

CSE 3027: Computer Networks

Project1

Concurrent Web Server using BSD Sockets

과목명 : 컴퓨터 네트워크

담당교수 : 이석복 교수님

소속학과 : 소프트웨어학부

학번 : 2018044657

이름 : 박지호

실행 환경 : Ubuntu 18.04.5 | Firefox/87.0

서버 설계(동작) 방식

1. 서버 소켓 생성 : `serv_sock = socket()`
2. 서버의 IP주소와 port를 소켓에 할당 : `bind()`
3. client에서 응답해줄 수 있도록 대기상태로 만들기 : `listen()`
4. loop를 돌면서 client의 요청을 받아오고 응답
 - A. 어떤 client에서 요청이 오면 수락해줌 : `accept()`
 - B. Client에게 받은 request 메시지를 바탕으로 response를 작성하여 client에게 전송 : `http_handler()`
 - C. 클라이언트 소켓을 닫음 : `close(clnt_sock)`
5. loop가 종료되면 `log.out`과 서버 소켓을 닫음 : `close(serv_sock), close(log_fd)`

Part A : Request Message를 화면에 출력하는 웹 서버 구현

```
jhpark@jhpark-VirtualBox:~/Desktop/computer-network-project1$ cat log.out
===== REQUEST
GET / HTTP/1.1
Host: localhost:3000
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
Cache-Control: max-age=0
```

Client에서 Server에 전달한 Request message를 `log.out`이라는 텍스트 파일에 기록해두었다.

```
GET / HTTP/1.1
Host: localhost:3000
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
Cache-Control: max-age=0
```

1번째 줄(**GET / HTTP/1.1**): 세 가지 요소로 이루어져있다.

“**GET**”은 HTTP 메서드로, 리소스를 클라이언트로 가져다 달라는 것을 뜻한다.

“/”는 도메인의 절대 경로를 나타내며, “/”경로에 해당하는 index.html 을 요청한다.

“**HTTP/1.1**”은 HTTP의 버전이 들어간다. HTTP/1.1형식을 사용한다는 것을 알 수 있다.

2번째 줄(**Host**): 요청이 전송되는 target의 host url이다. 현재 localhost:3000에 요청하고 있다.

3번째 줄(**User-Agent**): 요청을 보내는 클라이언트의 정보(사용하는 웹 브라우저 버전, OS 등)를 나타내고 있다. Mozilla/5.0기반을 사용하며, 사용자의 플랫폼은 X11; Ubuntu; Linux x86_64; rv:87.0, 그리고 Gecko엔진을 사용하는 Firefox/87.0버전을 사용한다는 것을 알 수 있다.

4번째 줄(**Accept**): 해당 요청이 받을 수 있는 응답 타입이다.

“<**MIME_type**>/<**MIME_subtype**>;q=” 형식을 가지며 q는 가중치이며 선호도를 나타낸다. text/html, application/xhtml+xml, application/xml;q=0.9, image/webp, */*;q=0.8이며 */*는 모든 MIME(Multipurpose Internet Mail Extensions) 타입을 나타낸다.

5번째 줄(**Accept-Language**): 어떤 언어를 클라이언트가 이해할 수 있는지, 지역 설정 중 어느 것이 더 선호되는지를 알려준다. 여기서 언어란 프로그래밍 언어가 아니라 영어같은 자연 언어를 말한다.

6번째 줄(**Accept-Encoding**): 클라이언트 자신이 원하는 문자 인코딩 방식을 정한다. gzip, deflate 방식으로 인코딩을 한다.

7번째 줄(**Connection**): 클라이언트와 서버 간 연결에 대한 옵션 설정을

한다. keep-alive는 TCP 커넥션을 계속 유지한다는 뜻이다.

