

2023-2 Computer Networks

# Socket Programming and Project #1

2023. 10. 11.

# 01

Project 개요

## Project 설명

TCP / UDP 소켓 프로그래밍을 활용한 채팅 application 개발



단계	구현 기능
1차 (10/11)	문자열 전송 기능
2차 (11/1)	파일 전송 기능
3차 (11/15)	주변 device 인식 기능

- 2인 1조 (1인 1조 가능)
  - 팀 구성 및 팀 명(application 이름) 선정하여 google sheet에 기입
  - [https://docs.google.com/spreadsheets/d/1UIHc5TzyP2PHysdAJCUYi2v2PZ1YbQMbxWqe\\_ex2j-8/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1UIHc5TzyP2PHysdAJCUYi2v2PZ1YbQMbxWqe_ex2j-8/edit?usp=sharing)
- GUI 구현된 기본 코드 제공
- 제공된 코드의 빈 함수를 완성하여 기능 구현
- 사용언어: Python

# Project 설명

main.py 실행 -> Server - Client 초기 연결 화면

- Server / Client 선택
- IP Address: 서버 역할 컴퓨터의 IP 주소 입력
- TCP / UDP port: 임의의 port number 입력

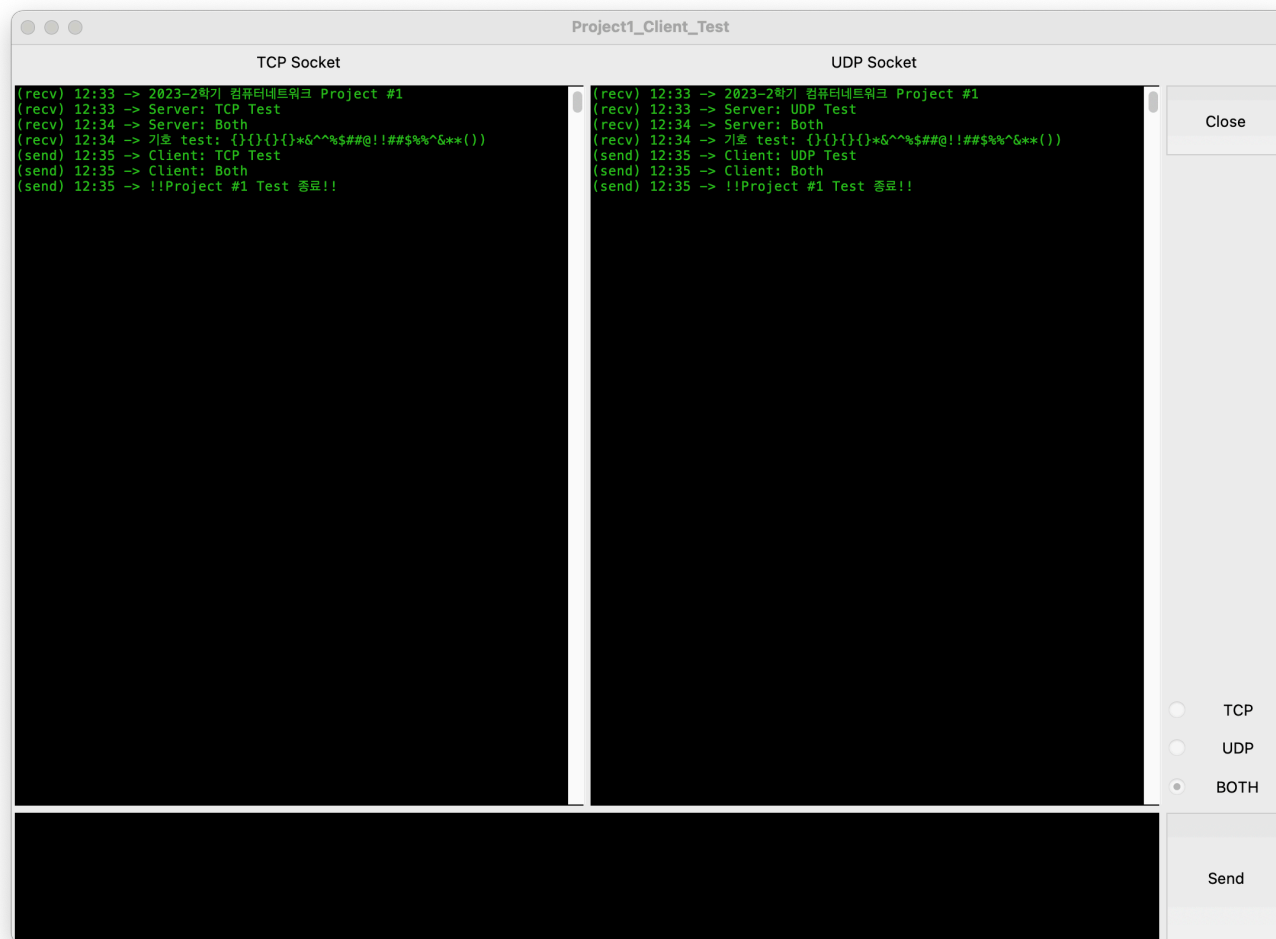
The screenshot shows the 'Computer Network Project' window with the 'Server' radio button selected. The 'IP Address' field is empty. The 'Input Address' field is empty. The 'TCP Port' field contains '4000' and the 'UDP Port' field contains '2000'. The 'Team Name' field is empty. The 'Connect' button is visible at the bottom.

