

Description of my server's design

1. 우선 `socket()`함수를 통해 -> TCP socket을 생성합니다.
2. 입력한 포트 번호를 `Bind()`함수를 사용해 -> 소켓에 서버 주소를 고정합니다.
3. `Listen()`함수를 이용해 client의 연결을 기다립니다.
4. Client로부터 request가 무한정으로 올 수 있으므로 `while(1)`을 이용해 반복문을 만듭니다.
즉, 아래 내용을 반복합니다.
5. Client와 연결이 되면 `accept()`함수를 통해 client 측 소켓을 생성합니다.
6. `Fork()`를 이용해 자식 프로세스를 생성해서 요청을 처리합니다.
7. 요청을 처리하는 방식은 우선 `read()`를 통해 client의 request를 받아옵니다.
8. 받은 request의 내용을 토대로 필요한 정보를 추출합니다. (요청한 파일명, http버전 등)
9. 요청한 파일 명을 토대로 파일을 호출합니다. 만약 파일이 없다면 response header에 404 not found를 전달해주고 client측 소켓을 닫습니다.
10. 파일이 있다면 파일의 상태나 형식 등 추출한 정보를 통해 response header를 작성합니다. (ex> Content-type)
11. 작성한 response header를 client에게 `write()`를 사용해 전달해줍니다.
12. Response의 data부분은 `fread()`를 통해 불러온 파일의 데이터를 1023byte씩 끊어서 담아서 `write()`를 사용해 client에게 전달해줍니다.
13. Client 측 소켓을 닫습니다.

어렵거나 난해했던 점

Browser에 response를 어떻게 보내야할 지부터 해서 프로젝트의 시작이 가장 어려웠다고 생각하지만 그래도 가장 저를 애먹였던 문제는 write() 함수였습니다.

어느정도 코드를 완성하고 test case들을 시험해보는 단계에서

Server에서 browser로 http response를 보내는 상황에서 Content-Type function에서 HTML도 받아지고 MP3, PDF도 받아지고 browser화면에도 출력이 되는데 오직 이미지 파일들 (jpeg,png,gif)이 browser에서 받아지는데 화면에 출력이 안되는 상황에 부딪혔습니다.

차라리 다른 파일들도 출력이 안된다면 문제점을 찾을 수 있겠는데 이미지만 출력이 안되다보니 이를 간 무엇이 문제인지 구글링도 수십번을 해보고 지인들에게 도움도 요청해보고 코드도 같이 읽어보고 했는데 해결이 안되다가 결국에 원인을 찾았습니다.

제 코드의 `write(sock, hpbuf, 256);` 이 http response의 헤더를 전송해주는 부분인데

제가 write에 대한 이해도가 부족했었는지 buffer의 크기를 256으로 고정시켜놔서

garbage 값이 response header부분에 들어갔던 것 같습니다.