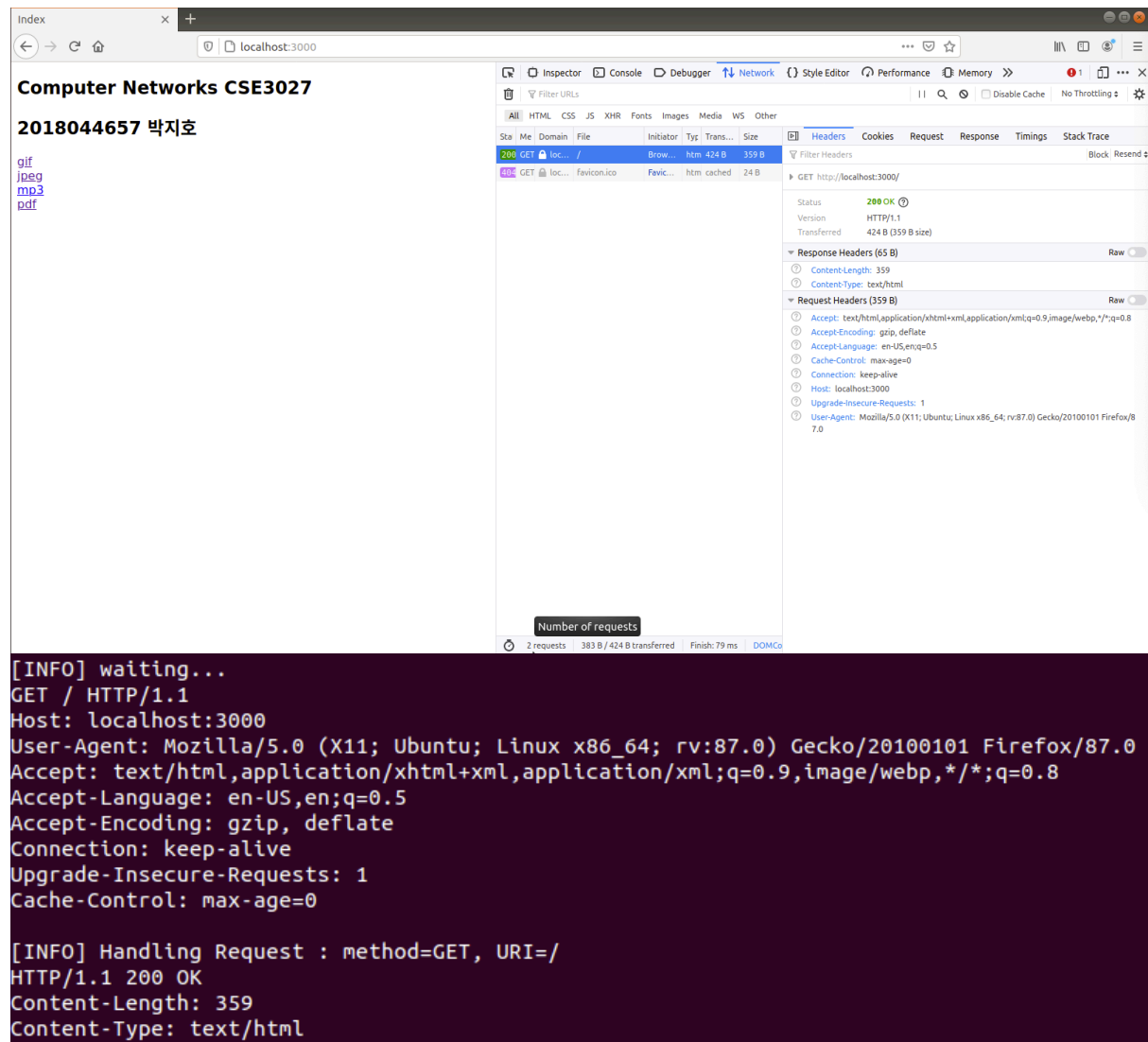
8번째 줄(**Upgrade-Insecure-Requests**): 클라이언트가 서버에게 Secure버전을 지원해달라고 요청하며, 서버는 안전한 버전으로 redirect할 수 있다. IE는 지원하지 않는다.

9번째 줄(**Cache-Control**): Caching 정책을 정의할 수 있다. max-age는 캐시 유효 시간을 나타내며 초단위로 설정한다. 해당 시간이 지나면 만료된 것으로 여겨진다. 0이므로 0초 후에 만료됐다고 여긴다.

Next Page ->

Part B : server가 client에게 받은 request를 response할 수 있도록 구현

Index.html



The screenshot shows a web browser window with the address bar set to `localhost:3000`. The page content is as follows:

```
Computer Networks CSE3027
2018044657 박지호
gif
jpeg
mp3
pdf
```

The browser's developer tools are open to the Network tab, showing a single request to `http://localhost:3000/`. The request is a `GET` method with a status of `200 OK`. The response headers are:

- `Content-Length: 359`
- `Content-Type: text/html`

The request headers include:

- `Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8`
- `Accept-Encoding: gzip, deflate`
- `Accept-Language: en-US,en;q=0.5`
- `Cache-Control: max-age=0`
- `Connection: keep-alive`
- `Host: localhost:3000`
- `Upgrade-Insecure-Requests: 1`
- `User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:87.0) Gecko/20100101 Firefox/87.0`