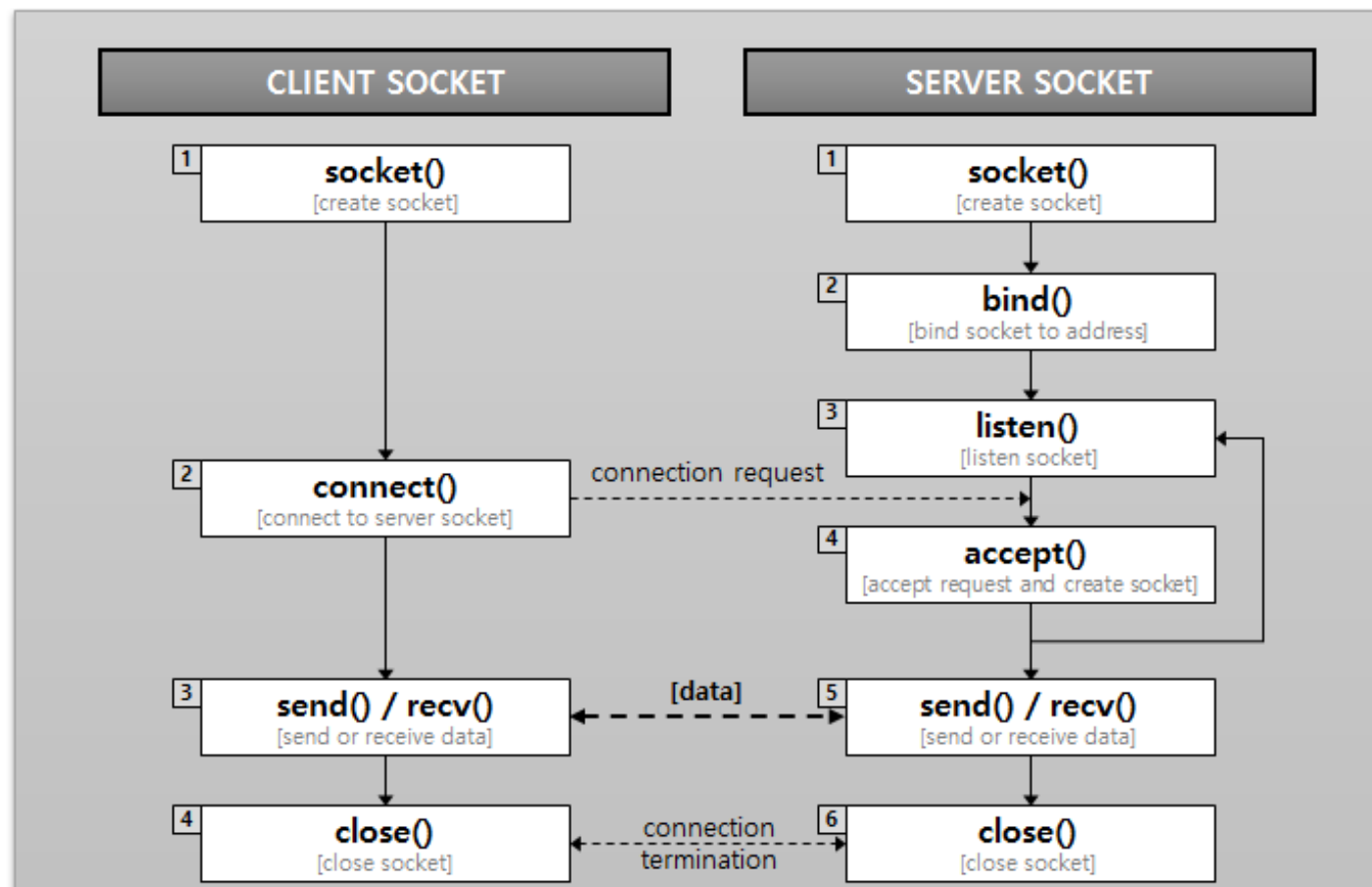
The screenshot shows the 'Computer Network Project' window with the 'Server' radio button selected. The 'IP Address' field is empty. The 'Input Address' field is empty. The 'TCP Port' field contains '4000' and the 'UDP Port' field contains '2000'. The 'Team Name' field is empty. The 'wait connection' button is visible at the bottom. The window has a red close button in the title bar.

