

REPORT



한양대학교 ERICA
Education Research Industry Cluster @ Ansan



과목명 | 컴퓨터네트워크

담당교수 | 이석복 교수님

학과 | 소프트웨어학부 컴퓨터전공

학년 | 3 학년

학번 | 2017012406

이름 | 이재희

제출일 | 2021-04-20

목 차

1. 웹 서버 구조	4
1.1 웹 서버 구성도	4
1.2 웹 서버 제작에 사용된 함수 흐름도	4
2. 구현 상 어려웠던 점과 극복 방법	5
2.1 구현 상 어려웠던 점	5
2.2 문제 해결	5
3. 웹 서버 실행 결과	6
3.1 웹 서버 실행 예시	6

그림 목차

그림 1-1 Web server 동작 구조.....	4
그림 1-2 web server 에 사용된 함수 흐름도	4
그림 3-1 기본 index html 파일 요청을 위한 HTTP request.....	6
그림 3-2 기본 index html 파일 실행 모습.....	6
그림 3-3 클라이언트에게 전송되는 HTTP Response 내용.....	7
그림 3-4 index html 파일 실행 시 개발자 모드 모습.....	7
그림 3-5 이미지 파일 요청을 위한 http request.....	7
그림 3-6 이미지 파일 출력.....	8
그림 3-7 pdf 파일 요청을 위한 http request.....	8
그림 3-8 pdf 파일 실행 모습	9

