컴퓨터공학과 201402391 이병만

실습 캡처

SSL 설정

```
mani@mani-VirtualBox:~/CN$ ls
chat_client.py  client.py  hash        server.csr  server.orig.key  trie
chat_server.py  garbage    server.crt  server.key  server.py
mani@mani-VirtualBox:~/CN$
```

[1] server

```
chat_client.py  chat_server.py  client.py  garbage  hash  server.crt  server.csr  server.key  server.
mani@mani-VirtualBox:~/CN$ py
py3clean      py3compile    py3versions   pydoc3       pydoc3.6      pygettext3    pygettext3.6  pyt
mani@mani-VirtualBox:~/CN$ python3 chat_server.py
server started on 127.0.0.1:1234
now listening.....
('127.0.0.1', 48346)  joined room
('127.0.0.1', 48348)  joined room
```

```
mani@mani-VirtualBox:~/CN$ python3 server.py
hi
mani@mani-VirtualBox:~/CN$
```

[2] client

```
199 packages can be updated.
0 updates are security updates.

Last login: Fri Dec  7 17:16:55 2018 from 192.168.43.69
mani@mani-VirtualBox:~$ cd CN/
mani@mani-VirtualBox:~/CN$ python3 chat_client.py
 you have connected successfully

TYPE>> hi

TYPE>>
('127.0.0.1', 48348) joined chat room
TYPE>>
('127.0.0.1', 48348):b'hihi'
TYPE>>
('127.0.0.1', 48348):b'hihi'
TYPE>>
```

```
1 새 세션 (5)  +
    https://ubuntu.com/livepatch

199 packages can be updated.
0 updates are security updates.

Last login: Fri Dec  7 17:19:21 2018 from 192.168.43.69
mani@mani-VirtualBox:~$ ls
CN       Documents  examples.desktop  Pictures  server.orig.key  Videos
Desktop  Downloads  Music             Public    Templates
mani@mani-VirtualBox:~$ cd CN/
mani@mani-VirtualBox:~/CN$ python3 chat_client.py
 you have connected successfully

TYPE>> hihi

TYPE>> hihi

TYPE>>
```

```
mani@mani-VirtualBox:~/CN$ python3 client.py
bye
mani@mani-VirtualBox:~/CN$
```

client code

```
def type_message():
     sys.stdout.write("\nTYPE>> ")
     sys.stdout.flush()

host = gethostname()
host = "127.0.0.1"
port = 1234
prop = (host, port)

context = ssl.SSLContext(ssl.PROTOCOL_TLSv1)

nsock = socket(AF_INET, SOCK_STREAM)

sock = context.wrap_socket(nsock, server_hostname=host)

try:
     sock.connect(prop) # 서버와 연결

except:
     print("unable to establish connection")
     sys.exit()

print(" you have connected successfully ")
type_message() # 값을 입력받음
```

```
while 3:
    all_sockets = [sys.stdin, sock] # 입력 값과 소켓을 저장
    readable, writable, error_s = select(all_sockets, [], [])

    for each_sock in readable:
        if each_sock == sock:
            data = each_sock.recv(2048) # 서버로부터 데이터를 받음

            if not data:
                print("\ndisconnected...")
                sys.exit()

            else:
                sys.stdout.write(data.decode('utf-8')) # 받은 데이터를 출력
                type_message()
        else:
             message = sys.stdin.readline()
             sock.send(message.rstrip('\n').encode()) # 입력 값을 데이터로 변환하여 전송
             type_message()
```

server code

```
def send_across(sock, client, message):
    for each_socket in all_sockets:
        if each_socket != sock and each_socket != client:
            try:
                message= "\n"+message
                each_socket.send(message.encode())
            except:
                each_socket.close()
                if each_socket in all_sockets:
                     all_sockets.remove(each_sock)

host = "127.0.0.1"
port = 1234
```

```
context = ssl.SSLContext(ssl.PROTOCOL_TLSv1)
context.load_cert_chain('server.crt', 'server.key')

all_sockets = []
properties = (host, port)

nsock = socket(AF_INET, SOCK_STREAM)
nsock.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)

print("server started on {0}:{1}".format(host, port))

nsock.bind(properties)
nsock.listen(10)

sock = context.wrap_socket(nsock, server_side = True)

print("now listening.....")

all_sockets.append(sock)
```

```
while True:
    readable_sock, writable_sock, error_sock = select(all_sockets, [], [], 0) #

    for all_socks in readable_sock:
        if all_socks == sock:
           sockdd, addr = sock.accept() # 새 클라이언트 연결
           all_sockets.append(sockdd) # 새 클라이언트를 추가

           print(addr, " joined room")
           to_send = str("{} joined chat room".format(addr))

           send_across(sock, sockdd, to_send) # 새로 연결된 클라이언트 정보를 모든 클라이언트에게 전송
        else:
            try:
               data = all_socks.recv(2048) # 클라이언트로부터 메시지를 받음

               if data:
                    send_across(sock, all_socks, str("{}:{}".format(addr,data))) # 작성한 클라이언트를 제외한 모든 클라이언트에게 전송

               else:
                    if all_socks in all_sockets:
                        all_sockets.remove(all_socks)

                    to_send2 = str("{} is offline".format(addr))
                    send_across(sock, all_socks, to_send2)

                    print(addr, " is offline")

            except:
                 to_send3= str("{} is offline".format(addr))
                 send_across(sock, all_socks,to_send3)

                 print(addr, "is offline\n")

                 continue
sock.close()
```