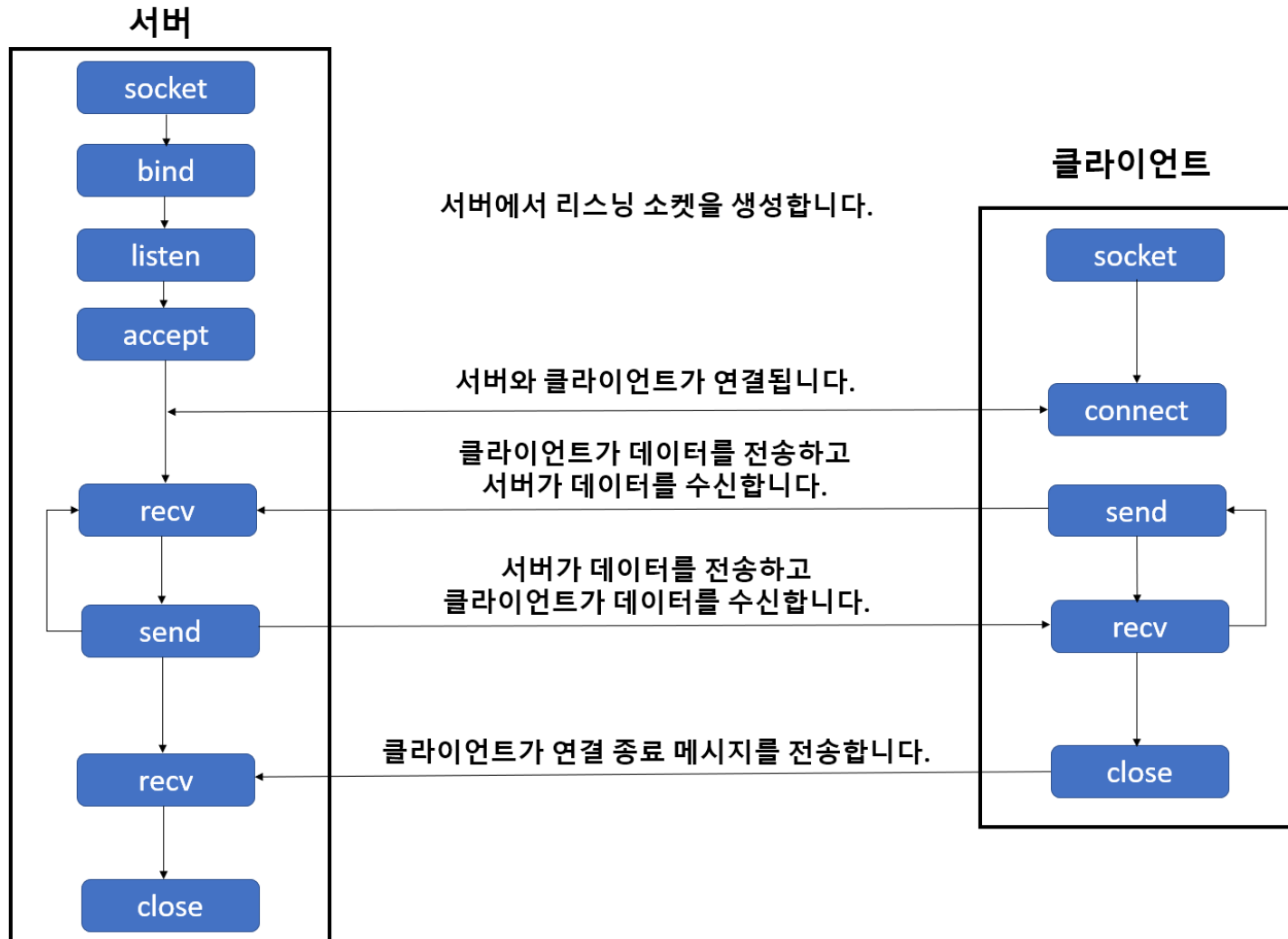

네트워크 프로그래밍

❖ TCP 통신 절차



네트워크 프로그래밍

❖ echo-server.py

```
import socket

HOST = '127.0.0.1'
PORT = 9999

# 주소 체계(address family): IPv4, 소켓 타입: TCP
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# 이미 열린 포트 충돌시 재사용 옵션 설정
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

server_socket.bind((HOST, PORT))
server_socket.listen()
# accept 함수에서 대기, 클라이언트 접속 시 새로운 소켓을 리턴
client_socket, addr = server_socket.accept()

# 접속한 클라이언트의 주소 출력
print('Connected by', addr)
```