1. 웹 서버 구조

1.1 웹 서버 구성도

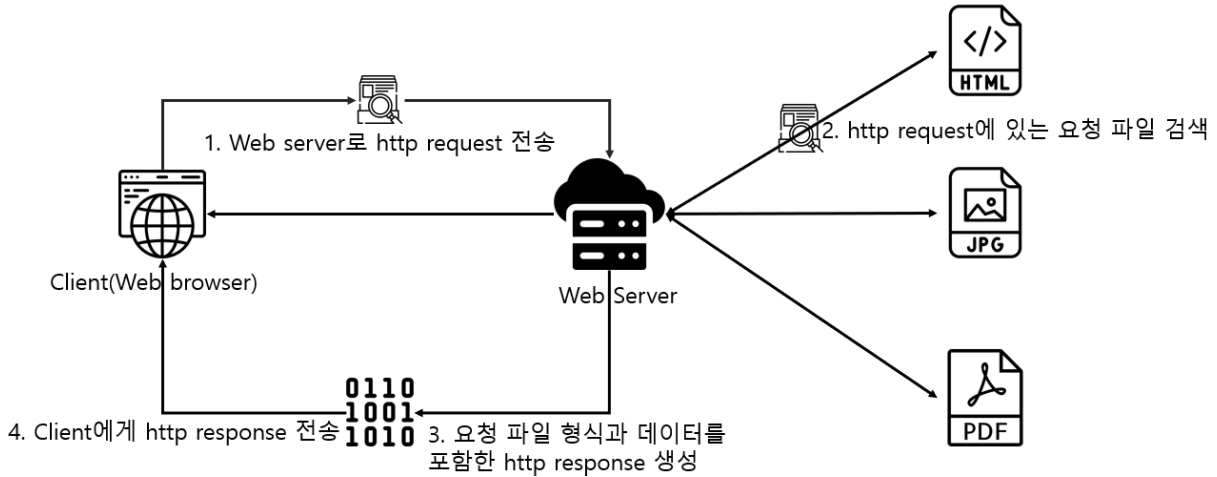


그림 1-1 Web server 동작 구조

1.2 웹 서버 제작에 사용된 함수 흐름도

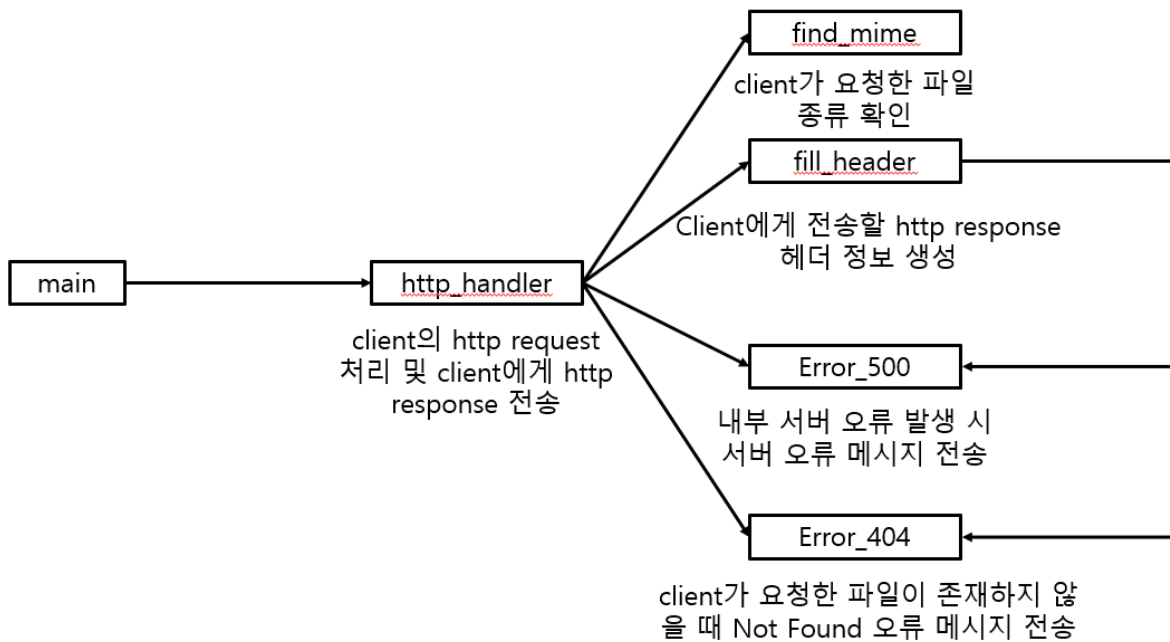


그림 1-2 web server 에 사용된 함수 흐름도

1. 웹 서버의 main 함수를 실행하여 서버의 소켓을 열고 서버 주소와 바인딩하여 클라이언트의 연결 요청을 기다린다.
2. 연결 요청이 들어오면 자식 프로세스를 생성하여 자식 프로세스에서 http_handler 함수를 실행한다.

3. 호출된 `http_handler` 함수를 통해 client로부터 받은 `http request` 내용을 터미널에 출력한다.
4. `http request` 에서 `request method` 와 `uri` 를 파싱하여 요청 파일을 찾기 위한 경로를 설정한다.
5. `uri` 를 기준으로 요청 파일을 검색한다. 이 때 `uri` 가 '/'으로 설정되어 있다면 기본 `html` 파일인 `index.html` 파일을 검색한다.
6. 찾은 파일의 크기, 종류, 데이터가 있는 내용을 포함하여 `http response` 를 생성한다.
7. 생성한 `http response` 를 client에게 전송한다.

2. 구현 상 어려웠던 점과 극복 방법

2.1 구현 상 어려웠던 점

웹 브라우저에서 웹 서버의 IP 주소와 포트 번호 입력 후 요청할 파일의 경로를 적어서 웹 서버에게 보내면 요청한 파일의 데이터가 원본 그대로 출력돼야 하지만 파일 데이터가 지워져서 출력된 경우, `http response` 를 받은 후 연결 종료를 하였음에도 소켓이 완전히 종료되지 않아 해당 포트 번호로 다시 연결할 수 없는 경우도 있었다. 그리고 `html` 파일을 요청하는 `html request` 발송 시 `html` 파일 외에 `favicon.ico` 라는 파일에 대한 `http request` 요청도 자동으로 같이 발송이 되는 경우도 있었다.