The screenshot shows the 'Computer Network Project' window with the 'Client' radio button selected. The 'IP Address' field is empty. The 'Input Address' field is empty. The 'TCP Port' field contains '4000' and the 'UDP Port' field contains '2000'. The 'Team Name' field is empty. The 'Connect' button is visible at the bottom.

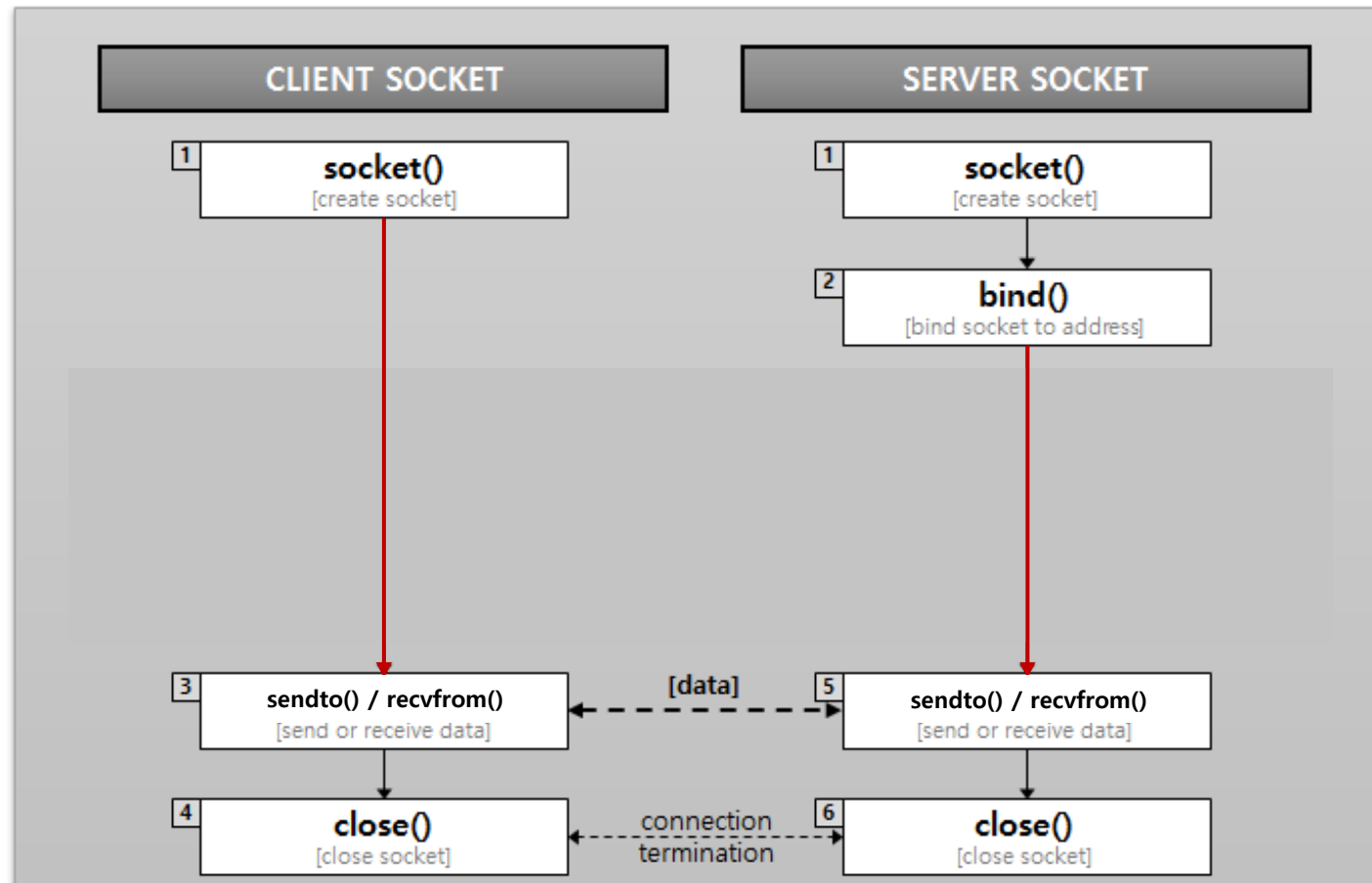
# 동작 화면

## Project 설명





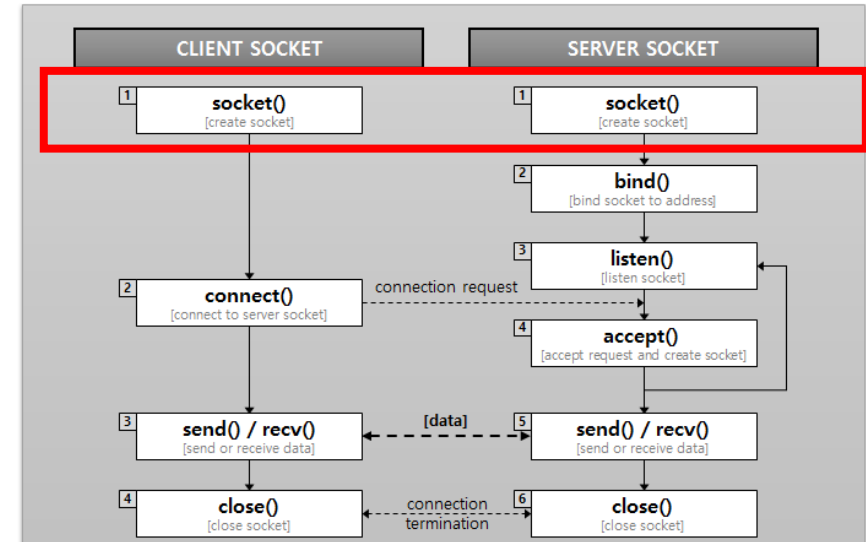
# Socket Programming





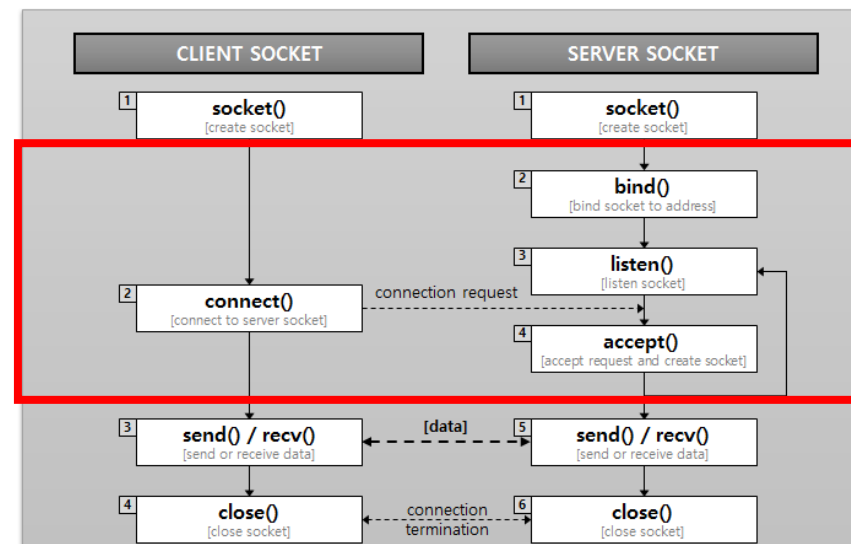
# Socket Programming

- Library 사용
  - import socket
- Socket 생성
  - sock = socket.socket(family, type)
    - family: AF\_INET(IPv4), AF\_INET6(IPv6)
