

Computer Network Project #1

2019061658 Kim Taehyeon

- Description of My Server's Design

1. Open the server

- Register the signal function named 'cleanExit' that calls exit function when the 'SIGTERM', 'SIGINT' signal is called
- Use the INADDR_ANY value on struct sockaddr_in -> sin_addr -> s_addr to allow connecting with any IP in the PC
- Get the custom port (1026~) from 'argv' variable and bind it

2. Get the request header whenever the client tries to connect

- The character array named 'buffer' stores the request data
- By using the string library (as like getline, append, and so on), get the lines on the variable named request_header
- While receiving the request message, the server understands the request method, file_path and the http_version from the first line of the request header message
- After receiving the request message, print it all
- Get the request messages within 10 (define BACKLOG 10) clients

3. Send the response header data (the function named 'send_response')

- The first line of request header is consist of ① Method, ② File Path, ③ HTTP Version
- If you finds that the ① Method is a 'GET', you have to send the response data with file data
- If you finds that the ② File Path is not valid, you have to send the response data with the status code 404 and send it
- If you find that the ② File Path is valid, make the string variable to be consist of the contents with the status code 200 (success) and the MIME (Multipurpose Internet Mail Extensions) type in Content-Type header
- Finally, send the response data with header and appended file data

4. Close the server whenever if you want it

- Difficulties that I faced and the Solution how solve them

1. The insufficient knowledge of the C++ language

- Situation : I didn't know how to develop the web server in C++.
- Solution : By using the presentation of the Computer Network lecture notes and the CourseWeb, I could learn how to use it and make it.

2. The insufficient knowledge of the web server

- Situation : I didn't know the principle how to operates the web server.
- Solution : With the RFC 1945 and searching internet, finally I could get the information the form of the header file and how to communicates on server between clients and server

- Sample Outputs

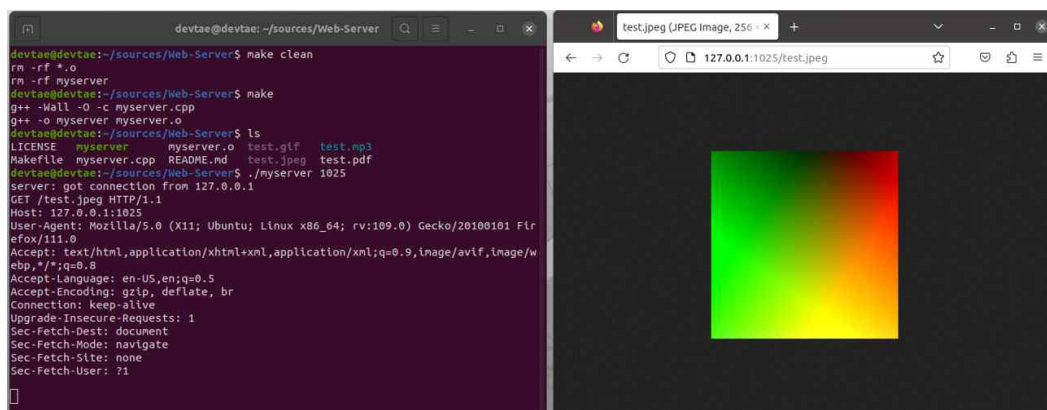
<Compile and Open the server>

```
devtae@devtae: ~/sources/Web-Server
devtae@devtae:~/sources/Web-Server$ make clean
rm -rf *.o
rm -rf myserver
devtae@devtae:~/sources/Web-Server$ make
g++ -Wall -O -c myserver.cpp
g++ -o myserver myserver.o
devtae@devtae:~/sources/Web-Server$ ls
LICENSE  myserver  myserver.o  test.gif  test.mp3
Makefile myserver.cpp  README.md  test.jpeg test.pdf
devtae@devtae:~/sources/Web-Server$ ./myserver 1025
```

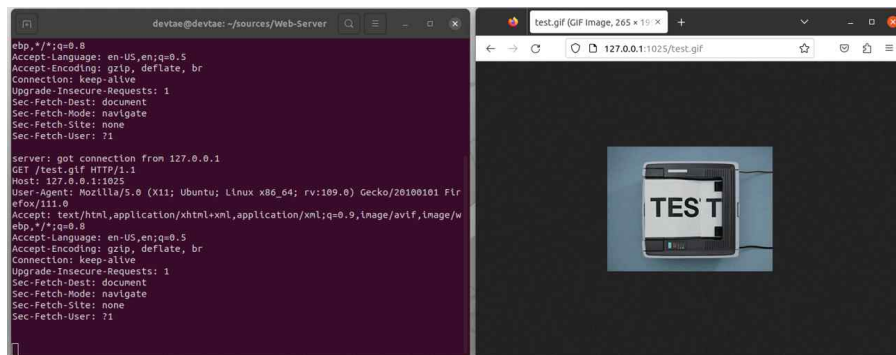
<Overview>

| Type | .jpeg | .gif | .mp3 | .pdf | External Connect |
|-------------------|-----------|-----------|-----------|-----------|------------------|
| Success / Failure | Success ! | Success ! | Success ! | Success ! | Success ! |

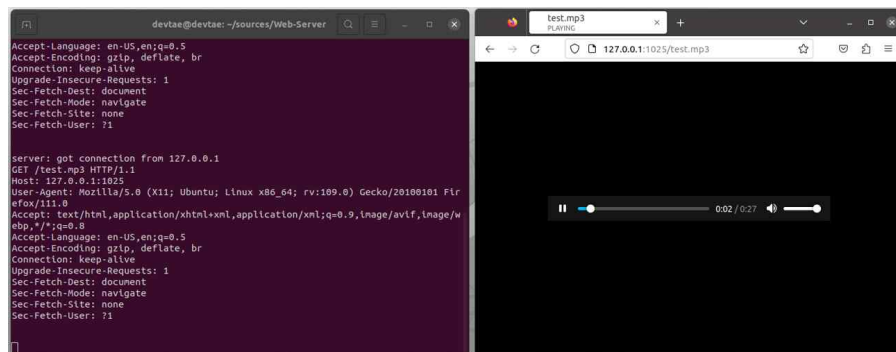
<Response the test.jpeg>



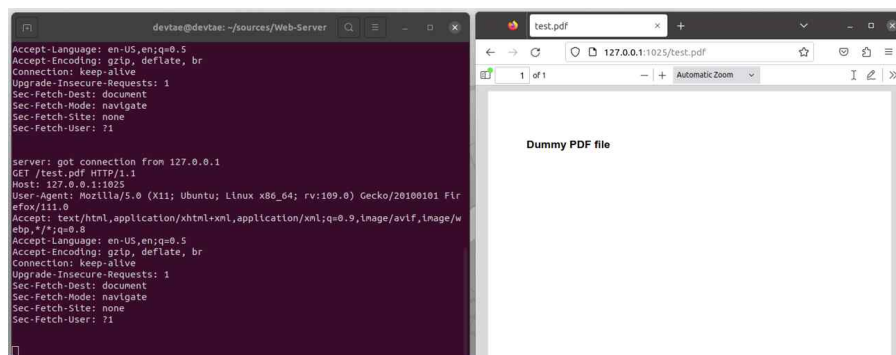
<Response the test.gif>



<Response the test.mp3>



<Response the test.pdf>



<Connect by external device>