2.2 문제 해결

소스 코드를 살펴보고 각 변수에 들어있는 데이터 값 확인을 위해 `printf` 문을 찍어보면서 프로그램 실행 흐름을 체크하던 중 지정 메모리 주소의 데이터를 0으로 초기화하는 `bzero` 함수를 실행하는 부분이 있는데 이것을 `http response` 생성 후에 실행하여 데이터가 초기화되는 문제를 발견하고 `bzero` 함수를 `http response` 생성 전으로 옮겨서 데이터가 초기화되는 문제를 해결할 수 있었다. 그리고 소켓 종료 문제의 경우 `http` 처리를 담당하는 자식 프로세스가 제대로 종료되지 않은 채로 부모 프로세스가 종료되어 고아 프로세스로 남은 자식 프로세스가 계속 실행되고 있어 소켓이 계속 바인딩되는 문제로 확인되어 자식 프로세스가 제대로 종료될 수 있도록 `exit` 함수를 추가하여 처리하였다. 하지만 `favicon.ico` 호출 문제에 관해서는 아직 해결점을 찾지 못하였다.

3. 웹 서버 실행 결과

3.1 웹 서버 실행 예시

```
[HTTP Request]
GET / HTTP/1.1
Host: localhost:10100
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:79.0) Gecko/20100101 Firefox/79.0
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
```

그림 3-1 기본 index html 파일 요청을 위한 HTTP request

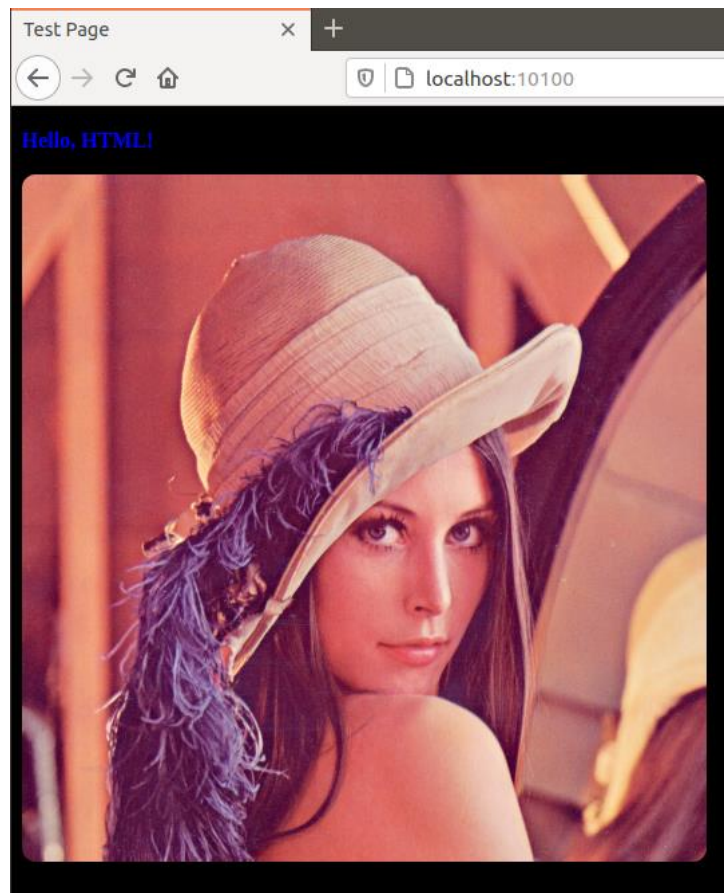


그림 3-2 기본 index html 파일 실행 모습

```
[HTTP Response]
<!doctype html>
<html>
  <head>
    <title>Test Page</title>
    <link rel="stylesheet" href="index.css">
  </head>
  <body>
    <p>
      <strong>Hello, HTML!</strong>
    </p>

    <script src="index.js"></script>
  </body>
</html>
```