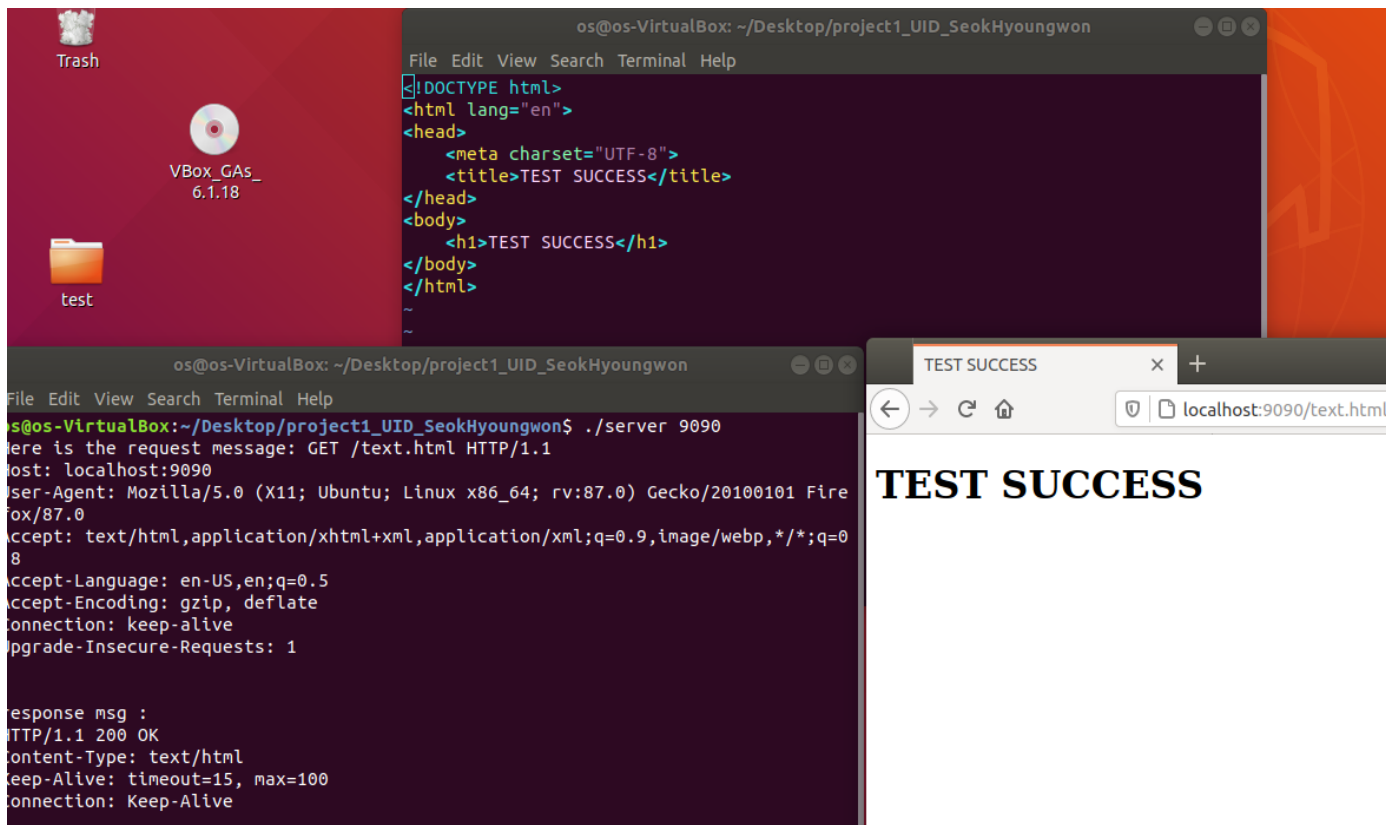
이를 `write(sock, hpbuf, strlen(hpbuf));` 로 수정함으로써 결국에 문제를 해결했습니다.

출력 화면

1. Request message (<http://localhost:9090/text.html>)

```
os@os-VirtualBox: ~/Desktop/project1_UID_SeokHyoungwon
File Edit View Search Terminal Help
os@os-VirtualBox:~/Desktop/project1_UID_SeokHyoungwon$ ./server 9090
Here is the request message: GET /text.html HTTP/1.1
Host: localhost:9090
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
```

2. Html (<http://localhost:9090/text.html>)



The screenshot displays a desktop environment with a red background. On the desktop, there is a 'Trash' icon, a 'VBox_GAs_6.1.18' icon, and a 'test' folder icon. A terminal window is open, showing the output of the './server 9090' command. The terminal output includes the request message and the response message. A web browser window is also open, displaying the content 'TEST SUCCESS'.

