

```

#select_server.py
#select를 이용한 TCP 멀티 에코 서버

import socket, select

if __name__=="__main__":
    #읽기 가능 소켓 목록
    socks = []
    BUFFER = 1024
    port = 2500

    s_sock = socket.socket() #TCP socket
    s_sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    s_sock.bind(('', port))
    s_sock.listen(5)

    #서버 소켓을 소켓 목록에 추가
    socks.append(s_sock)
    print(str(port) + "에서 접속 대기 중 ")

    #소켓에서 발생한 이벤트 처리
    while True:
        #이벤트 발생 소켓 설정(읽기 이벤트만 조사)
        r_sock, w_sock, e_sock = select.select(socks, [], [])

        #
        # 필요하면 이 부분에서 다른 코드를 처리할 수 있다
        #

        #읽기 소켓 리스트의 소켓 조사
        for s in r_sock:
            #읽기 이벤트 발생 소켓이 서버 소켓이면 연결 요청
            if s == s_sock:
                c_sock, addr = s_sock.accept() #새로운 클라이언트 연결
                socks.append(c_sock) #연결된 소켓을 소켓 리스트에 추가
                print(" Client (%s, %s) connectd" %addr)

            #읽기 이벤트 발생 소켓이 클라이언트면 데이터 수신
            else:
                #연결되지 않았으면 예외가 발생할 수 있으므로 try로 처리
                try:

```

```
#데이터를 수신하여 출력하고 다시 송신(에코)
data = s.recv(BUFFER) #데이터 수신
print("Received: ", data.decode())
if data:
    s.send(data) #수신 데이터를 다시 전송

#연결되지 않아 수신 데이터가 없음
except:
    print("Client (%s, %s) is offline" %addr)
    #소켓을 닫는다
    s.close()

#연결 종료된 소켓을 목록에서 제거
socks.remove(s)
continue

s_sock.close()
```