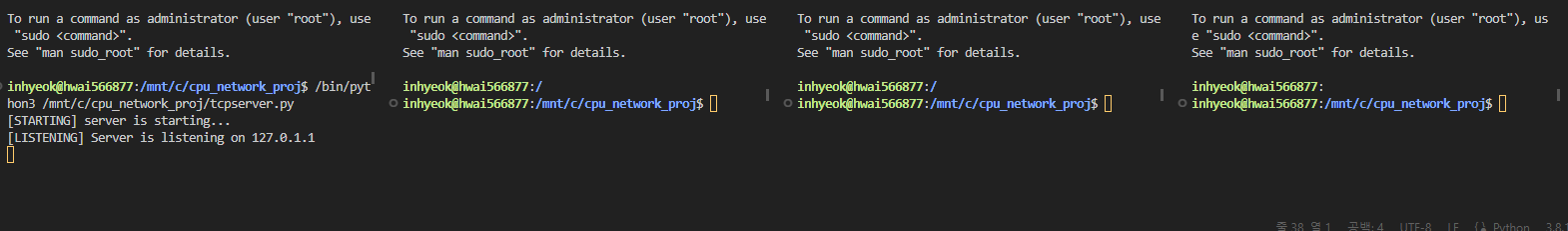
멀티 프로세스 기반 소켓 프로그래밍

2021058995 황인혁

멀티 프로세스 기반의 서버와 클라이언트를 제작하였습니다. 언어는 python을 사용하였으며, 서버에서 multiprocessing이라는 모듈을 사용하여 여러 클라이언트 프로세스들이 서버에 연결하여 서로 실시간 상호작용할 수 있도록 구현하였습니다.

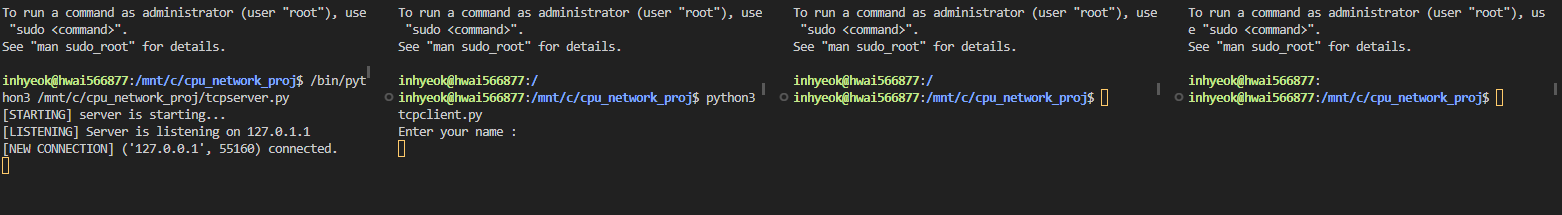
**코드 실행 결과**

1. 서버만 실행된 상태



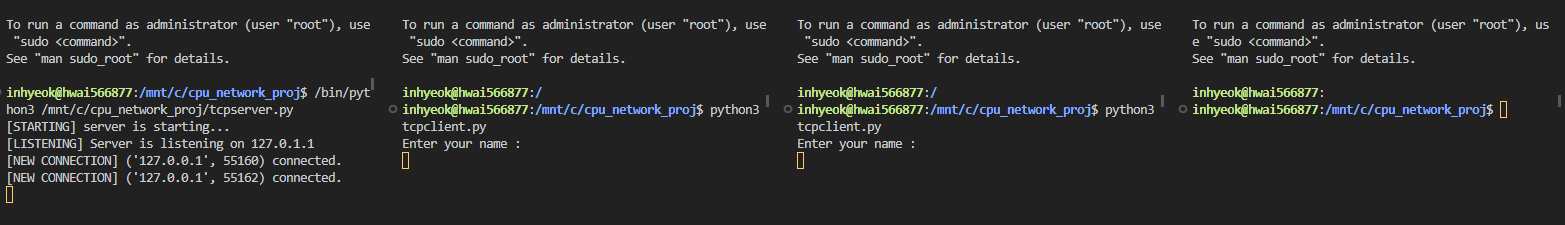
서버가 실행되면 시작되었다는 메시지와 함께 listen()함수를 실행하고 올바르게 listen이 이루어지고 있다는 사실을 출력합니다.

