Network project 1 report

2018046408 응용물리학과 김도하

1. Description of my server’s design

로컬 호스트를 통해 서버를 열었다. 서버가 열리면, 클라이언트는 웹 브라우저(크롬, 파이어폭스 등)를 통해 서버에 접속할 수 있고, 서버의 역할은 클라이언트가 보낸 HTTP request message를 읽고 적절한 응답을 보내는 것이다. 따라서 서버는 다음과 같은 역할을 수행할 수 있도록 디자인되어야 한다. (실제로도 이렇게 디자인되었다.)

* 요청 Path에서 “/”을 읽어 적절한 리소스를 반환해야 한다.
* 응답에 적절한 상태 코드(200, OK)와 헤더를 적어줘야 한다. 필수적으로 리소스 파일의 크기를 나타내는 Content-Length, 파일 확장자에 따른 MIME 값을 나타내는 Content-Type을 반드시 헤더에 적어줘야 한다.
* Content-Length, Content-Type의 값을 틀리게 보내면 안 된다. Content-Length는 브라우저가 헤더 다음 몇 바이트를 읽어야 하는지를 알려준다. Content-Length를 실제 데이터 값보다 더 적은 값으로 보냈다면 데이터가 중간에 잘려 데이터 손실이 발생할 수 있다. Content-Type을 엉뚱하게 보내면 파일 데이터를 읽을 수 없거나, 의도하지 않은 방향으로 보여줄 수 있다. 예를 들어, HTML 파일을 보내주는데 Content-Type을 text/plain으로 알려주면, 브라우저는 HTML 코드 그대로를 보여줄 것이다.

서버의 작동 방식은 다음과 같다. 먼저 클라이언트와 통신할 수 있는 소켓을 생성한다. Bind()함수를 통해 소켓에 주소를 할당하고, listen()함수를 통해 연결 요청을 대기한다. Accept()함수로 클라이언트의 연결 요청을 수락한다. 클라이언트의 요청이 있기 전까지 서버는 accept()에서 대기한다. 클라이언트가 http request를 보내면, 서버는 이 소켓이 write()한 값을 read()하여 그 내용을 분석한 후(이 때 http request message를 서버의 콘솔에 출력되도록 했다), 적절하게 처리한 뒤 소켓을 닫음으로써 클라이언트와의 접속을 해제한다. 이 과정에서 멀티 스레드 방식을 채택하지 않고 멀티 프로세스 방식을 채택하였다.

