

# 컴퓨터네트워크

## HW #2 (30 Points)

Due date : 2024/5/7 (eCampus)

제출물 : HW2\_학번.c, HW2\_학번.cpp, HW2\_학번.java, HW2\_학번.py (하나의 파일에 모든 것을 다 구현함)

이 과제는 간단한 웹서버를 구현하는 것이다. 이 웹서버는 HTTP Request와 HTTP response 메시지를 잘 이해하고 처리해야 구현할 수 있다.

- .html 파일 처리가 가능하다.
- .jpg 파일 처리가 가능하다.

아래는 실행 예이다. 소스를 컴파일해서 hw2라는 실행파일을 만들어 냈다.

그리고 웹서버 프로그램인 “hw2”를 실행하는데, 파라미터로 포트번호를 제공한다. 이 예에서는 포트번호가 10000이다. 프로그램은 학번과 이름을 출력한다.

```
sanghwan@PC: ~/dbox/classes241/network/homework/hw242_webserver
sanghwan@PC-:~/dbox/classes241/network/homework/hw242_webserver$ ls -l *.html *.jpg
-rwxrwxrwx 1 sanghwan sanghwan 44 Apr 22 2016 a.html
-rwxrwxrwx 1 sanghwan sanghwan 44 Apr 22 2016 b.html
-rwxrwxrwx 1 sanghwan sanghwan 10818 Apr 22 2016 biga.html
-rwxrwxrwx 1 sanghwan sanghwan 142740 Apr 11 2022 palladio.jpg
sanghwan@PC-:~/dbox/classes241/network/homework/hw242_webserver$ gcc -o hw2 hw2sol.c
sanghwan@PC-:~/dbox/classes241/network/homework/hw242_webserver$ ./hw2 10000
Student ID : 2000000
Name : Sanghwan Lee
```

이제 웹브라우저 (구글 크롬 등 상용 웹브라우저를 사용함)를 실행해서 위 .html 파일과 .jpg 파일 중 하나를 요청한다.

이 때 URL은 아래와 같다. 같은 호스트에서 요청하기 때문에 도메인 네임은 localhost를 사용하였다.

**http://localhost:10000/palladio.jpg**

브라우저에 위 URL을 입력하고 엔터를 치면 palladio.jpg 파일을 서버로 요청하게 된다.  
hw2는 이 브라우저가 요청한 HTTP Request 메시지의 첫번째 라인과 User-Agent 정보를 출력한다.

또한 헤더 필드의 수를 같이 출력한다.

**참고. 혹시 연결이 안될 경우 방화벽에서 10000 번 포트번호를 허용하도록 변경하세요.**

그리고, 해당하는 파일을 찾아서 HTTP Response 메시지로 만들어서 브라우저로 보낸다.

hw2가 사용하는 HTTP Response 메시지는 아래와 같다.

**HTTP/1.0 200 OK**

Connection: close

Content-Length: **142740**

Content-Type: **image/jpeg**

요청 파일이 html 일 경우는 아래와 같다.

