

# Computer Network



Computer Network

소켓프로그래밍 구현

2016025469

서건식

## 1. 구현 환경

Windows 10 , Python 3

서버와 클라이언트 프로그램을 각각 실행하여 테스트

## 2. 구현 사항

- 1) 로컬 호스트 환경에서 서버 - 클라이언트 간의 연결
- 2) 클라이언트 - 클라이언트 간에 채팅(1대다 통신) 구현

## 3. Server code description

```
#서건식_2016025469
#server

#모듈 import
import socket
import threading
import time

#서버의 주소와 포트
host = "127.0.0.1"
port = 8089

#소켓 생성
#user는 멀티쓰레드(1대다 클라이언트 접속)을 위한 배열
#접속한 클라이언트를 넣는다.
global user
user = []
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((host,port))
s.listen(1)
```

대부분 주석에 description을 달아 놓았습니다.

User는 클라이언트가 서버에 접속할 때 마다 리스트에 추가됩니다. (접속중인 클라이언트)

```
#서버 handle 함수
def chat(s):
    try:
        #채팅에 입장하면 입장 메시지를 출력
        id=s.recv(1024)
        message = "{} 님이 입장하셨습니다.".format(id.decode('utf-8'))

        #자신을 제외한 user들에게 출력시킨다.
        for i in user:
            if s!=i:
                i.send(message.encode('utf-8'))

        #메세지를 받을때마다 자신을 제외한 클라이언트에게 메세지를 보낸다.
        while s:
            print(message)
            data=s.recv(1024)
            message = "{} : {}".format(id.decode('utf-8'),data.decode('utf-8'))
            for i in user:
                if s!=i:
                    i.send(message.encode('utf-8'))

        #접속을 종료했을 때
    except:
        user.remove(s)
        message = "{} 님이 퇴장하셨습니다.".format(id.decode('utf-8'))
        print(message)
        if user:
            for i in user:
                i.send(message.encode('utf-8'))
```

Utf-8로 받을 때는 decode, 보낼 때는 encode함수를 사용하여 한글 채팅이 가능하도록 구현하였습니다.

접속에 성공하면 user에 클라이언트가 추가되고 자신을 제외하고 입장하였다는 메시지에 보내고, 접속을 종료하면 자신을 제외하고 퇴장하였다는 메시지를 보내고 user에서 클라이언트를 삭제합니다.

while문은 메시지를 받으면 메시지를 보낸 클라이언트를 제외하고 채팅 메시지를 보내주는 부분입니다.

```
#접속 대기
print("{}에 접속 대기중입니다.".format(host))

#accept를 받을 때 마다 새로운 클라이언트 쓰레드를 생성
while 1:
    connect_soc, addr = s.accept()
    user.append(connect_soc)
    threading._start_new_thread(chat,(connect_soc,))
```

Accept를 받을 때 마다 user에 클라이언트를 추가해주고, 쓰레드도 클라이언트마다 새로 생성해줍니다.

#### 4. Client code description

서버와 기본적인 설정은 거의 동일하므로, 함수만 설명하겠습니다.

```
#메세지를 받는 함수
#한글도 지원(utf-8)
def recv(s):
    while 1:
        data = s.recv(1024)
        data = data.decode('utf-8')

        if not data:
            continue
        else:
            print(data)

#메세지를 보내는 함수
#한글도 지원(utf-8)
def send(s):
    while 1:
        data = input("")
        if not data:
            continue
        else:
            s.send(data.encode('utf-8'))
```

Receive 와 send 함수입니다.

데이터를 받고, 맞다면 출력해주고 데이터를 입력 받아 보내는 함수입니다.

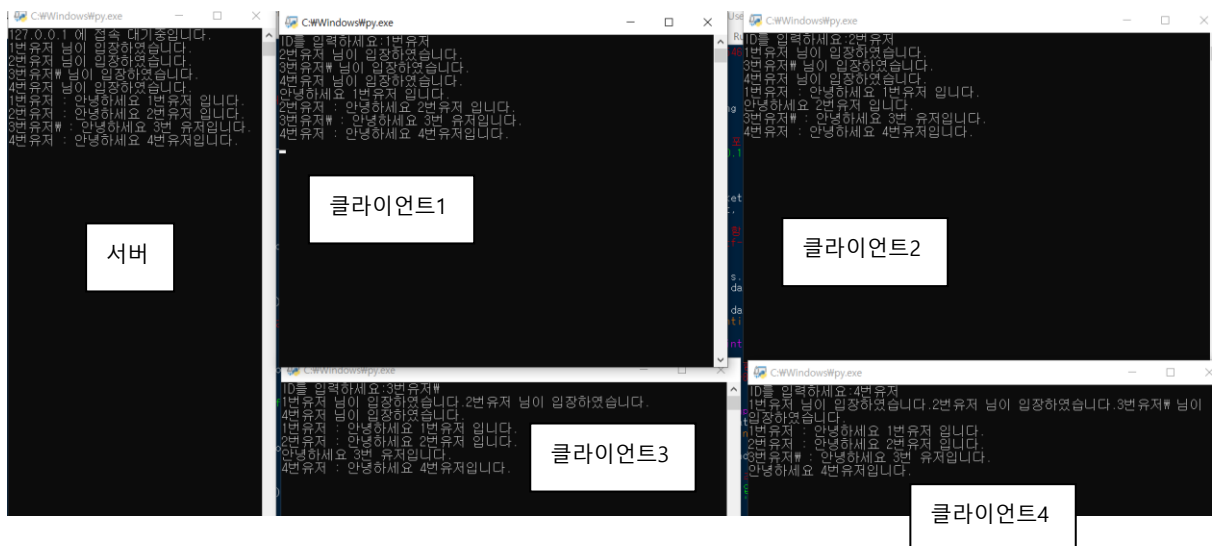
```
#id를 서버로 보내주고, | 상대에게 보낼 때 메시지와 함께 출력
id = input("ID를 입력하세요:")
s.send(id.encode('utf-8'))

#쓰레드를 각각 실행하여 동시에 일어날 수 있도록 함
threading._start_new_thread(send,(s,))
threading._start_new_thread(recv,(s,))

#인터프리터 언어이기 때문에 쓰레드를 계속 실행시키기 위한 코
while 1:
    pass
```

Description은 생략합니다.

## 5. 실행 결과



실행결과 입니다.

클라이언트 1번 -> 2번 ->3번 ->4번 순으로 접속 하였으며

메시지 전송도 같은 순서로 하였습니다.

결과에서 보이듯 자신이 서버로 보낸 내용은 자신의 화면에서 나오지 않고, 다른 사람이 보낸 메시지는 실시간으로 받는 것을 확인할 수 있습니다.