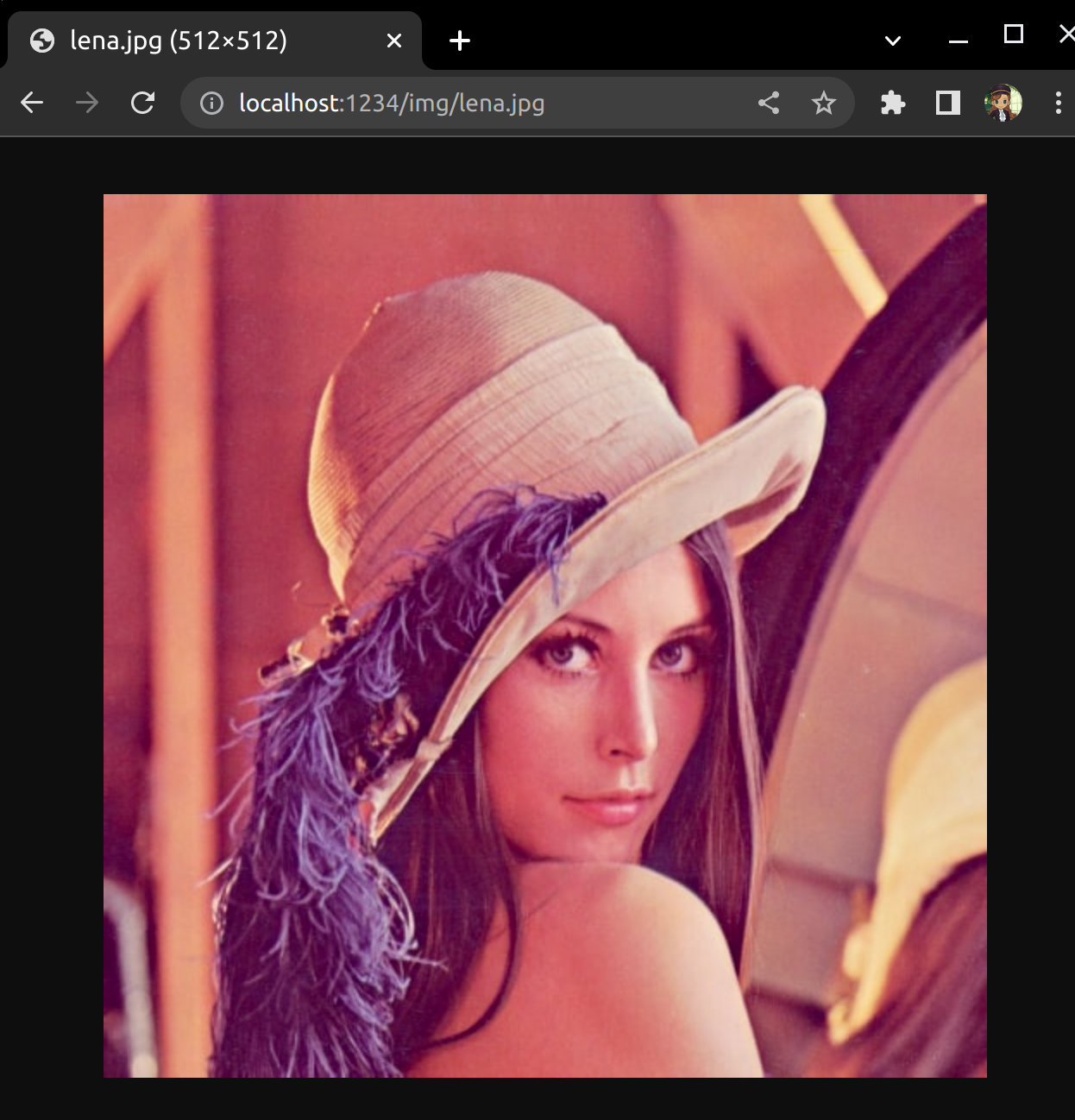