**HTTP/1.0 200 OK**

Connection: close

Content-Length: **20**

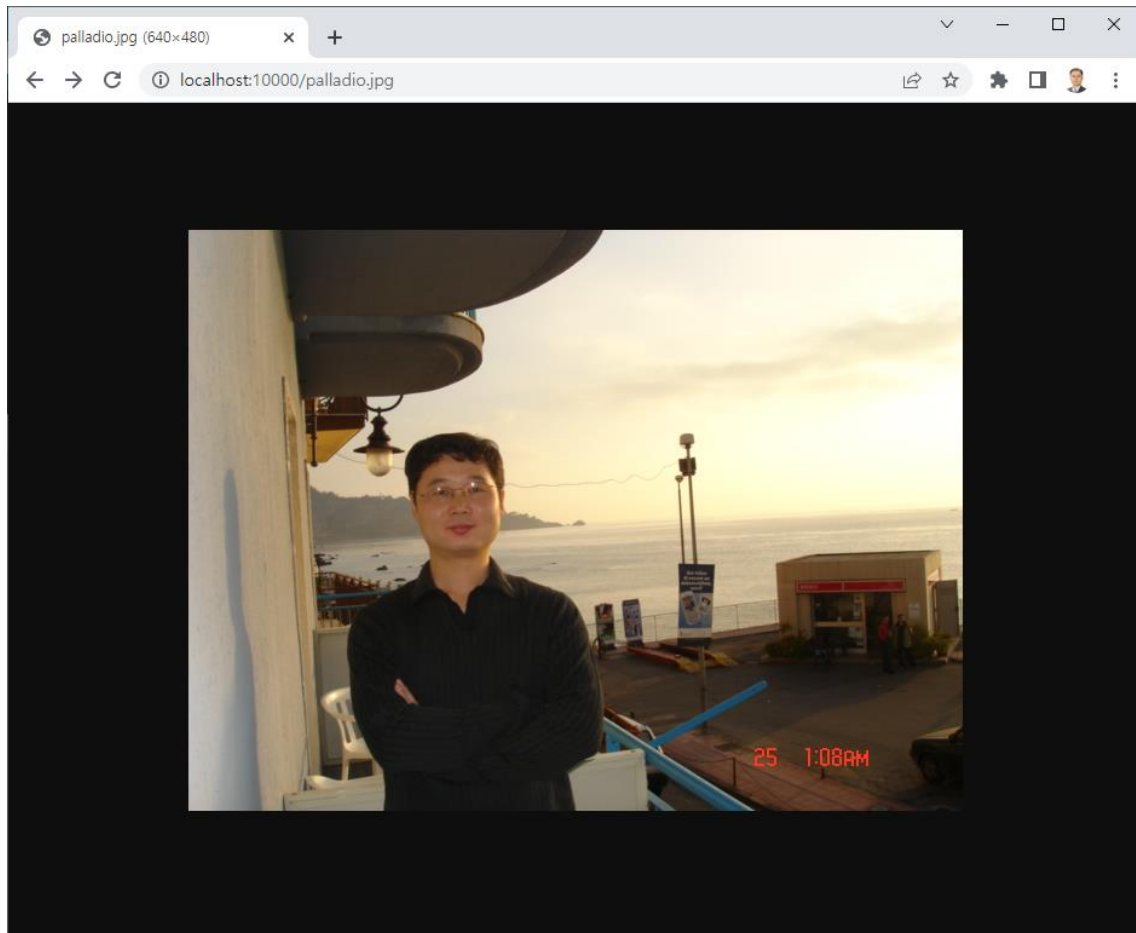
Content-Type: **text/html**

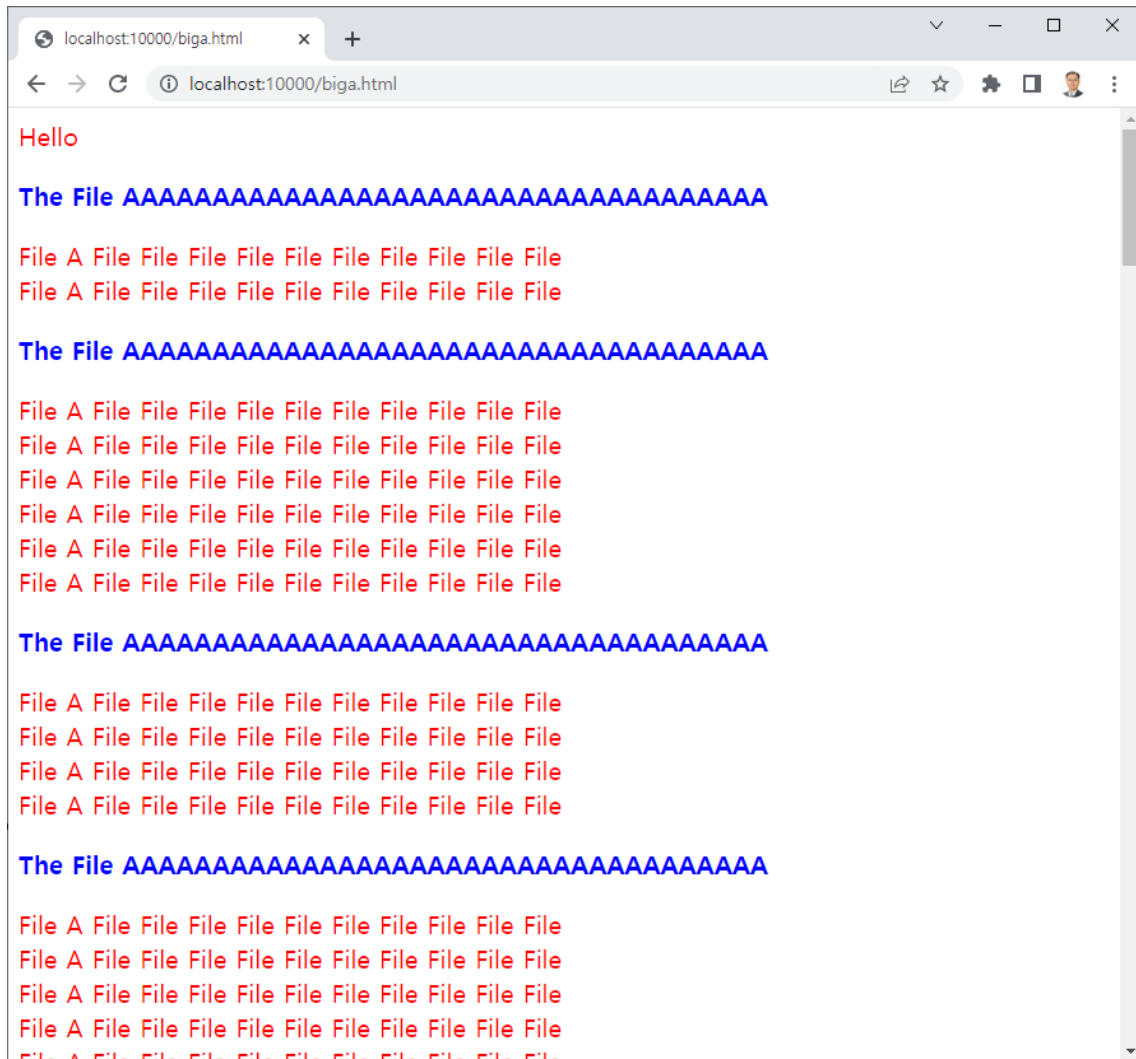
Content-Length 헤더 필드에는 파일의 크기를 입력한다.