Below the browser window, a terminal window displays the raw request and response details:

```
[INFO] waiting...
GET / HTTP/1.1
Host: localhost:3000
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
Cache-Control: max-age=0

[INFO] Handling Request : method=GET, URI=/
HTTP/1.1 200 OK
Content-Length: 359
Content-Type: text/html
```

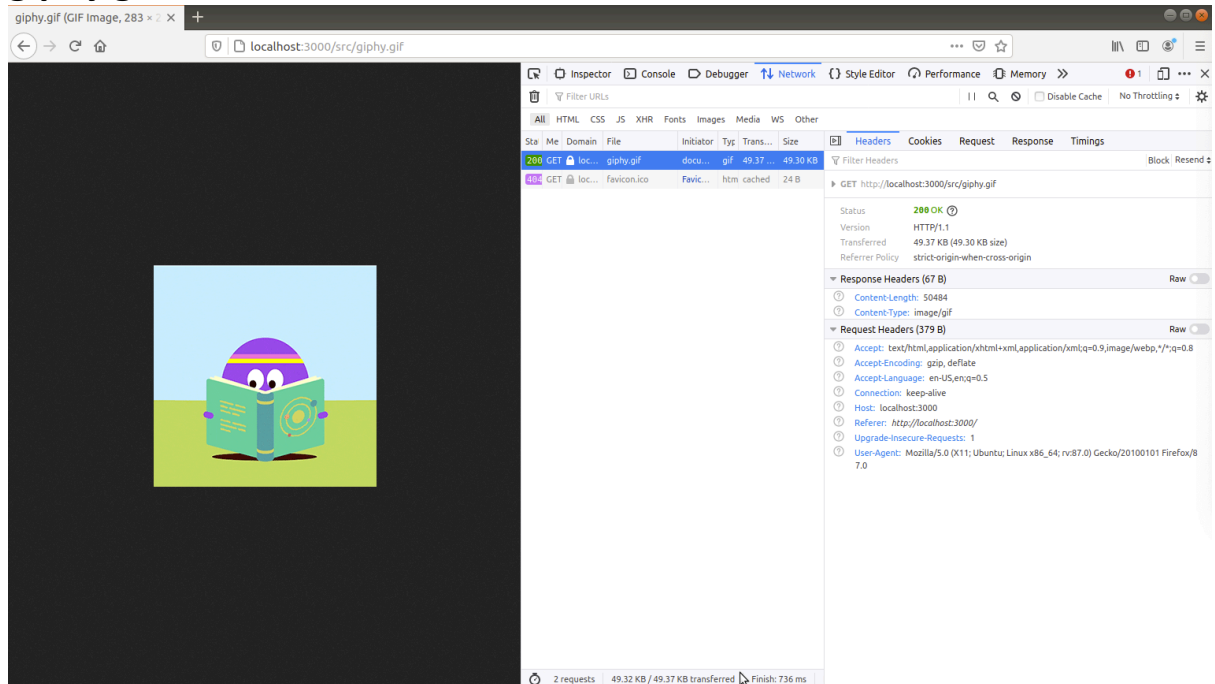
공백 위의 request message에서 **HTTP/1.1** 버전에서 **GET** 방식으로 `/` URI에 요청한 모습이다.

요청한 도메인은 **localhost:3000**이라는 것을 알 수 있다.

마지막 3줄은 response header이다.

200 OK 응답을 받았고, 본문 크기는 byte단위로 나타냈는데 **359 byte**인 것을 알 수 있다. 콘텐츠의 타입(MIME)은 **text/html** 이다.

giphy.gif



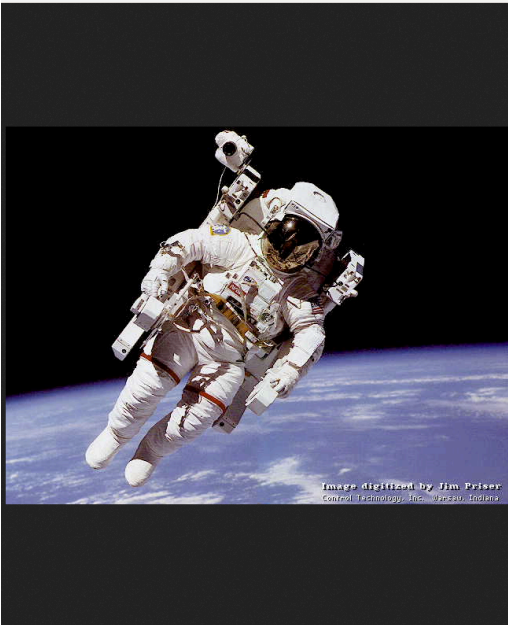
```
[INFO] waiting...
GET /src/giphy.gif HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://localhost:3000/
Upgrade-Insecure-Requests: 1

[INFO] Handling Request : method=GET, URI=/src/giphy.gif
HTTP/1.1 200 OK
Content-Length: 50484
Content-Type: image/gif
```

