILE을 요청하여 두 클라이언트는 연결되고 유저2가 보낸 파일(FILE1_2,FILE2_2,FILE3_2) 리스트를 출력하여 원하는 파일을 선택하도록 하는 실행화면

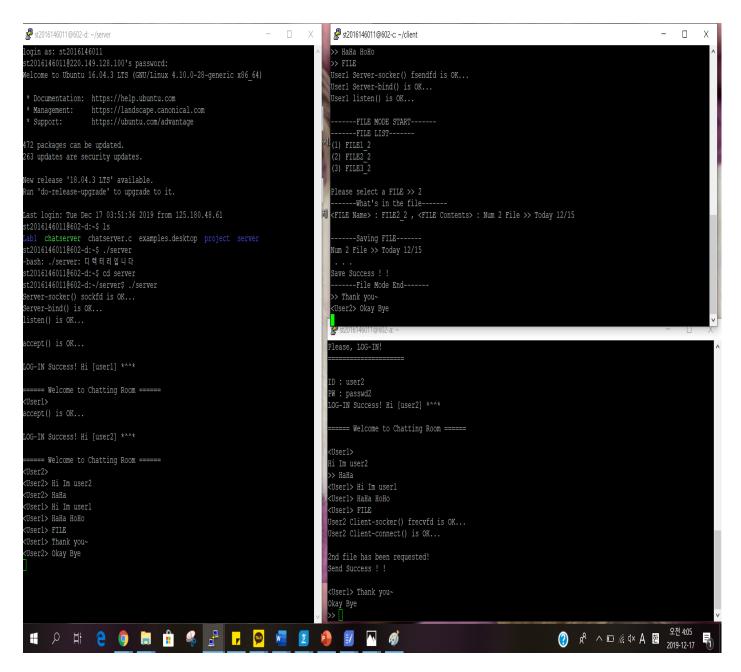


유저1이 파일을 선택(2번째 파일)하면 유저2에는 2번째파일을 요청받았음을 알리는 문구가 출력되고 Send Success!!문구와 함께 유저1에는 파일이름과 내용을 출력, 같은 내용/이름으로 파일을 저장한다.

st2016146011@602-c:~/client\$ ls client client.c

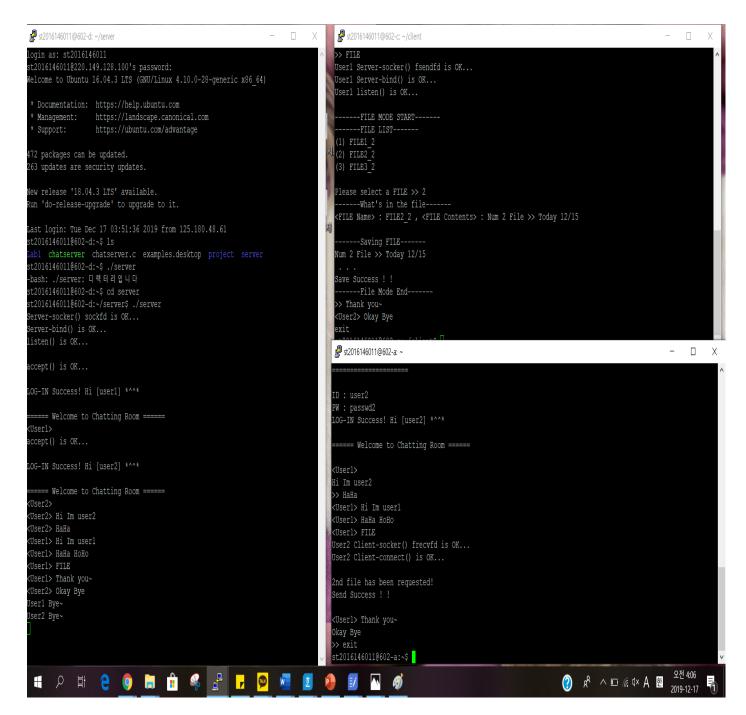
유저1 파일전송 받기 전

st2016146011@602-c:~/client\$ ls FILE2_2 client client.c



파일전송이 끝나면 다시 채팅모드로 들어가는 것을 확인할 수 있다.

서버 쪽에도 메시지가 도착확인



두 클라이언트 모두 exit라는 메시지를 통해 프로그램을 종료함을 알 수 있음