

데이터통신 2주차 실습 보고서

학번: 202102675

이름: 이문영

문제 1

과제에서, Sender와 Receiver 중 어느 쪽이 Server이고 어느 쪽이 Client인지 적고, 이유를 간단히 설명하시오.

서버(server) : 먼저 실행되어 있으며, 네트워크를 통해 어떤 연결 요청을 수락하는 역할

클라이언트(client) : 서버에 연결 요청을 시도하는 역할

서버와 클라이언트의 역할을 고려할 때, 요청을 대기하는 reciever.py가 서버라고 할 수 있다.

위 코드로 특정 ip와 port에 바인딩 후 sock.listen(1)로 해당 소켓으로의 연결 요청을 대기한다.

그리고 socket.accept()를 통해 클라이언트의 연결을 수락(accept)하고 conn.recv(1)로 이를 읽어들인다.

클라이언트의 경우 다음과 같이 어떤 ip 주소의 port로 연결(sock.connect)을 요청하는 sender.py가 클라이언트라 할 수 있다.

문제 2

파일 기반 통신과 소켓 기반 통신의 차이를 비교했을 때, 소켓 통신의 장점을 3가지 서술하시오.

- 파일 기반 통신은 데이터를 디스크에 저장하고 다시 읽어야 하므로 디스크 I/O가 발생하여 속도가 상대적으로 느리다. 반면에, 소켓 통신은 메모리에서 직접 데이터를 송수신하는 네트워크 I/O를 사용하므로 파일 기반보다 성능이 우수하다.
- 파일 통신의 경우 송수신 데이터의 동기화를 위해 signal을 통해 현재 파일의 상태를 계속 검사해야 하는 로직 "poling"이 필요하다. socket 통신의 경우 이 과정이 불필요하기에 파일 통신보다 CPU 자원을 효율적으로 사용한다고 볼 수 있다.
- 파일 기반 통신은 로컬에서만 가능하지만, 소켓 기반 통신은 서로 다른 컴퓨터 간의 통신도 가능하다.

01_file/reciever.py

```

BIT : 0
BIT : 0
BIT : 0
BIT : 1
BIT : 1
BIT : 0
BIT : 1
BIT : 1
BIT : 1
BIT : 1
Data received : ['01101000', '01100101', '01101100', '01101100', '01101111']
Message Converted: hello
이문영@DESKTOP-DVQT6QL MINGW64 ~/Desktop/3-1/01. 데이터 통신/데이터 통신 실습/week02/01_file
$

```

01_file/sender.py

```
cket
$ cd ..

이문영@DESKTOP-DVQT6QL MINGW64 ~/Desktop/3-1/01. 데이터 통신/데이터 통신 실습/week02
$ cd 01_file/

이문영@DESKTOP-DVQT6QL MINGW64 ~/Desktop/3-1/01. 데이터 통신/데이터 통신 실습/week02/01_file
$ python sender.py
Input string : hello
Bit string to send: 0110100001100101011011000110110001101111
Data sent : 0110100001100101011011000110110001101111

이문영@DESKTOP-DVQT6QL MINGW64 ~/Desktop/3-1/01. 데이터 통신/데이터 통신 실습/week02/01_file
$
```

02_socket/reciever.py

```
BIT : 0
BIT : 0
BIT : 1
BIT : 1
BIT : 0
BIT : 1
BIT : 1
BIT : 1
BIT : 1
Received data : 0110100001100101011011000110110001101111
Data received : ['01101000', '01100101', '01101100', '01101100', '01101111']
Message Converted: hello

이문영@DESKTOP-DVQT6QL MINGW64 ~/Desktop/3-1/01. 데이터 통신/데이터 통신 실습/week02/02_socket
cket
$
```

02.socket/sender.py

```
cket
$ python sender.py
Enter a string: hello
Bit string to send: 0110100001100101011011000110110001101111
Data sent: 0110100001100101011011000110110001101111

이문영@DESKTOP-DVQT6QL MINGW64 ~/Desktop/3-1/01. 데이터 통신/데이터 통신 실습/week02/02_so
cket
$ python sender.py
Enter a string: hello
Bit string to send: 0110100001100101011011000110110001101111
Data sent: 0110100001100101011011000110110001101111

이문영@DESKTOP-DVQT6QL MINGW64 ~/Desktop/3-1/01. 데이터 통신/데이터 통신 실습/week02/02_so
cket
$
```

제출 기한: 2025.03.16 + 추가제출 03.19까지

제출 방법: 사이버캠퍼스 제출 (보고서 + 실습 소스코드 + 과제 소스코드)