

# 컴퓨터네트워크

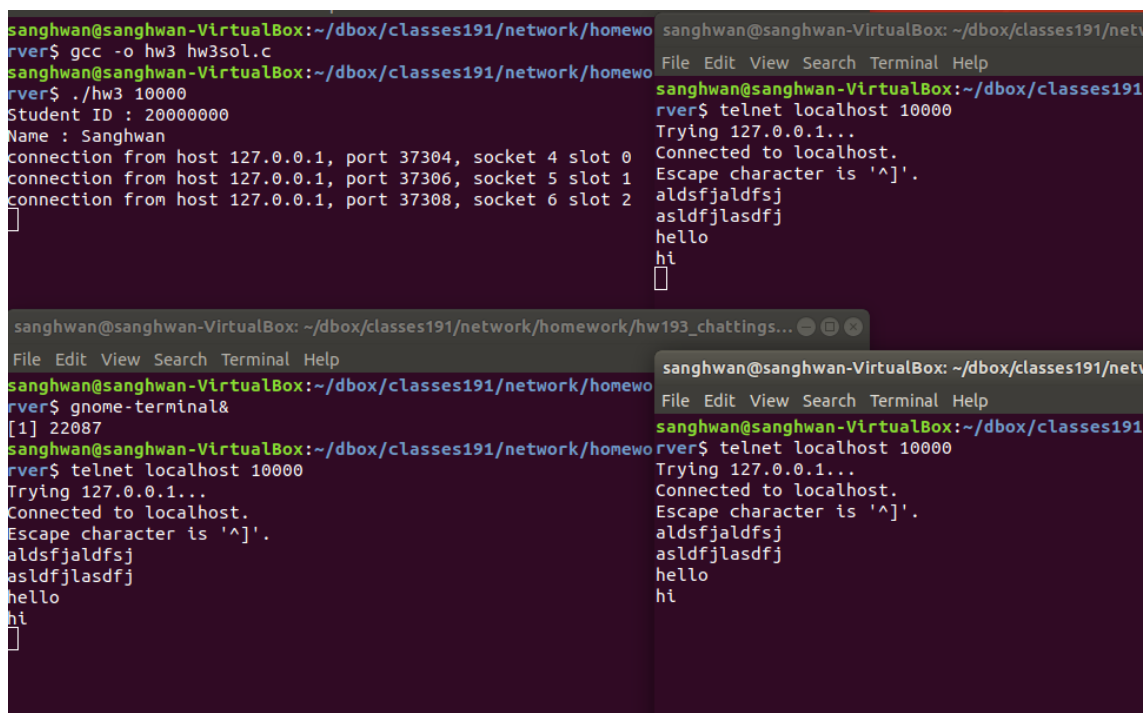
## HW #3 (30 Points)

Due date : 2019/5/20 월요일 (eCampus)

제출물 : studentid.c, studentid.py File (모든 코드를 이 하나의 파일에서 구현하면 됨.)

이 과제는 단체 채팅 서버를 구현하는 내용이다. 채팅 클라이언트는 telnet으로 대신한다.  
아래는 실행 예이다.

하나 받으면 다른 클라이언트한테도 보내줌



```
sanghwan@sanghwan-VirtualBox:~/dbox/classes191/network/homework
rver$ gcc -o hw3 hw3sol.c
sanghwan@sanghwan-VirtualBox:~/dbox/classes191/network/homework
rver$ ./hw3 10000
Student ID : 20000000
Name : Sanghwan
connection from host 127.0.0.1, port 37304, socket 4 slot 0
connection from host 127.0.0.1, port 37306, socket 5 slot 1
connection from host 127.0.0.1, port 37308, socket 6 slot 2
[1] 22087
rver$ telnet localhost 10000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
aldfjaldfsj
asldfjlasdfj
hello
hi
[1] 22087
rver$ telnet localhost 10000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
aldfjaldfsj
asldfjlasdfj
hello
hi
[1] 22087
rver$ telnet localhost 10000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
aldfjaldfsj
asldfjlasdfj
hello
hi
```

이 서버는 tcp 포트 번호를 command line argument로 입력 받아서 시작한다. 즉 이 응용을 실행할 때 아래와 같이 1개의 argument를 입력 받는다.

\$ hw3 tcpport

- tcpport : 클라이언트로부터 들어오는 연결 요청을 받기 위한 TCP port 번호

위 그림에서 3명의 클라이언트에서 “telnet localhost 10000” 이라는 명령어로 채팅서버에 접속을 하였다. 같은 컴퓨터에서 실행중이기 때문에 localhost 라는 hostname을 사용한 것이다. 서버의 포트번호로 10000을 사용한다.