request message에서 **HTTP/1.1** 버전에서 **GET** 방식으로 **localhost:3000**도메인에 **‘/src/giphy.gif’** URI에 요청한 모습이다.

200 OK 응답을 받았고, 본문 크기는 byte단위로 나타냈는데 **50484 byte** 인 것을 알 수 있다. 컨텐츠의 타입(MIME)은 **image/gif** 이다.

space.jpeg



The screenshot shows a web browser displaying an image of an astronaut in space. The browser's address bar shows the URL `localhost:3000/src/space.jpeg`. The network inspector is open, showing the request details for `space.jpeg`. The request is a `GET` method with a status of `200 OK`. The response headers indicate a `Content-Length` of 61079 and a `Content-Type` of `image/jpeg`. The request headers show the browser's user agent and other details.

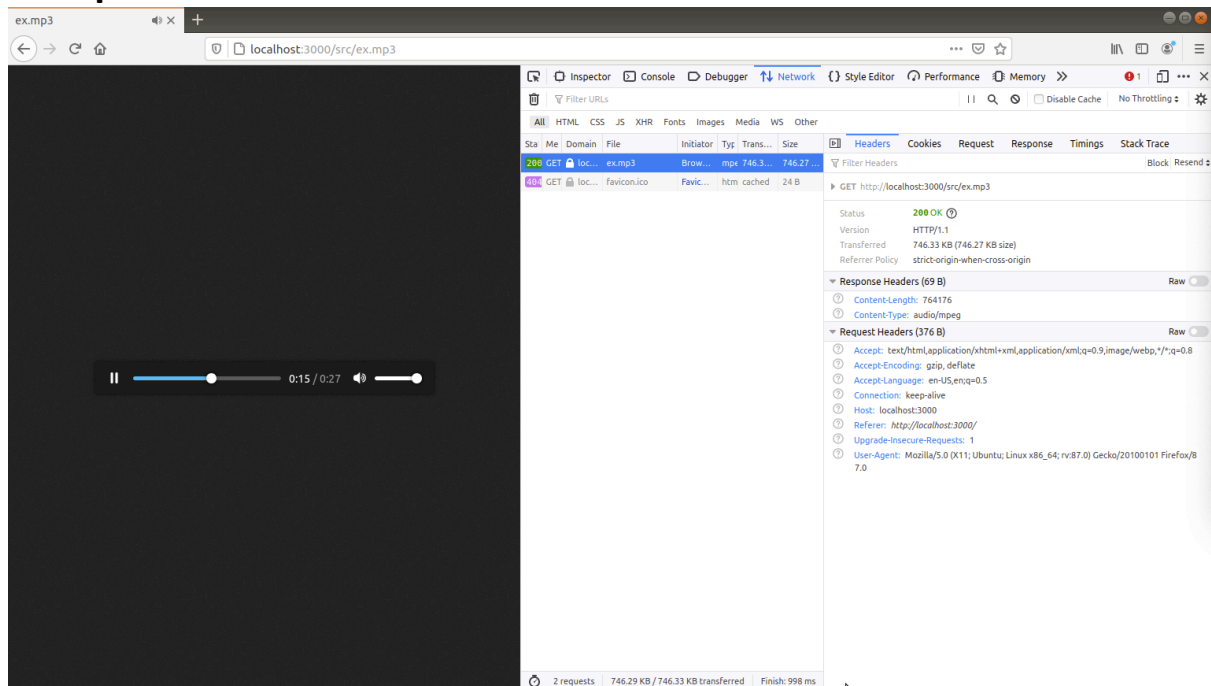
```
[INFO] waiting...
GET /src/space.jpeg HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:3000/
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

[INFO] Handling Request : method=GET, URI=/src/space.jpeg
HTTP/1.1 200 OK
Content-Length: 61079
Content-Type: image/jpeg
```

request message에서 **HTTP/1.1** 버전에서 **GET** 방식으로 **localhost:3000**도메인에 `/src/space.jpeg` URI에 요청한 모습이다.