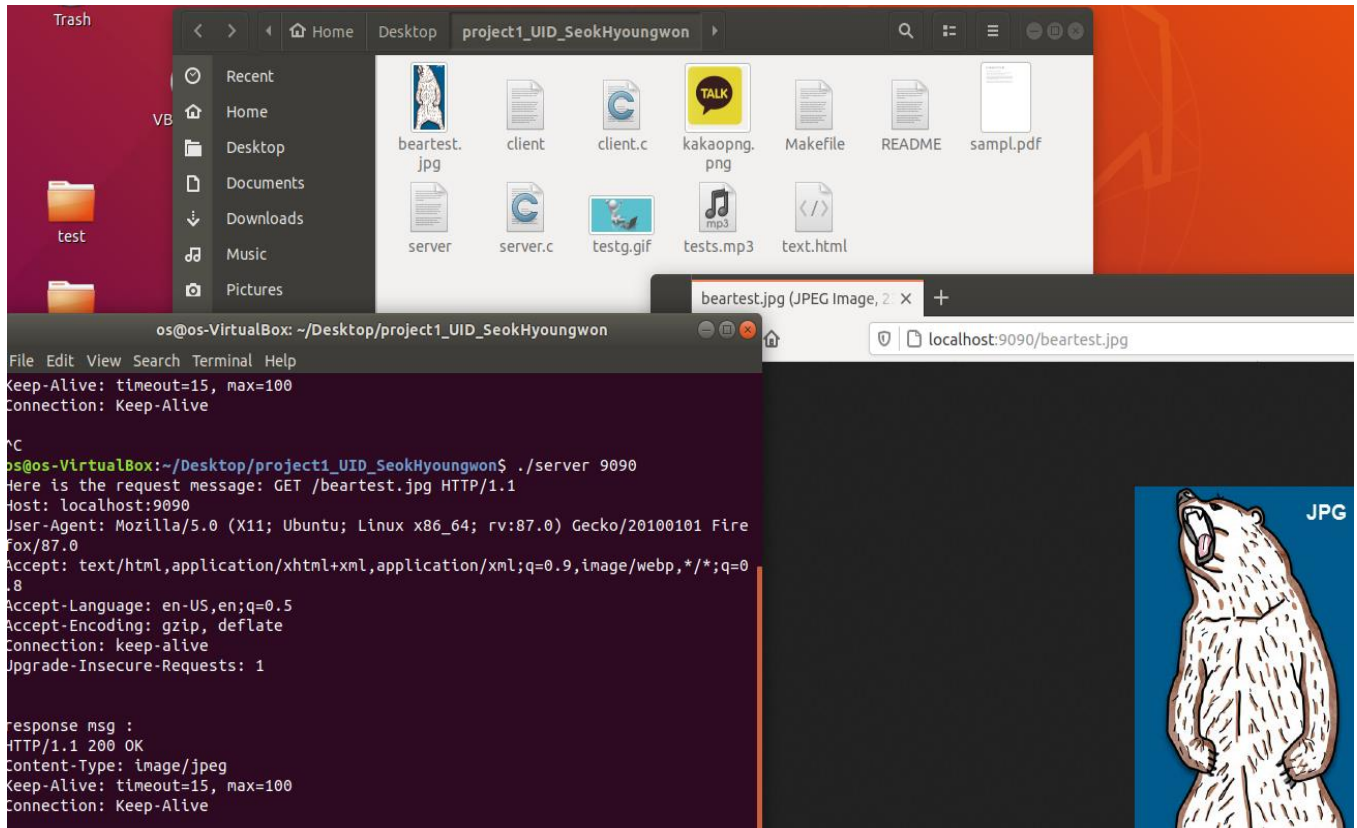
```
os@os-VirtualBox: ~/Desktop/project1_UID_SeokHyoungwon
File Edit View Search Terminal Help
os@os-VirtualBox:~/Desktop/project1_UID_SeokHyoungwon$ ./server 9090
Here is the request message: GET /text.html HTTP/1.1
Host: localhost:9090
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:87.0) Gecko/20100101 