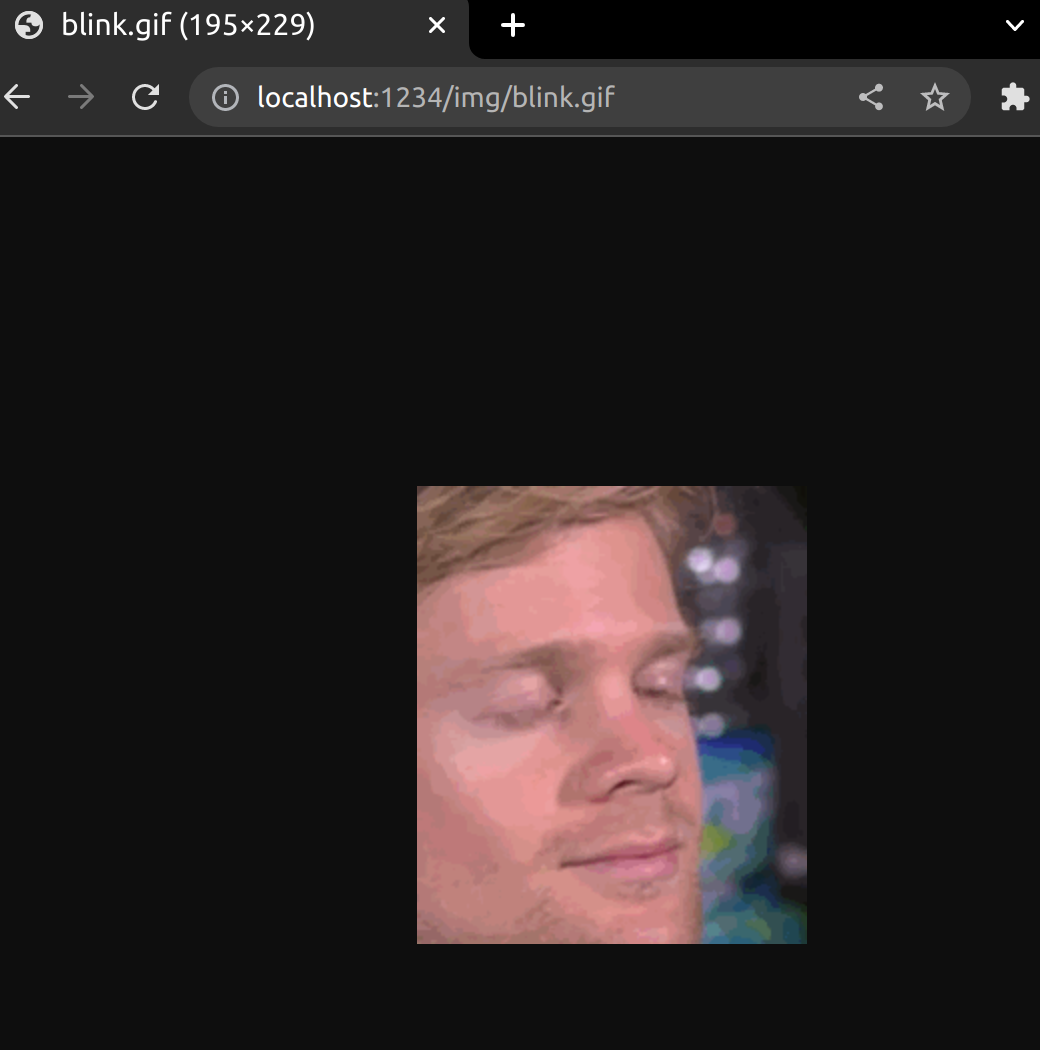
디자인한 서버의 단점을 몇 가지 생각해 볼 수 있다.