200 OK 응답을 받았고, 본문 크기는 byte단위로 나타냈는데 **61079 byte** 인 것을 알 수 있다. 콘텐츠의 타입(MIME)은 **image/jpeg** 이다.

ex.mp3



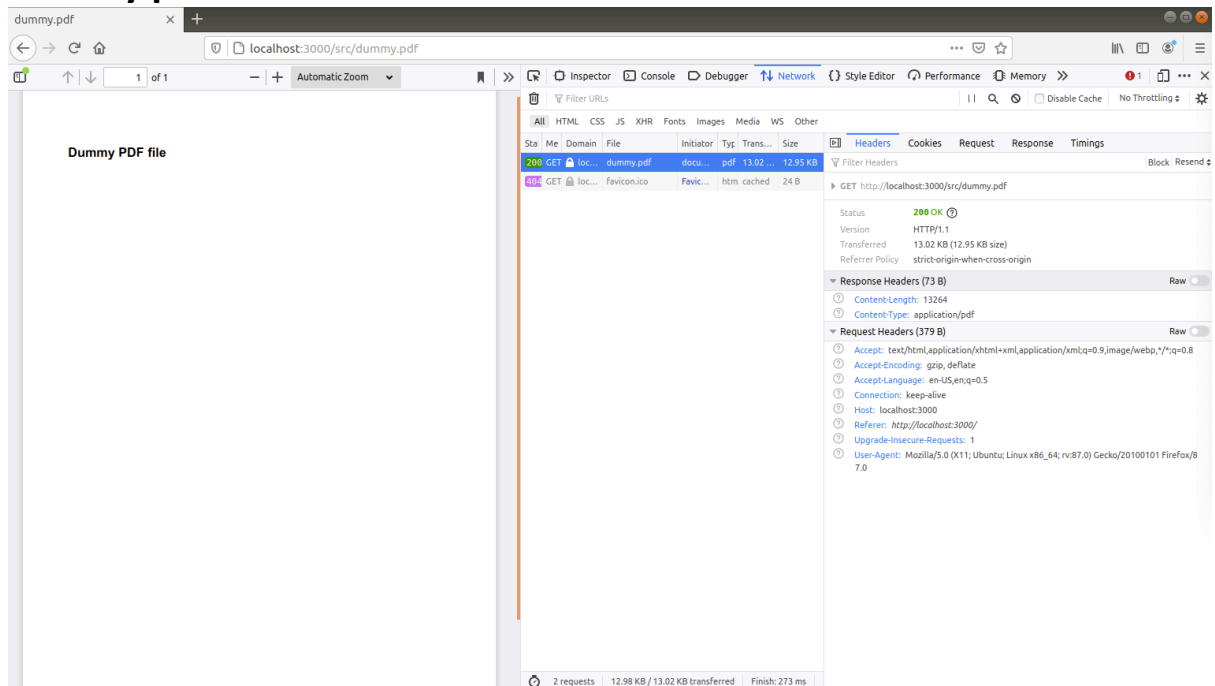
```
[INFO] waiting...
GET /src/ex.mp3 HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:3000/
Connection: keep-alive
Upgrade-Insecure-Requests: 1

[INFO] Handling Request : method=GET, URI=/src/ex.mp3
HTTP/1.1 200 OK
Content-Length: 764176
Content-Type: audio/mpeg
```

request message에서 **HTTP/1.1** 버전에서 **GET** 방식으로 **localhost:3000**도메인에 **‘/src/ex.mp3’** URI에 요청한 모습이다.