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

response msg :
HTTP/1.1 200 OK
Content-Type: text/html
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive

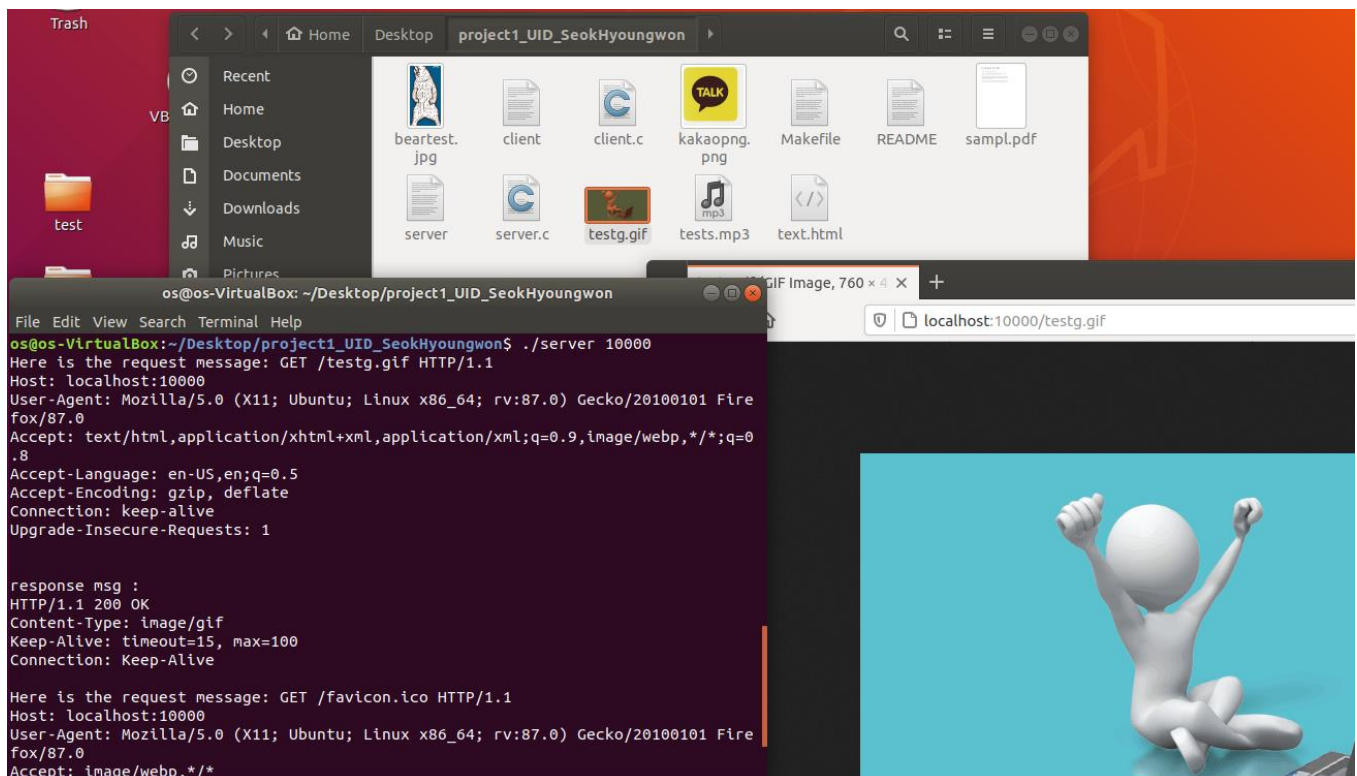
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>TEST SUCCESS</title>
</head>
<body>
  <h1>TEST SUCCESS</h1>
</body>
</html>
```

