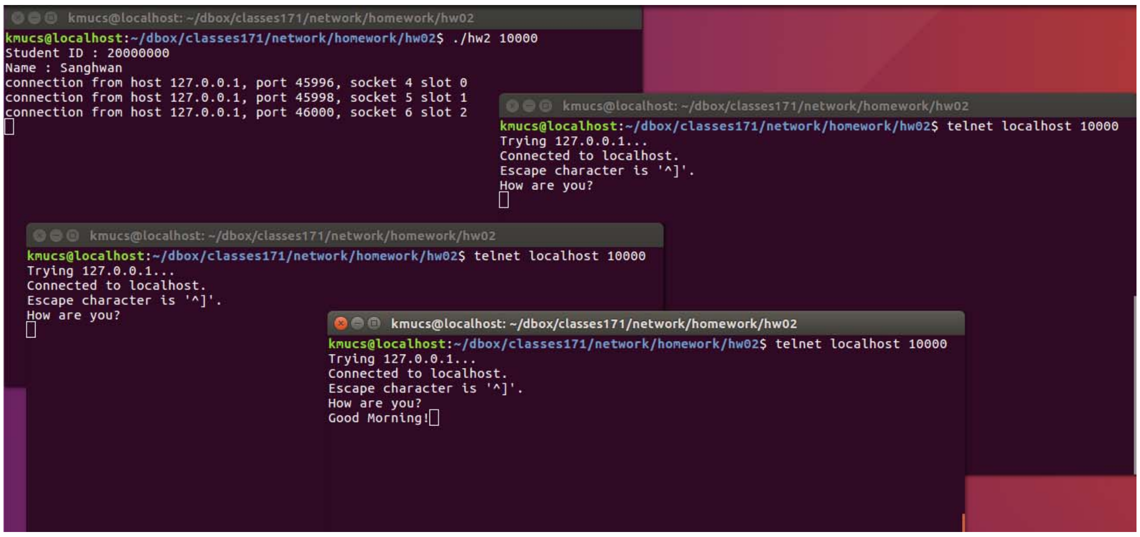
Computer Networks

HW #2 (30 Points)

Due date : 2017/5/8 Monday(eCampus)

Things to Submit :   studentid.c File (All code should be in one file.)

This homework is to implement a chatting server. telnet is used as a chatting client. The following image shows the use cases.



```
kmucs@localhost:~/dbox/classes171/network/homework/hw02$ ./hw2 10000
Student ID : 20000000
Name : Sanghwan
connection from host 127.0.0.1, port 45996, socket 4 slot 0
connection from host 127.0.0.1, port 45998, socket 5 slot 1
connection from host 127.0.0.1, port 46000, socket 6 slot 2
```

```
kmucs@localhost:~/dbox/classes171/network/homework/hw02$ telnet localhost 10000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
How are you?
```

```
kmucs@localhost:~/dbox/classes171/network/homework/hw02$ telnet localhost 10000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
How are you?
```

```
kmucs@localhost:~/dbox/classes171/network/homework/hw02$ telnet localhost 10000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
How are you?
Good Morning!
```

The server receives the tcp port number through the command line arguments. In other words, the server application accepts one argument with the following syntax.

$ hw2 tcpport

- tcpport : The tcp port number used by the clients.

In the above figure, 3 clients contact the server by the command "telnet localhost 10000". Since the clients and the server are running in the same machine, the clients use localhost as the server hostname. The server port number is 10000.

When the client sends the connection request to the server, the server prints out some information including the IP address of the client, the port number of the socket, and the socket descriptor. The last info, slot, is for debugging purposes, so it does not have to be printed.