200 OK 응답을 받았고, 본문 크기는 byte단위로 나타냈는데 **764716** byte인 것을 알 수 있다. 콘텐츠의 타입(MIME)은 **audio/mpeg** 이다.

dummy.pdf



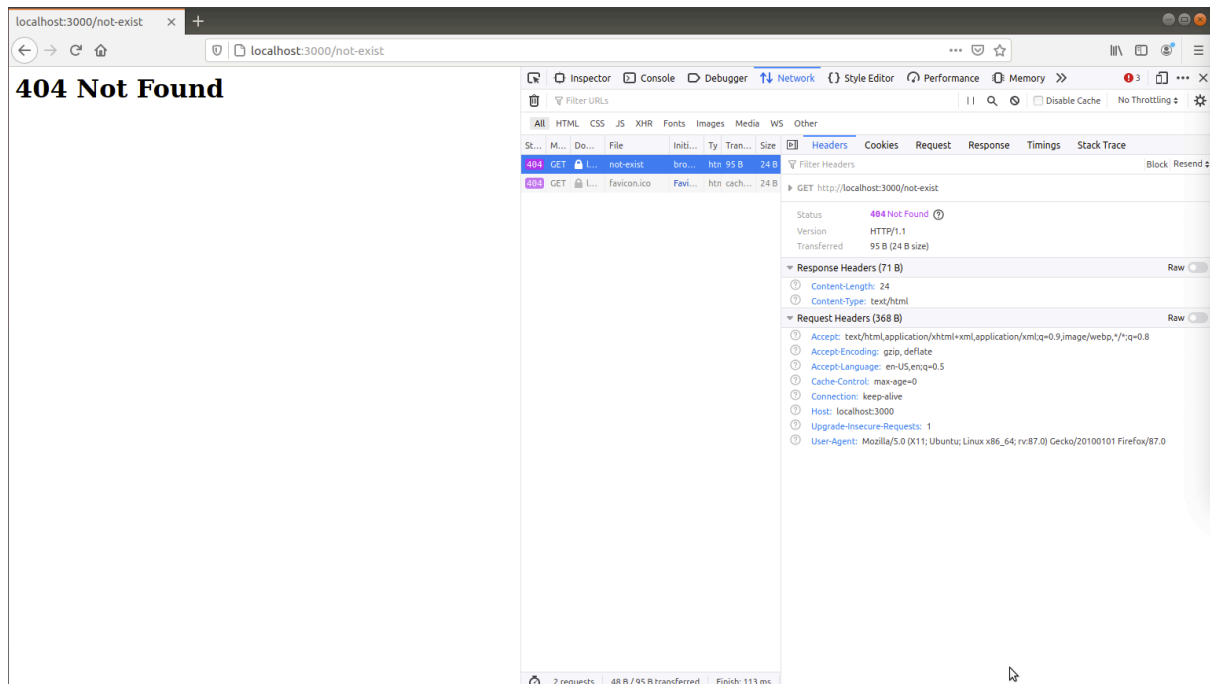
```
[INFO] waiting...
GET /src/dummy.pdf HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:87.0) Gecko/20100101 Firefox/87.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://localhost:3000/
Upgrade-Insecure-Requests: 1

[INFO] Handling Request : method=GET, URI=/src/dummy.pdf
HTTP/1.1 200 OK
Content-Length: 13264
Content-Type: application/pdf
```

request message에서 **HTTP/1.1** 버전에서 **GET** 방식으로 **localhost:3000**도메인에 **‘/src/dummy.gif’** URI에 요청한 모습이다.

200 OK 응답을 받았고, 본문 크기는 byte단위로 나타냈는데 **13264 byte** 인 것을 알 수 있다. 컨텐츠의 타입(MIME)은 **application/pdf** 이다.

404 Not Found



```
[INFO] waiting...
GET /not-exist HTTP/1.1
Host: localhost:3000
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
Cache-Control: max-age=0

[INFO] Handling Request : method=GET, URI=/not-exist
HTTP/1.1 404 Not Found
Content-Length: 24
Content-Type: text/html
```

request message에서 **HTTP/1.1** 버전에서 **GET** 방식으로 **localhost:3000**도메인에 **'/not-exist'** URI에 요청한 모습이다.

404 Not Found 응답을 받았고, 본문 크기는 byte단위로 나타냈는데 **24 byte**인 것을 알 수 있다. 컨텐츠의 타입(MIME)은 **text/html** 이다.

Next page ->

과제 중 어려움과 해결

c언어 소켓 프로그래밍이라는 단어는 많이 들어보았지만 개념이나 구현 방법에 대해 전혀 아는게 없어 많이 고생했었다.

이곳 저곳 구글링을 하며 c 웹 서버의 방식을 이해하고 구현에 성공하였다.

Request나 Response를 가져오고 작성할 때 어떻게 해야할지 몰라 많이 헤맸었다. 요청을 어떻게 받아오고 응답 필드에 어떤 내용을 넣어야 할지 몰라 막막했다.

MDN Web Docs같은 사이트에서 이러한 점을 해결하기 위해 충분히 공부해 둔 뒤, Client가 요청하고 Server가 응답하는데 성공하였다.