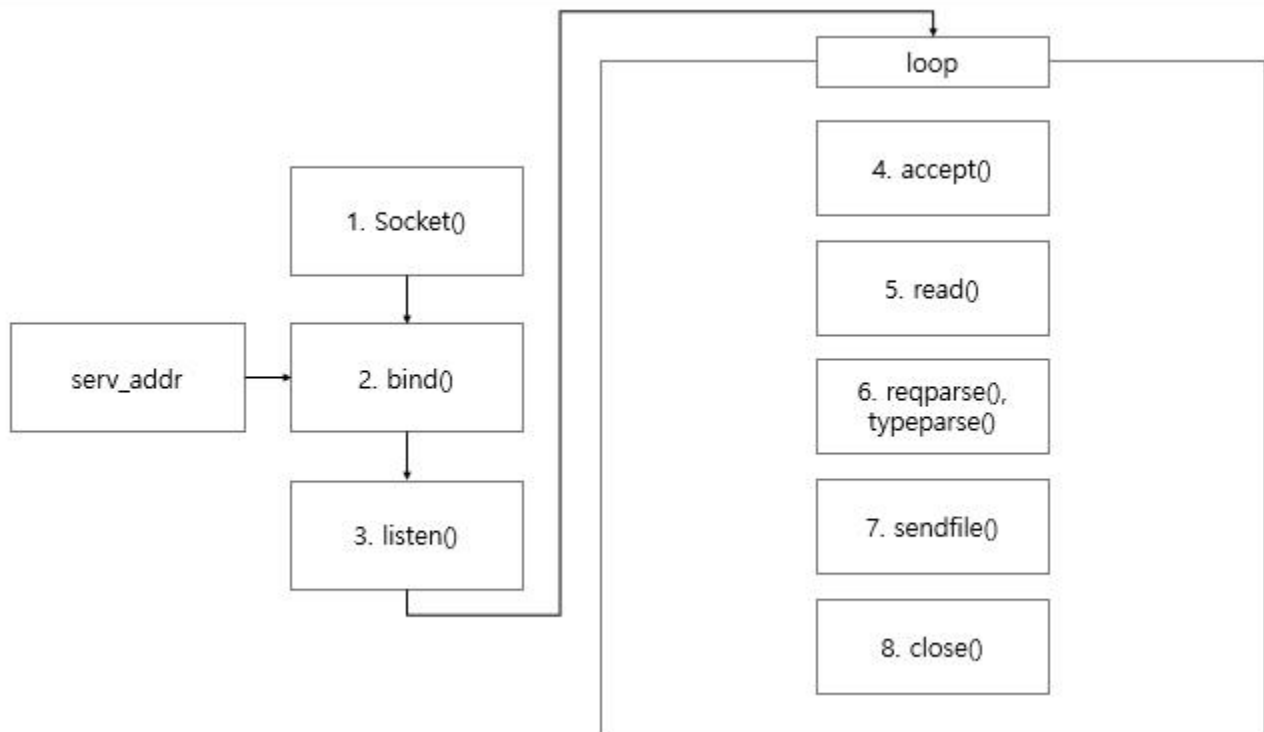


# Network Socket Programming Report

2017012106 손민우

## 1. server's design



## Description

1. socket() : 소켓 생성, 인자로 AF\_INET(internet address domain), SOCK\_STREAM(stream type) 줌
2. bind() : 서버 정보가 담긴 serv\_addr 구조체와 생성한 socket을 bind해줌
3. listen() : 생성한 socket을 인자로 주고 request가 왔을 시 accept 할 수 있도록 대기상태에 둬
4. accept() : client에서 request가 왔을 때 새로운 socket을 생성해 기존 socket과 client 정보를 담은 구조체를 인자로 주면서 통신 시작
5. read() : client가 request를 보낸 내용을 인자로 준 buffer 배열에 저장
6. reqparse() : request message를 parsing해 w/wn으로 구분된 field정보 수집. req, page 배열을 인자로 주고 요청된 파일 이름을 page 배열에 저장. (req는 request message)
- 6-1. typeparse() : reqparse()에서 수집한 page 배열에 저장된 정보로 content-type 정보 알아내어 type 배열에 저장. html, gif, jpeg, mp3, pdf 지원
7. sendfile() : 6에서 알아낸 정보 기반으로 response 전달. header 배열에 response field 형식에 맞추어 HTTP/1.1 200 OK, Content-length, Content-Type 저장. 요청된 파일을 open()으로 열어 buffer 배열에 저장. header와 buffer를 target배열에 저장한 후 write()로 response 전송
8. close() : 통신이 끝나면 socket 연결 종료

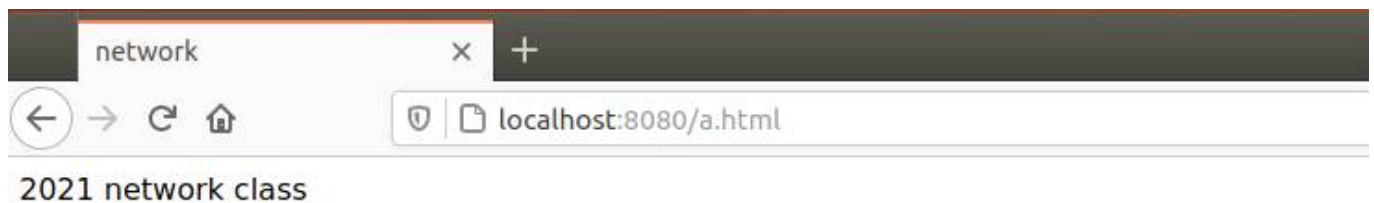
## 2. What difficulties did you face and how did you solve them?

response header 작성 시 content-length 넣는 부분에서 작성을 잘못하여 connection reset 이 계속 떠서 이 부분을 해결하는 것이 어려웠습니다.

## 3. Include and briefly explain some sample outputs of your client-server

```
<meta charset="UTF8">
<html>
<head><title>network</title>
<body>
2021 network class
</body>
</html>
```

<a.html>



↑ <firefox browser output>

```
os@os-VirtualBox:~/Desktop/ClientServer_Example/server$ ./server 8080
Here is the message: GET /a.html HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

Here is the message: GET /favicon.ico HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://localhost:8080/a.html
```

↑ <server output>

Client인 firefox browser에서 요청한 request가 <server output>에 보인다. GET method로 a.html 파일을 HTTP/1.1 버전으로 요청했고 그 외 request header 내용들이 protocol field 형식에 맞춰 server로 request 되었다. html파일과 더불어 favicon또한 요청했다.

Client에 대한 response로 server가 보낸 message를 browser에서 해석해 <firefox browser output>에 출력되었다. a.html에 있는 내용이 잘 전송된 것을 알 수 있다. favicon은 요청되었지만 폴더 내에 없으므로 전송되지 않은 모습을 볼 수 있다.