1. 첫번째 클라이언트 연결



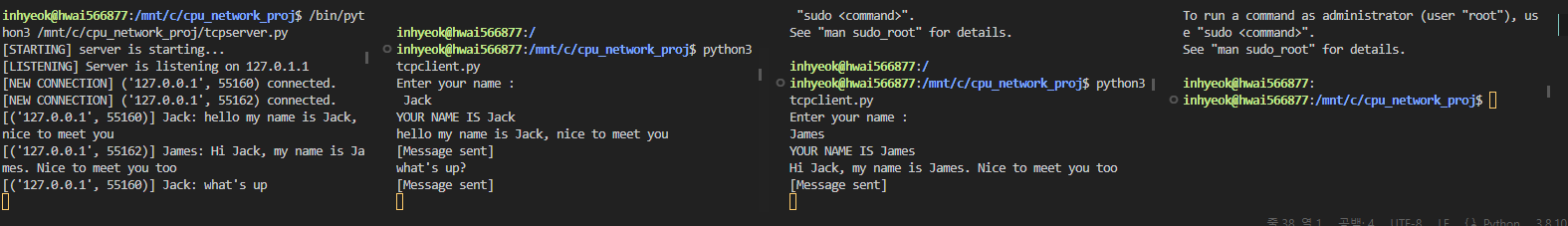
새로운 터미널에서 클라이언트를 실행합니다. 서버에서는 accept()함수를 통해 새로운 클라이언트에 대한 IP와 포트, 그리고 해당 클라이언트로 정보를 전달하는 객체인 conn을 반환합니다. 접속한 클라이언트에 대하여 process를 할당하고 실행합니다.

1. 두번째 클라이언트 연결



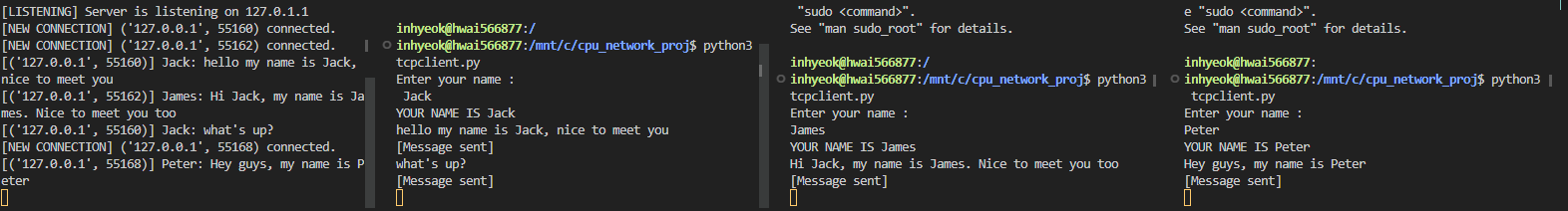
또 다른 터미널에서 클라이언트를 실행합니다. 2)번과 마찬가지의 작업이 server에서 수행됩니다.

1. 클라이언트 1과 2가 이름을 입력하고 상호작용



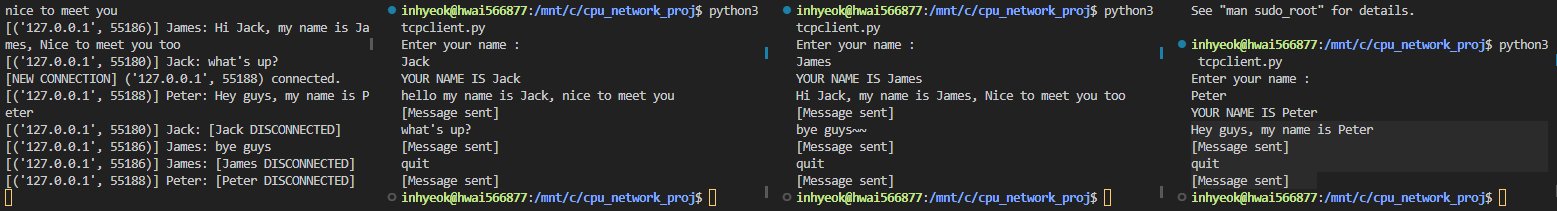
client에서 send함수를 통해 메시지를 보내면 server에서 recv함수를 통해 받고 누가 메세지를 보냈는 지와 함께 출력합니다. Server가 client의 메시지를 잘 받았다면 client에게도 메시지가 올바르게 잘 전송되었다는 사실을 보내줍니다.

1. 3번째 클라이언트 연결



또 다른 터미널에서 클라이언트를 실행합니다. 2)번과 마찬가지의 작업이 server에서 수행됩니다.

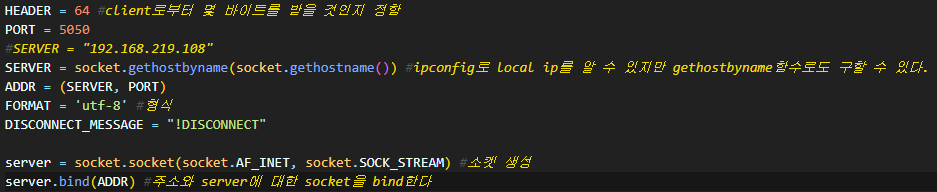
1. 연결 끊기



클라이언트에서 quit을 입력하면 클라이언트의 연결을 종료하고 server에서 conn.close() 함수를 실행합니다.

**코드의 구체적 흐름**

1. **SERVER**



기본적으로 필요한 변수들을 정의해주고 socket.socket()함수로 서버에 대한 소켓을 생성합니다.

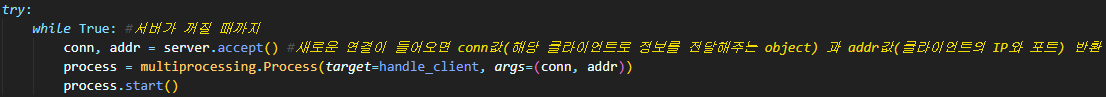
bind함수를 통해 소켓에 주소를 부여합니다.