다 보내고 나면 실제 전송한 바이트 수와 파일의 바이트 수를 출력한다. 이 두 값은 같아야 정상이다. 아래 화면은 이러한 내용을 보여준다.

```
sanghwan@PC: ~/dbox/classes241/network/homework/hw242_webserver
-rwxrwxrwx 1 sanghwan sanghwan 44 Apr 22 2016 a.html
-rwxrwxrwx 1 sanghwan sanghwan 44 Apr 22 2016 b.html
-rwxrwxrwx 1 sanghwan sanghwan 10818 Apr 22 2016 biga.html
-rwxrwxrwx 1 sanghwan sanghwan 142740 Apr 11 2022 palladio.jpg
sanghwan@PC:~/dbox/classes241/network/homework/hw242_webserver$ gcc -o hw2 hw2sol.c
sanghwan@PC:~/dbox/classes241/network/homework/hw242_webserver$ ./hw2 10000
Student ID : 2000000
Name : Sanghwan Lee
Connection : Host IP 127.0.0.1, Port 64514, socket 4
GET /palladio.jpg HTTP/1.1
User-Agent: Mozilla/5.0
14 headers
finish 142740 142740
Connection : Host IP 127.0.0.1, Port 64515, socket 4
GET /favicon.ico HTTP/1.1
User-Agent: Mozilla/5.0
13 headers
Server Error : No such file ./favicon.ico!
Connection : Host IP 127.0.0.1, Port 64518, socket 4
```

위에서 14 headers라고 표시된 부분을 보면 header field의 수를 표시하고 있다.  
브라우저는 파일을 받으면 화면에 보여주게 된다.





hw2는 이제 다른 요청을 받을 준비가 된다.

요청한 파일이 없으면 아래와 같은 에러 메시지를 출력한다.

Server Error : No such file **./favicon.ico!**

Server Error : No such file **./aaa.html!**

그리고, 아래와 같은 HTTP Response 메시지를 보낸다.