TEST SUCCESS

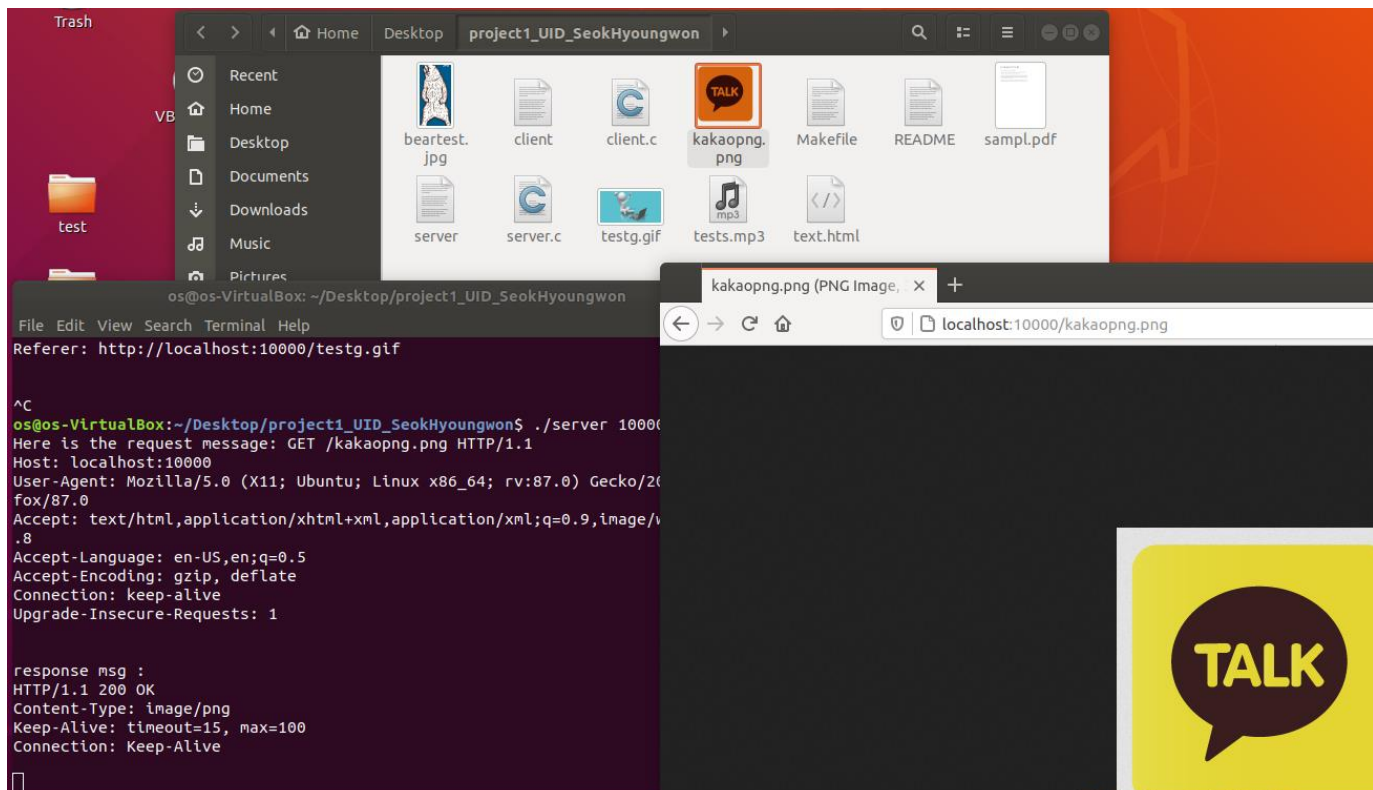
3. Jpeg (<http://localhost:9090/beartest.jpg>)