Server는 두개의 함수  와 로 이루어져 있습니다.

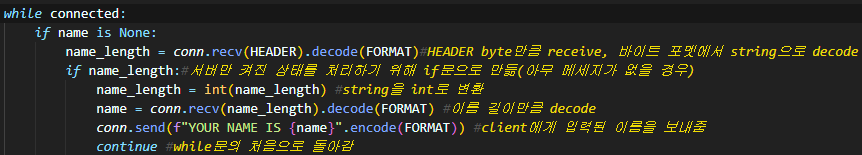
Server를 실행시키면  ‘server is starting…’을 출력하고 start 함수를 호출합니다.

Start 함수 내부에서는  가장 먼저 listen 함수를 통해 client를 기다리고 ‘[LISTENING] Server is listening on {서버IP}’를 출력하여 서버가 올바르게 listen 함수를 수행했다는 것을 알려줍니다.

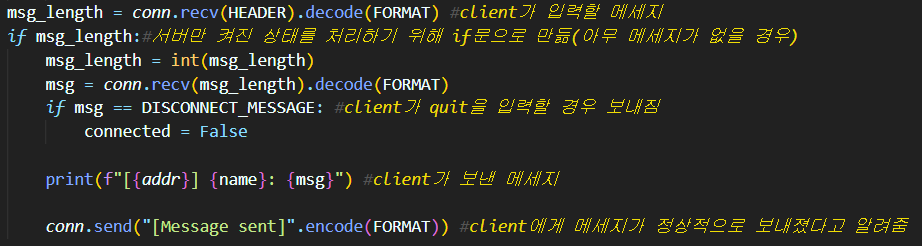
server.accept()함수는 새로운 client가 연결하기 전까지 기다립니다. 클라이언트 측에서 해당 서버로 접속한다면 accept함수의 반환 값을 매개변수로 handle\_client함수를 호출하고 process를 할당합니다.



handle\_client함수는 처음에 client로부터 입력한 이름을 받아옵니다. 이름은 앞으로 해당 client가 메시지를 입력할 때마다 ‘이름 : 메시지’ 형식으로 출력될 것입니다.

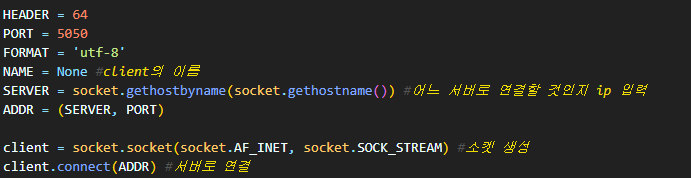


client로부터 이름이 입력되었다면 서버는 client로부터 메시지를 받아옵니다. 그 메시지가 만약 quit이라면 connection을 False로 만들어 연결을 끊습니다. print문을 통해서 받아온 해당 메시지를 주소, 이름과 함께 출력합니다.



Connected가 False가 되면서 while문을 빠져나오게 된다면 다음 함수를 통해서 연결을 종료합니다.

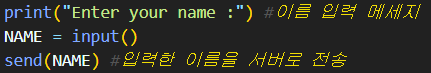
1. **Client**



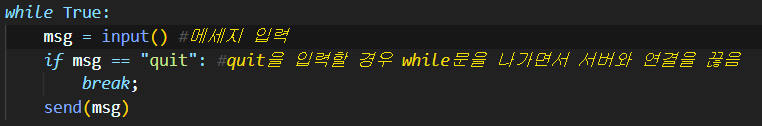
기본적으로 필요한 변수들을 정의해주고 socket.socket()함수로 클라이언트에 대한 소켓을 생성합니다.

Connect함수를 통해 서버로 연결합니다.

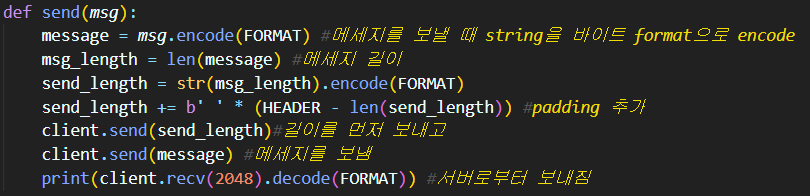
Client는 가장 먼저 이름을 입력하도록 되어있습니다. 입력한 이름을 send 함수를 통해 서버로 전송합니다.



이름이 입력된 뒤 client가 입력하는 값들은 msg 값입니다. Quit이 입력되지 않는 한 계속 입력이 가능하고 입력된 값은 서버로 전송됩니다.



Send 함수는 다음과 같이 구성되어 있습니다. 보내기 전에 string형식의 msg를 byte형식으로 encode한 다음에 전송하여야 한다. 서버가 메시지를 잘 받았다면 다시 클라이언트에게 [Message sent]를 보내며 서버에 메시지가 올바르게 전송되었다는 것을 확인해준다.



마지막으로 클라이언트는 연결을 해제 했다는 것을 서버로 전송함.