**HTTP/1.0 404 NOT FOUND**

Connection: close

Content-Length: 0

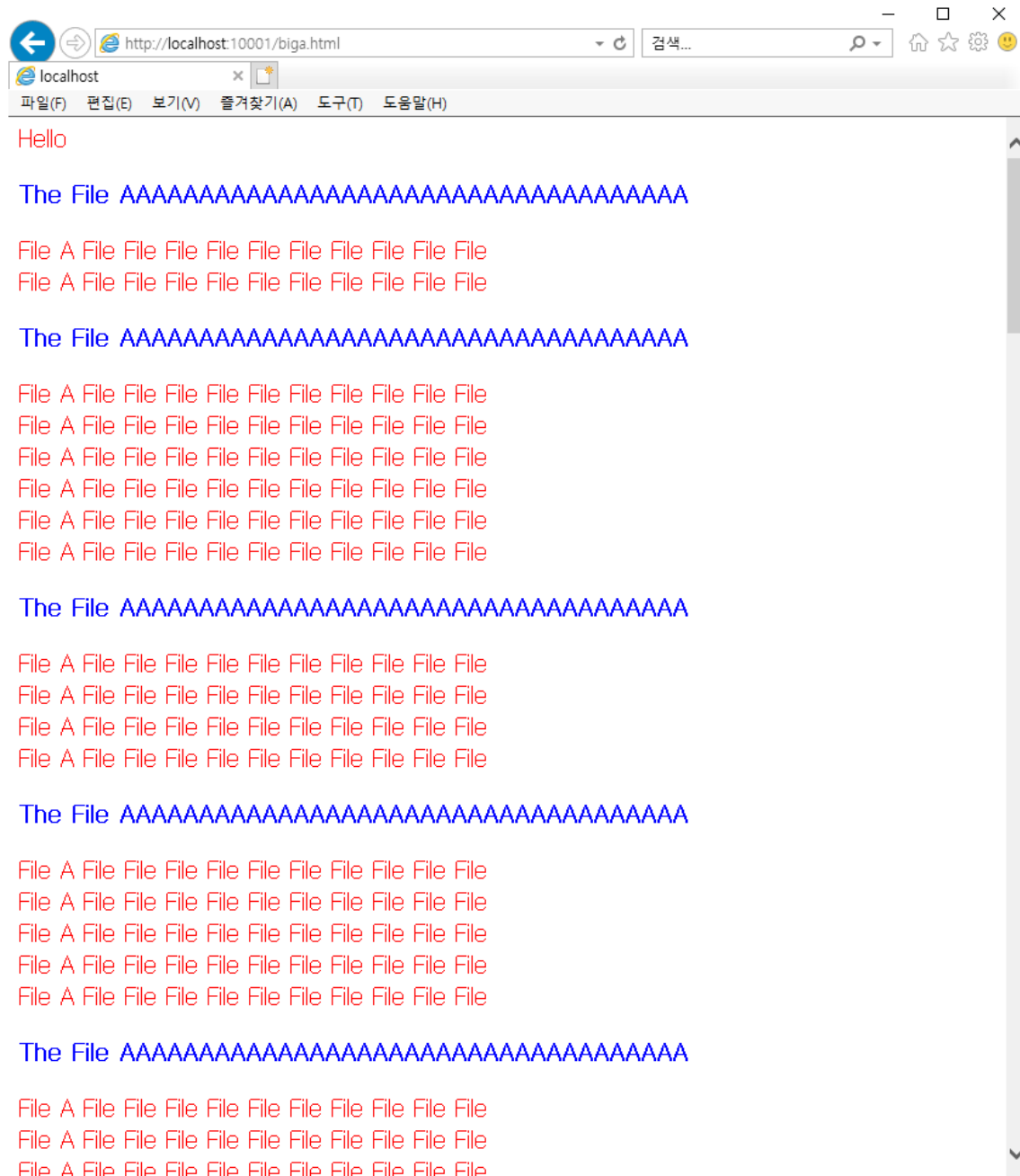
Content-Type: text/html

```
sanghwan@PC: ~/dbox/classes241/network/homework/hw242_webserver
Connection : Host IP 127.0.0.1, Port 64515, socket 4
GET /favicon.ico HTTP/1.1
User-Agent: Mozilla/5.0
13 headers
Server Error : No such file ./favicon.ico!
Connection : Host IP 127.0.0.1, Port 64518, socket 4
close error: Bad file descriptor
No Data: close connectionConnection : Host IP 127.0.0.1, Port 64547, socket 4
GET /biga.html HTTP/1.1
User-Agent: Mozilla/5.0
14 headers
finish 10818 10818
Connection : Host IP 127.0.0.1, Port 64548, socket 4
GET /aaa.html HTTP/1.1
User-Agent: Mozilla/5.0
14 headers
Server Error : No such file ./aaa.html!
Connection : Host IP 127.0.0.1, Port 64549, socket 4
```

HTTP Response 메시지가 브라우저로 잘 도착하는지를 확인하고 싶을 때는, 브라우저 대신에 nc를 사용해서 hw2와 연결하면 정상적인 response 메시지를 볼 수 있다.

```
sanghwan@PC: ~  
sanghwan@PC-:~$ nc localhost 10000  
GET /a.html HTTP/1.0  
Host: netapp.cs.kookmin.ac.kr  
  
HTTP/1.0 200 OK  
Connection: close  
Content-Length: 44  
Content-Type: text/html  
  
<HTML>  
<BODY>  
Hello  
File A  
</BODY>  
</HTML>
```

다른 브라우저도 가능하고, hw2를 다른 포트번호를 이용해서 실행해도 서버의 역할을 수행할 수 있다.



#### 세부 구현 설명

- 프로그램이 실행되면, 학번과 이름을 위 그림과 같이 출력한다. 영어로 출력한다.
- 이 과제를 위해서는 **server.c** 적절하게 수정하면 쉽게 구현할 수 있다.