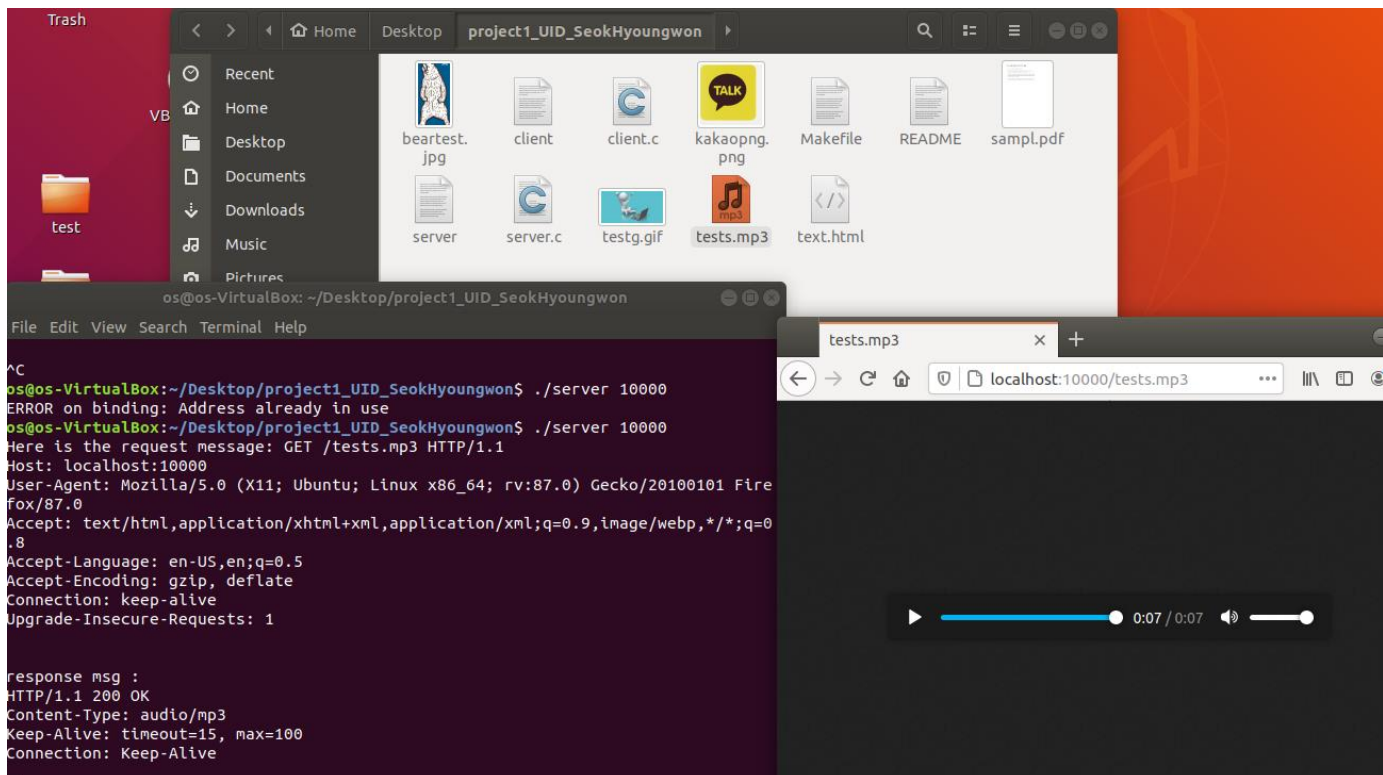
4. Gif (<http://localhost:10000/testg.gif>)



5. Png (<http://localhost:10000/kakaopng.png>)



6. mp3 (<http://localhost:10000/tests.mp3>)



7. pdf (<http://localhost:9090/sample.pdf>)

The screenshot shows a VirtualBox window with three main components:

- File Manager:** Displays the contents of the directory `~/Desktop/project1_UID_SeokHyoungwon`. Files include `beartest.jpg`, `client`, `client.c`, `kakaopng.png`, `Makefile`, `README`, `sample.pdf`, `server`, `server.c`, `testg.gif`, `tests.mp3`, and `text.html`.
- Terminal:** Shows the execution of a web server on port 9090. It receives a GET request for `/sample.pdf` from a Mozilla/5.0 Gecko browser. The server responds with `HTTP/1.1 200 OK` and `Content-Type: application/pdf`.
- Web Browser:** Displays the `sample.pdf` file from `localhost:9090/sample.pdf`. The PDF content is a simple text file titled "A Simple PDF File" with placeholder text.

8. not file (<http://localhost:10000/>)

The screenshot shows a VirtualBox window with three main components:

- File Manager:** Same as in the previous screenshot, showing the directory `~/Desktop/project1_UID_SeokHyoungwon` with various files.
- Terminal:** Shows the execution of a web server on port 10000. It receives a GET request for `/` from a Mozilla/5.0 Gecko browser. The server responds with `HTTP/1.1 200 OK` and `Content-Type: application/pdf`.
- Web Browser:** Displays the `localhost:10000/` page. The page content is `localhost:10000 404`, indicating a file not found error.