클라이언트에서 서버로 연결요청이 들어오면 간단하게 클라이언트의 IP와 포트 번호 소켓

descriptor 등을 출력한다. **마지막의 slot은 디버깅용이므로 출력하지 않아도 됨.**

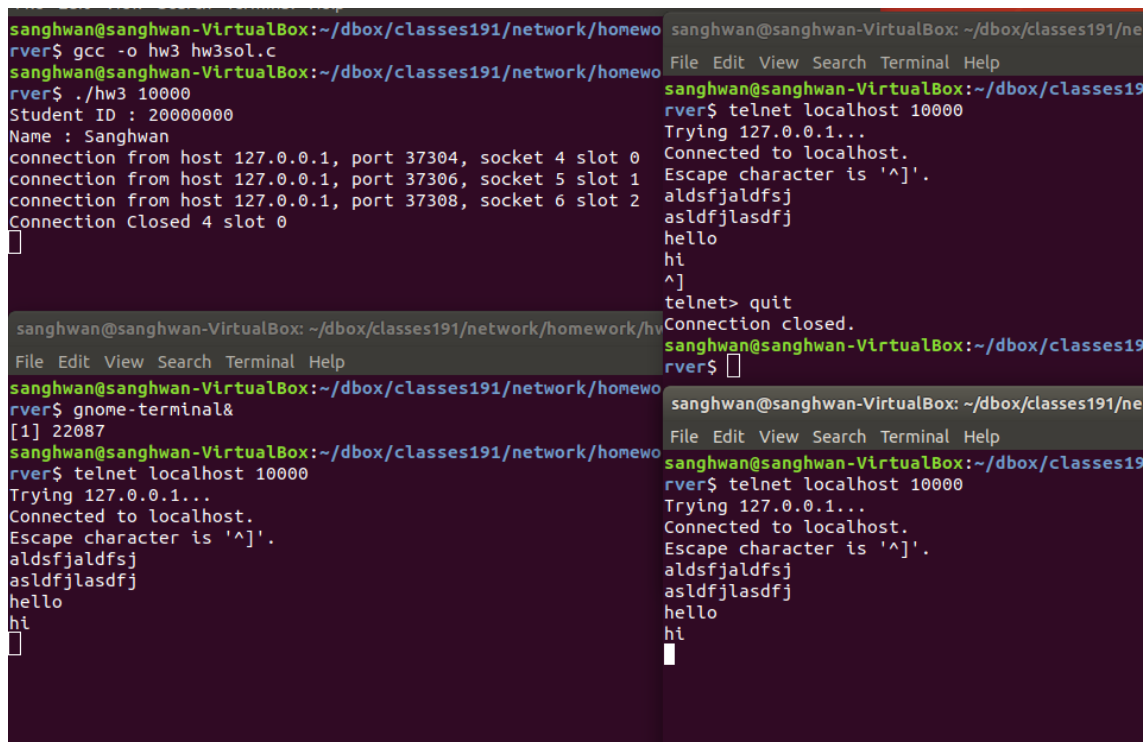
connection from host 127.0.0.1, port 45996, socket 4

이 상황에서 하나의 클라이언트 창에 문자를 타이핑하고 엔터를 치면, 모든 다른 클라이언트의 창에 그 메시지가 전달되어 보여진다.

한 클라이언트가 채팅을 종료하면, 서버도 socket이 끊어진 것으로 그 사실을 알게 된다. 이 때 아래와 같이 출력한다. Slot은 디버깅용이므로 출력하지 않아도 된다.

Connection Closed sd

- sd : 해당 연결의 socket descriptor



```
sanghwan@sanghwan-VirtualBox: ~/dbox/classes191/network/homework/hw3
rver$ gcc -o hw3 hw3sol.c
sanghwan@sanghwan-VirtualBox: ~/dbox/classes191/network/homework/hw3
rver$ ./hw3 10000
Student ID : 20000000
Name : Sanghwan
connection from host 127.0.0.1, port 37304, socket 4 slot 0
connection from host 127.0.0.1, port 37306, socket 5 slot 1
connection from host 127.0.0.1, port 37308, socket 6 slot 2
Connection Closed 4 slot 0
[1] 22087
sanghwan@sanghwan-VirtualBox: ~/dbox/classes191/network/homework/hw3
rver$ telnet localhost 10000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
aldfjaldfsj
asldfjlasdfj
hello
hi
^]
telnet> quit
Connection closed.
sanghwan@sanghwan-VirtualBox: ~/dbox/classes191/network/homework/hw3
rver$
```

한 클라이언트가 접속을 종료해도 남은 사용자들간의 채팅은 가능하다.

## 제공된 실행파일

- 실제로 구현된 실행파일을 첨부함. 테스트용으로 사용하기 바람.

## Template

- 주어진 “hw3.c” 를 수정하여 구현하기 바람.
- 맨 마지막 void display() 함수에 자신의 학번과 이름을 영어로 쓸 것.

## 구현방법

- 서버는 시작되면 주어진 포트 번호로 연결 대기한다. (bind, listen, select, accept 사용)
- 클라이언트로 연결요청이 들어올 때마다 생성된 socket descriptor를 배열에 저장한다. 최대 **16개**의 연결을 지원하도록 할 것
- 각 socket descriptor에 대해 **select()**를 통해 입력이 들어오는지 파악하고, 입력이 들어오면, 그 내용을 입력이 들어온 소켓을 제외하고, 나머지 소켓으로 그대로 전송한다 (read, write 사용). 즉 for loop를 돌면서 소켓들로 write를 진행하면 됨.

## 추가. 파이썬 사용자를 위하여...

이 과제는 파이썬으로 구현하여도 된다. 파이썬 3.0 이상을 사용하도록.

필요시 nonblocking IO에 관한 내용을 공부할 것.

<https://medium.com/vaidik Kapoor/understanding-non-blocking-i-o-with-python-part-1-ec31a2e2db9b>

소켓이 연결된 것이 있는 지 확인하기 위해 read()를 사용하면 blocking IO  
-> thread(), fork() 사용

```
File Edit View Search Terminal Help
sanghwan@sanghwan-VirtualBox:~/dbbox/classes191/network/homework/hw193_chattingse
rver$ python hw3.py 10000
Student ID : 20000000
Name : Sanghwan Lee
█
```