## 참고.

## 테스트 방법

첨부한 labtest.sh, .html, .jpg 파일을 소스코드가 있는 곳에 복사한 후 10000 포트를 사용하여 실행한다.

이후 브라우저에서 다음 URL들이 정상적으로 동작하는지 확인한다.

또한 User-Agent가 잘 출력되는지, 헤더의 개수가 잘 출력되는지 확인한다.

`http://localhost:10000/biga.html`

`http://localhost:10000/a.html`

`http://localhost:10000/b.html`

`http://localhost:10000/palladio.jpg`

## 자동화테스트

위 매뉴얼 테스트가 잘되면 아래와 같이 자동화 테스트를 진행한다. wget 설치 필요.

첨부한 labtest.sh, .html, .jpg 파일을 소스코드가 있는 곳에 복사한다.

그리고 구현한 파일 확장자에 따라 다음 명령 중 하나를 실행. 아래 그림 참조.

```
$ bash -x labtest.sh hw2sol.c 20000
```

```
$ bash -x labtest.sh hw2sol.java 20000
```

```
$ bash -x labtest.sh hw2sol.py 20000
```

```
$ bash -x labtest.sh hw2sol.cpp 20000
```

최종 라인에 “Result: 63 6 hw2sol.c”이 나와야 한다.

참고로 이 자동화테스트는 실행파일을 강제로 죽이기 때문에 stdout으로 출력한 내용이 있으면 **flush** 를 **시켜야** 정상적으로 출력 내용이 출력됨.



```
sanghwang@PC: ~/dbox/classes241/network/homework/hw242_webserver$ bash -x labtest.sh hw2sol.c 20000
+ rm -f a.out down_temp_a down_temp_b down_temp_biga down_temp_pal output.txt
+ src=hw2sol.c
+ portnum=20000
+ echo hw2sol.c
hw2sol.c
+ echo 20000
20000
+ H1='HW: 2'
+ H2='PROF: LEE'
+ [[ hw2sol.c =~ W.cpp ]]
+ [[ hw2sol.c =~ W.c ]]
+ gcc hw2sol.c
+ sleep 1
+ ./a.out 20000
+ wget -O down_temp_biga '--header=HW: 2' '--header=PROF: LEE' -t 2 -T 10 http://localhost:20000/biga.html
--2024-04-15 13:17:58-- http://localhost:20000/biga.html
Resolving localhost (localhost)... 127.0.0.1
Connecting to localhost (localhost)|127.0.0.1|:20000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10818 (11K) [text/html]
Saving to: 'down_temp_biga'

down_temp_biga      100%[=====>] 10.56K --.-KB/s   in 0s

2024-04-15 13:17:58 (56.9 MB/s) - 'down_temp_biga' saved [10818/10818]

+ sleep 1
+ wget -O down_temp_a '--header=HW: 2' '--header=PROF: LEE' -t 2 -T 10 http://localhost:20000/a.html
--2024-04-15 13:17:59-- http://localhost:20000/a.html
Resolving localhost (localhost)... 127.0.0.1
Connecting to localhost (localhost)|127.0.0.1|:20000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 44 [text/html]
Saving to: 'down_temp_a'
```

```
+ succcnt=0
+ diff biga.html down_temp_biga
+ [[ 0 == 0 ]]
+ let succpat=succpat+1
+ let succcnt=succcnt+1
+ diff a.html down_temp_a
+ [[ 0 == 0 ]]
+ let succpat=succpat+2
+ let succcnt=succcnt+1
+ diff b.html down_temp_b
+ [[ 0 == 0 ]]
+ let succpat=succpat+4
+ let succcnt=succcnt+1
+ cmp palladio.jpg down_temp_pal
+ [[ 0 == 0 ]]
+ let succpat=succpat+8
+ let succcnt=succcnt+1
+ grep 'User-Agent: Wget' output.txt
User-Agent: Wget/1.19.4
User-Agent: Wget/1.19.4
User-Agent: Wget/1.19.4
User-Agent: Wget/1.19.4
+ [[ 0 == 0 ]]
+ let succpat=succpat+16
+ let succcnt=succcnt+1
+ grep '7 headers' output.txt
7 headers
7 headers
7 headers
7 headers
+ [[ 0 == 0 ]]
+ let succpat=succpat+32
+ let succcnt=succcnt+1
+ echo 'Result: ' 63 6 hw2sol.c
Result: 63 6 hw2sol.c
sanghwang@PC: ~/dbox/classes241/network/homework/hw242_webserver$
```