

Project #0 : TCP Socket Programming

이름 : 윤건우

학번 : 2019031776

과제 설명 - 내용 및 목적 서술

TCP Socket은 양쪽의 Socket이 연결된 상태여야 가능한 프로토콜로, 연결지향 프로토콜이라고 부르기도 한다.

이 과제에서는 실제로 TCP Socket을 파이썬 3.8 버전으로 구현하여 서버와 클라이언트 간 데이터를 직접 통신해 보는 것을 목표로 한다.

source Files : 본인이 작성한 소스 파일을 캡처 및 중요하고 필요하다고 생각하는 부분을 서술

<Server Source Code>

```
import socket
HOST = '127.0.0.1'
PORT = 50000
BUFFER = 4096
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.bind((HOST, PORT))
sock.listen(0)
print('tcpServer listen at: %s:%s\n\r' % (HOST, PORT))
while True:
    client_sock, client_addr = sock.accept()
    print('%s:%s connect' % client_addr)
    while True:
        recv = client_sock.recv(BUFFER)
        if not recv:
            client_sock.close()
            break
        print('[Client %s:%s said]:%s' % (client_addr[0], client_addr[1], recv))
        client_sock.sendall('tcpServer has received your message'.encode('utf-8'))
sock.close()
```

<Client Source Code>

```
import socket
HOST = '127.0.0.1'
PORT = 50000
BUFFER = 4096
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect((HOST, PORT))
sock.sendall('hello, tcpServer!'.encode('utf-8'))
recv = sock.recv(BUFFER)
print('[tcpServer said]: %s' %recv)
sock.close()
```

Socket의 send 함수의 경우 3.8 버전에서는 매개변수로 String이 아니라 Byte 형식의 매개변수가 들어와야 하기 때문에 sendall 함수를 사용하였고, 송신하고자 하는 메시지 내용을 String 클래스에 내장되어 있는 함수인 encode 함수를 이용하였다. '원하는 메시지'.encode('utf-8')을 통해서 utf-8 형식으로 인코딩하여 함수에 매개변수로 넘겨주었다.

Instructions: 본인의 소스 실행 방법 (본인이 실행 시킨 방법)

먼저 서버를 작성한 코드를 먼저 실행시키고, 그 이후 확인을 위해 Client의 코드를 실행시키는 방식으로 실습을 진행하였다.

How the program works: 프로그램 구동 방식

메커니즘 및 프로그램의 작동 방식 서술

서버를 먼저 실행시키면, while문에서 서버를 무한루프를 통해 실행시킨다. 서버를 무한루프로 실행시키는 이유는 서버는 보통 항상 실행되어 있는 상태로 client의 요청이나 연결을 기다리고 있어야하기 때문인데, 이러한 이유 때문에 이와 같은 코드를 작성하여 아마존 웹 서버와 같은 것을 이용하여 서버를 계속 켜 놓는다.

서버가 켜진 상태에서 Client 소켓에서는 서버 소켓으로 연결 요청을 하고, 해당 소켓에 메시지를 보내면 서버 소켓에서는 해당 메시지를 받고, 이 때 메시지를 BUFFER를 통해 받았을 경우 서버에서도 메시지를 받았다는 안내 메시지를 다시 클라이언트에게 보냄으로써 정상적으로 연결이 되었음을 클라이언트 소켓에게 전달한다.

Results: 결과 화면 캡처 및 설명.

<Server 실행 결과>

```
= RESTART: C:\Users\Inticoy\OneDrive\University\2020\2학기\컴퓨터네트워크\TCPsocket\server.py
tcpServer listen at: 127.0.0.1:50000
127.0.0.1:57157 connect
[Client 127.0.0.1:57157 said]:b'hello, tcpServer!'
```

<Client 실행 결과>

```
= RESTART: C:\Users\Inticoy\OneDrive\University\2020\2학기\컴퓨터네트워크\TCPsocket\client.py
[tcpServer said]: b'tcpServer has received your message'
>>>
```

파이썬에서 작성한 서버 소켓을 먼저 실행시킨 뒤, 실행되고 있음을 확인한 다음 Client 소켓을 실행시켜 원활하게 두 소켓이 메시지를 주고받음을 확인했다.

something Else: 과제에 대해서 건의점, 조교에게 전달되는 점, 질문, 앞으로의 과제에서의 희망사항 등