그림 3-3 클라이언트에게 전송되는 HTTP Response 내용

Status	Met...	Domain	File	Initiator	Type	Transferred	Size
200	GET	localhost:1...	/	document	html	368 B	303 B
200	GET	localhost:1...	index.css	stylesheet	css	185 B	121 B
200	GET	localhost:1...	index.js	script	js	130 B	60 B
200	GET	localhost:1...	lena.png	img	png	462.79 KB	462....
	GET	localhost:1...	favicon.ico	FaviconLoade...		0 B	0 B

그림 3-4 index html 파일 실행 시 개발자 모드 모습

```
[HTTP Request]
GET /img/lena.png HTTP/1.1
Host: localhost:10100
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:79.0) Gecko/20100101 Firefox/79.0
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
```

그림 3-5 이미지 파일 요청을 위한 http request

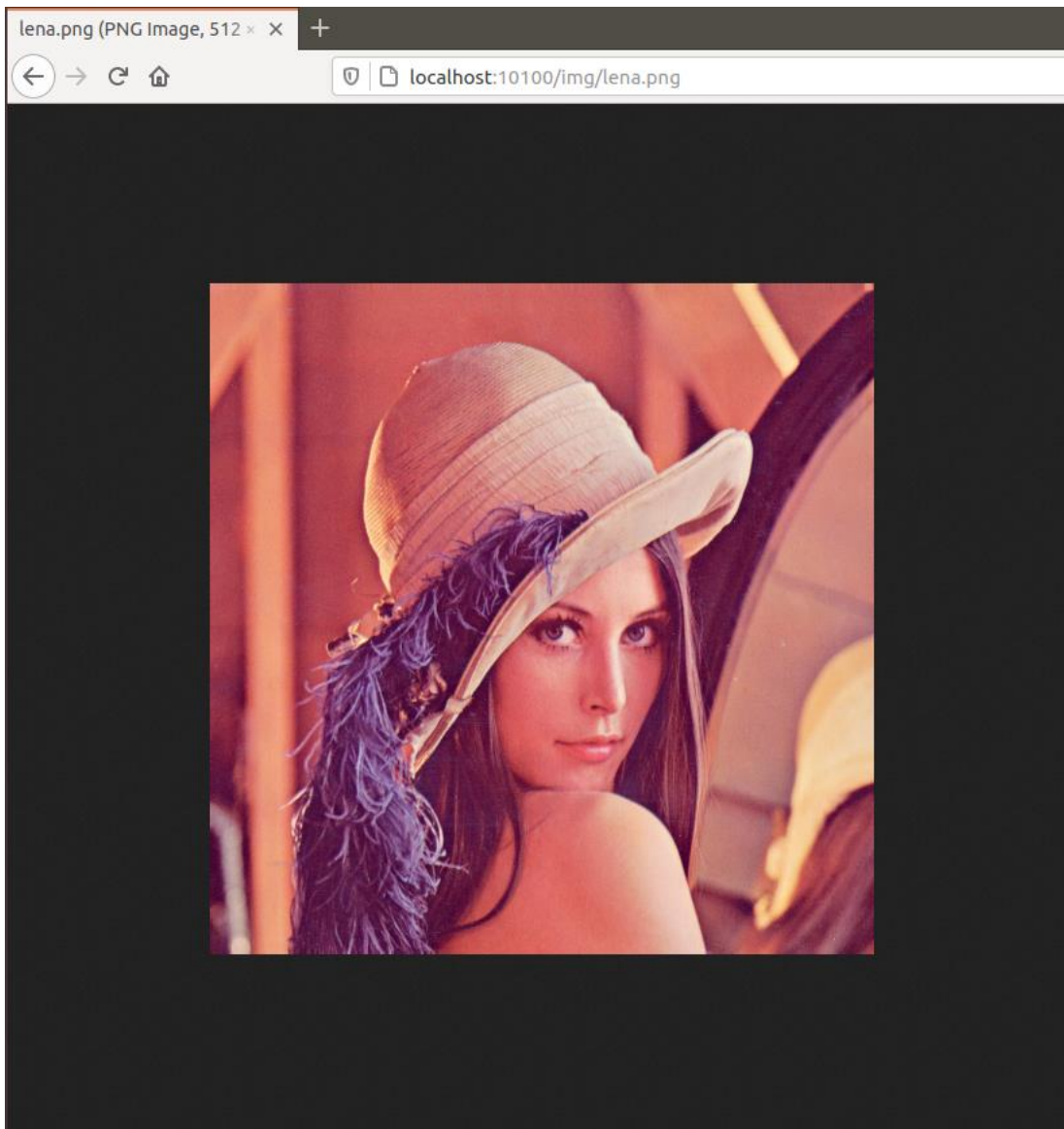


그림 3-6 이미지 파일 출력

```
[HTTP Request]
GET /project1.pdf HTTP/1.1
Host: localhost:10100
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:79.0) Gecko/20100101 Firefox/79.0
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
```

그림 3-7 pdf 파일 요청을 위한 http request

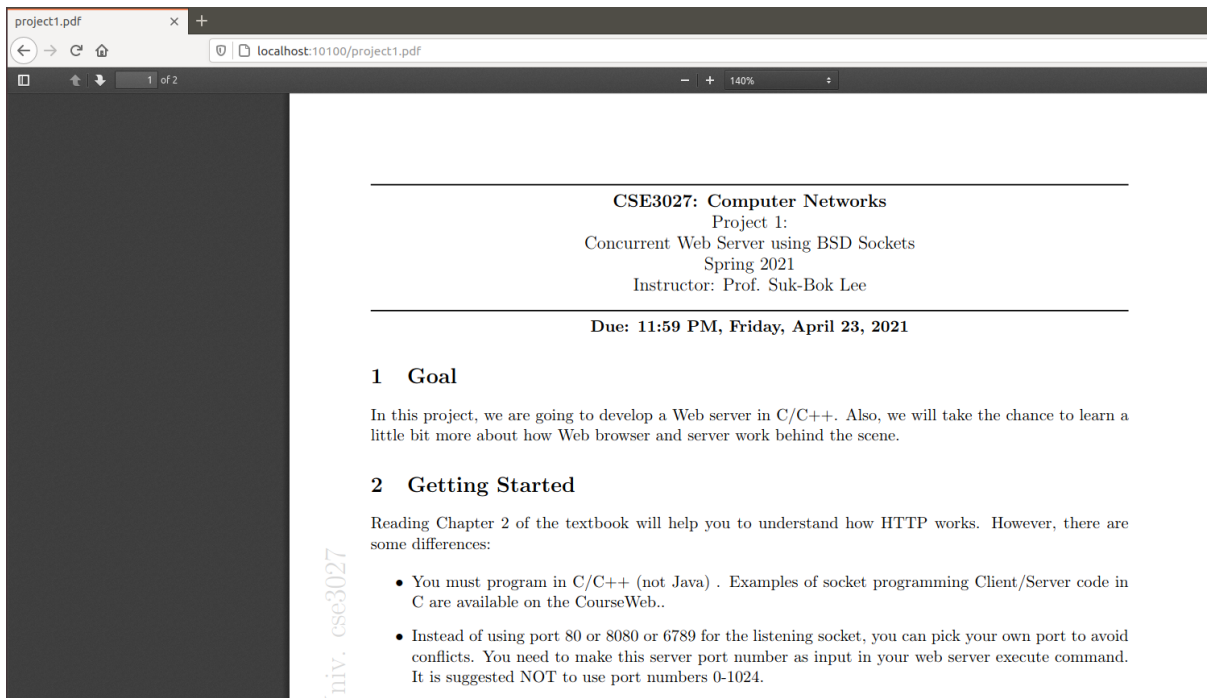


그림 3-8 pdf 파일 실행 모습