    - type: SOCK\_STREAM(TCP), SOCK\_DGRAM(UDP)
    - ex) sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)



## Server

- IP address 및 port number 할당
  - `sock.bind((ip_address, port))`
- 연결 대기
  - `sock.listen(max_num)`
  - `max_num`: 대기 가능한 연결의 최대 개수
- 연결 허용
  - `conn_sock, addr = sock.accept()`
  - `conn_sock`: 데이터 전송을 위한 새로운 socket

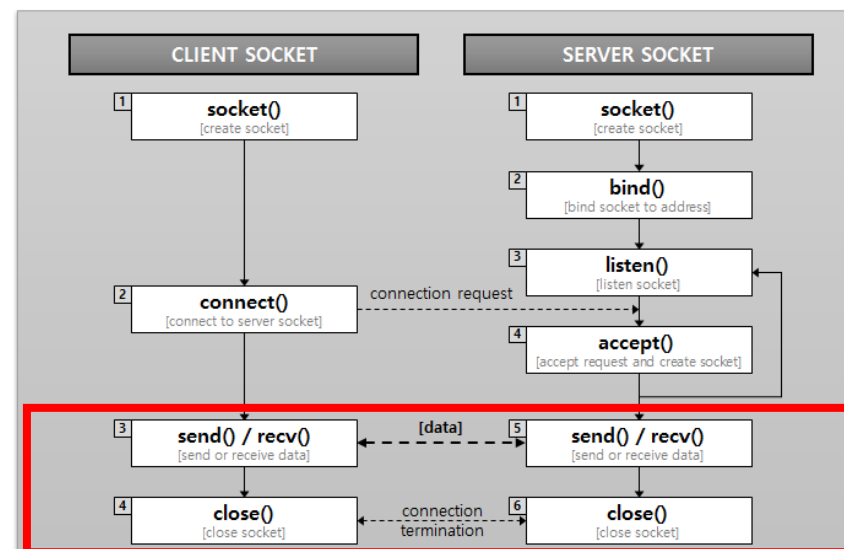


## Client

- 연결 요청
  - `sock.connect((ip_address, port))`
  - Server의 ip address와 port number

# Socket Programming

- 데이터 전송
  - TCP: `conn.send(data)`
  - UDP: `sock.sendto(data, (ip_address, port))`
- 데이터 수신
  - TCP: `conn.recv(buff_size)`
  - UDP: `data, addr = sock.recvfrom(buff_size)`
  - `buff_size`: 수신 가능한 데이터의 최대 크기
- 소켓 닫기
  - `conn.close()` / `sock.close()`



<https://docs.python.org/3/library/socket.html>

## 03 유의사항

- pj\_1.py 파일 내 빈 함수 구현
  - 각 함수에서 요구하는 기능 구현
- 상위 레이어 프로토콜 사용 불가
- GUI 변경 가능
- Server와 Client가 서로 10회 이상 다른 메시지를 주고 받을 수 있어야 함

## 04 제출방법

- 팀원 중 한 명만 제출
- 제출 파일 (총 2개의 파일)
  - Project 압축 파일 (파일명: 팀이름\_zip)
  - 보고서 (파일명: 팀이름\_pdf)
    - ✓ 구현 환경: OS 정보, python version
    - ✓ 구현 코드: 스크린샷/텍스트 첨부 가능, 구현 코드에 대한 간단한 설명
    - ✓ 정상 동작 스크린샷 (server와 client 실행 창)
- 제출 마감
  - 10월 29일(일), 23시 59분
  - 단, 질문은 27일(금) 18시까지만 가능

# Thank You