네트워크 프로그래밍

❖ echo-server.py

```
while True:
    # 메시지 수신 대기
    data = client_socket.recv(1024)
    if not data:
        break

    # data(byte array)를 문자열로 변환하여 출력
    print('Received from', addr, data.decode())

    # 받은 문자열을 다시 클라이언트로 전송(에코)
    client_socket.sendall(data)

client_socket.close()
server_socket.close()
```

네트워크 프로그래밍

❖ echo-client.py

```
import socket

HOST = '127.0.0.1'
PORT = 9999

client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# 서버와 연결
client_socket.connect((HOST, PORT))

# 메시지 전송
client_socket.sendall('안녕'.encode())

# 메시지 수신
data = client_socket.recv(1024)
print('Received', repr(data.decode()))

client_socket.close()
```

네트워크 프로그래밍

❖ 쓰레드를 사용한 에코 서버와 클라이언트

- 기존 서버는 한 클라이언트만 서비스하고 종료
 - 동시에 여러 클라이언트를 서비스하지 못함
- 동시에 여러 클라이언트를 서비스 제공
 - 접속 하나 당 전담 작업 스레드 운영

네트워크 프로그래밍

❖ multiconn-server.py

```
import socket
from _thread import *

def threaded(client_socket, addr):
    print('Connected by ', addr[0], ':', addr[1])
    while True:
        try:
            # 데이터가 수신되면 클라이언트에 다시 전송합니다.(에코)
            data = client_socket.recv(1024)
            if not data:
                print('Disconnected by ' + addr[0],':',addr[1])
                break

            print('Received from ' + addr[0],':',addr[1] , data.decode())
            client_socket.send(data)

        except ConnectionResetError as e:
            print('Disconnected by ' + addr[0],':',addr[1])
            break

    client_socket.close()
```

네트워크 프로그래밍

❖ multiconn-server.py

```
HOST = '127.0.0.1'
PORT = 9999

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server_socket.bind((HOST, PORT))
server_socket.listen()

print('server start')

while True:
    print('wait')
    client_socket, addr = server_socket.accept()
    start_new_thread(threaded, (client_socket, addr))

server_socket.close()
```

파일 전송 프로그램

파일 전송 프로그램

❖ file_sender.py

```
import os
import socket

HOST = "127.0.0.1"
PORT = 6000
FILE_PATH = "c:/temp/ShopDB.sql"

def file_read(file_path):
    with open(file_path, 'rb') as f:
        while True :
            data = f.read(1024)
            if not data :
                break
            yield data
```

파일 전송 프로그램

❖ file_sender.py

```
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    try :
        s.connect((HOST, PORT))
        fileSize = os.path.getsize(FILE_PATH)

        #파일 크기 전송
        print("전송 파일 크기", fileSize)
        s.sendall(str(fileSize).encode())

        # 준비 상태 수신
        isready = s.recv(1024).decode()
        if isready == "ready":
            # 파일 전송
            for data in file_read(FILE_PATH):
                s.sendall(data)
            print("전송완료")

    except Exception as e:
        print(e)
```

네트워크 프로그래밍

❖ file_receiver.py

```
from _thread import *
import socket

HOST = "127.0.0.1"
PORT = 6000
FILE_PATH = "c:/temp/received/data"

def receive_thread(client_socket, addr):
    try :
        ## 파일 크기 수신
        size = client_socket.recv(1024)
        size = int(size.decode())
        print("수신할 파일 크기:", size)

        # 준비상태 전송
        client_socket.send("ready".encode())
```

파일 전송 프로그램

❖ file_receiver.py

```
# 파일 수신
total_size = 0
with open(FILE_PATH, "wb") as f:
    while True:
        data = client_socket.recv(1024)
        f.write(data)
        total_size += len(data)
        if total_size >= size: break

    print(f"수신 완료: {total_size} bytes")
except Exception as e:
    print(e)
finally:
    client_socket.close()
```

파일 전송 프로그램

❖ file_receiver.py

```
with socket.socket(family=socket.AF_INET, type=socket.SOCK_STREAM) as s:
    try:
        s.bind((HOST, PORT))
        s.listen(1)

        while True:
            client_socket, addr = s.accept() # 접속 대기
            # 스레드 기동
            start_new_thread(receive_thread, (client_socket, addr))

    except Exception as e:
        print(e)
```