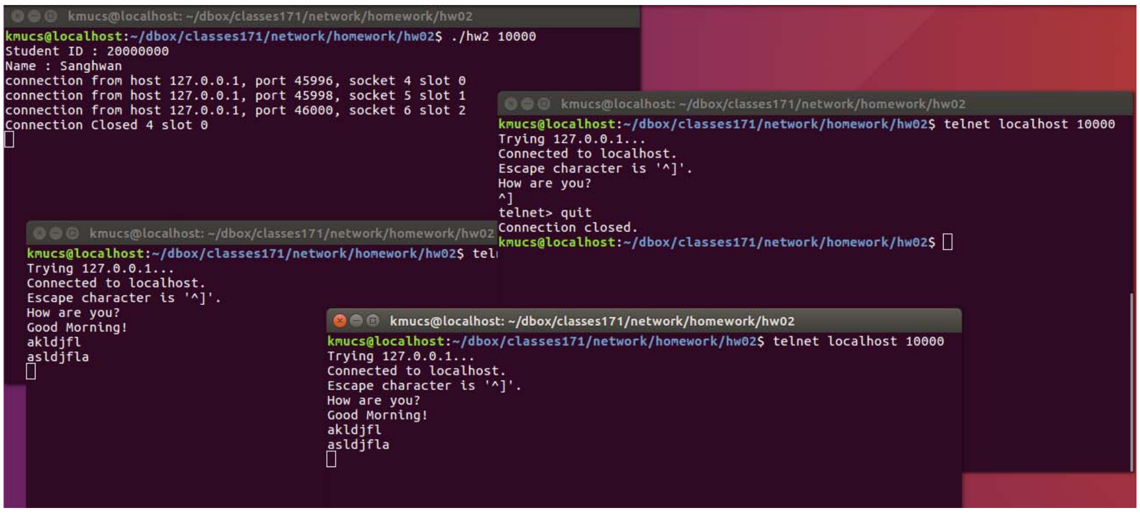
connection from host 127.0.0.1, port 45996, socket 4 slot 0

Under this situation, if one client types a string and enter key, the string should be sent to all the other clients. The other clients should display the string.

If one client ends the telnet application, the server should know that the socket is closed. Then, the following information should be display. Slot is for debugging purpose, so it does not have to be displayed.

Connection Closed sd slot sn

- sd : socket descriptor of the closed connection



```
kmucs@localhost: ~/dbox/classes171/network/homework/hw02
kmucs@localhost:~/dbox/classes171/network/homework/hw02$ ./hw2 10000
Student ID : 20000000
Name : Sanghwan
connection from host 127.0.0.1, port 45996, socket 4 slot 0
connection from host 127.0.0.1, port 45998, socket 5 slot 1
connection from host 127.0.0.1, port 46000, socket 6 slot 2
Connection Closed 4 slot 0
```

```
kmucs@localhost: ~/dbox/classes171/network/homework/hw02
kmucs@localhost:~/dbox/classes171/network/homework/hw02$ telnet localhost 10000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
How are you?
^]
telnet> quit
Connection closed.
kmucs@localhost:~/dbox/classes171/network/homework/hw02$
```

```
kmucs@localhost: ~/dbox/classes171/network/homework/hw02
kmucs@localhost:~/dbox/classes171/network/homework/hw02$ tel
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
How are you?
Good Morning!
akldjfl
asldjfla
```

```
kmucs@localhost: ~/dbox/classes171/network/homework/hw02
kmucs@localhost:~/dbox/classes171/network/homework/hw02$ telnet localhost 10000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
How are you?
Good Morning!
akldjfl
asldjfla
```

Even though one client ends the connection, the remaining clients should be able to chat.

An Executable file

- A working server executable file is attached. Use this file for test purpose.

Template

- Hw2.c contains some skeleton codes. You can start to implement from hw2.c
- Write your name and id at the function, void display(), which is at the end of hw2.c

Tips for implementation

- The server waits for the connection request through the port number given as the command line parameters. (bind, listen, select, accept functions are useful.)
- Whenever a client sends the connection request, the newly created socket

descriptor should be stored in an array. Maximum of 16 connections should be supported.

- Monitor each socket descriptor for any data transmission. If there is any input from a socket descriptor, the input data should be forwarded to all the other sockets except the incoming socket. (Use read and write function). To be specific, you may want to iterate a loop and call write() to one socket at a time.