1. 접근 가능한 디렉토리를 한정시키거나, 요청할 수 있는 확장자를 제한해야 한다. 디렉토리를 요청한 경우도 고려해야 한다. 이 서버에서는 이를 제한하지 않았기 때문에 보안상 취약점이 있을 수 있다.
2. Method를 고려하지 않고 있다. Method가 GET인 경우만 다루고 있을 뿐, 이외의 경우를 처리하지 않는다(대표적으로 POST).
3. 데이터의 형식에 따라 HTML에서 어떻게 보여줘야 할 지 생각해야 한다. 지금의 경우 모든 파일 형식에 대해 헤더의 Content-Disposition: inline으로 설정함으로써 데이터를 그대로 웹 페이지에 디스플레이 하도록 했지만, 경우에 따라 다운로드가 필요한 경우도 있을 것이고, pdf 확장자의 경우 크롬의 경우 pdf 뷰어를 지원하지만 이외의 경우 클라이언트의 컴퓨터에 pdf 뷰어를 따로 설치한 경우에만 볼 수 있다. 이 문제는 C에서는 다루기 어려운 부분이다(Python, Ruby 계열의 서버 프레임워크에서는 이 기능을 쉽게 제공하고 있다.)
4. Difficulties and Solutions

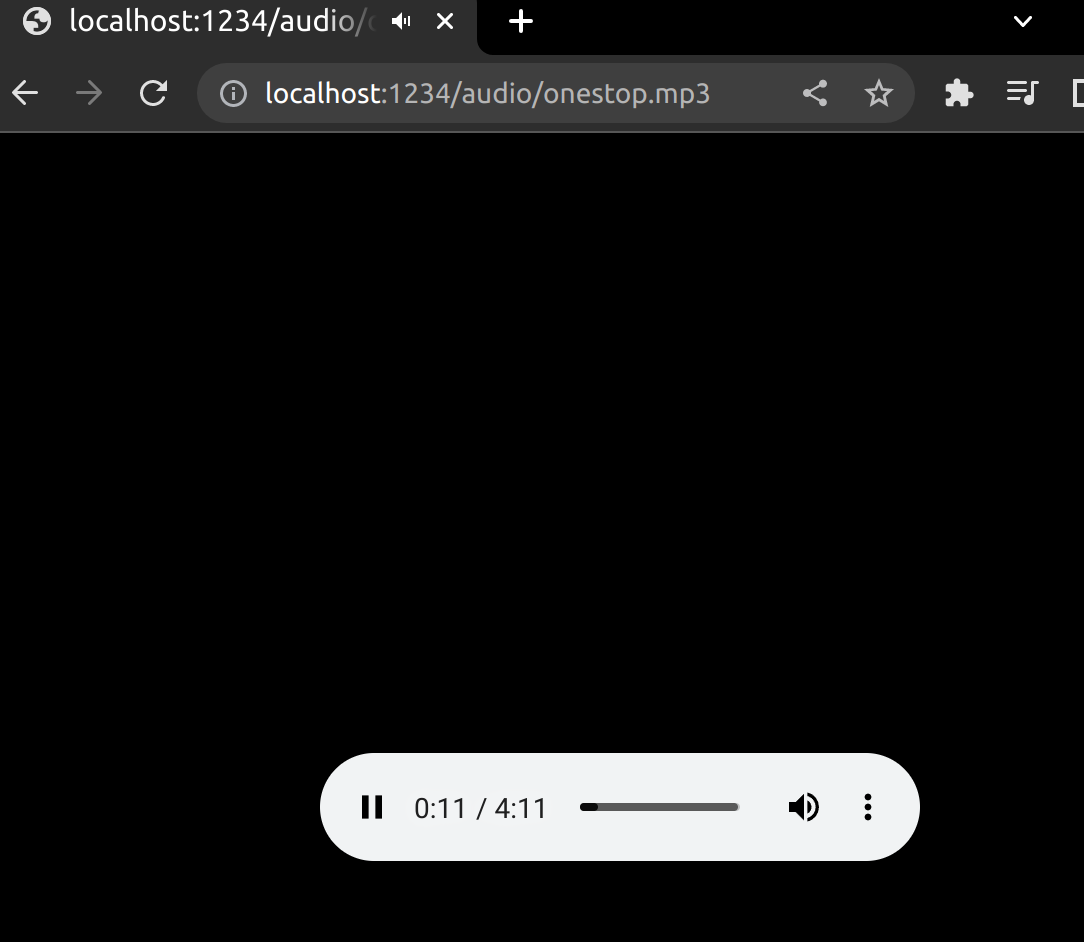
소켓 프로그래밍 방식은 수업을 통해 들었기 때문에 어느정도 이해가 있었지만, HTTP request와 respond의 경우 각각의 프로토콜과 헤더가 어떤 식으로 이루어져야 하는지, 무슨 내용을 작성해야 하는지 등을 알아내야 하는 점이 어려웠다. HTTP 관련 문서들을 찾아보고 헤더의 의미를 하나하나 검색함으로써 문제를 해결했다. 또한 소켓 프로그래밍 자체를 c언어에서 해본 적이 없기 때문에 관련 라이브러리와 코드를 짜는 법이 전부 난감했지만, 인터넷에서 자료를 검색하여 보는 것으로 해결했다. 또한 멀티 스레드나 멀티 프로세스 방식을 사용해 본적이 없어 이 부분을 이해하기도 난감했다. 현재 운영체제 과목을 듣는 친구에게 물어보며 어렵게 해결했다.

1. Explanation of sample outputs

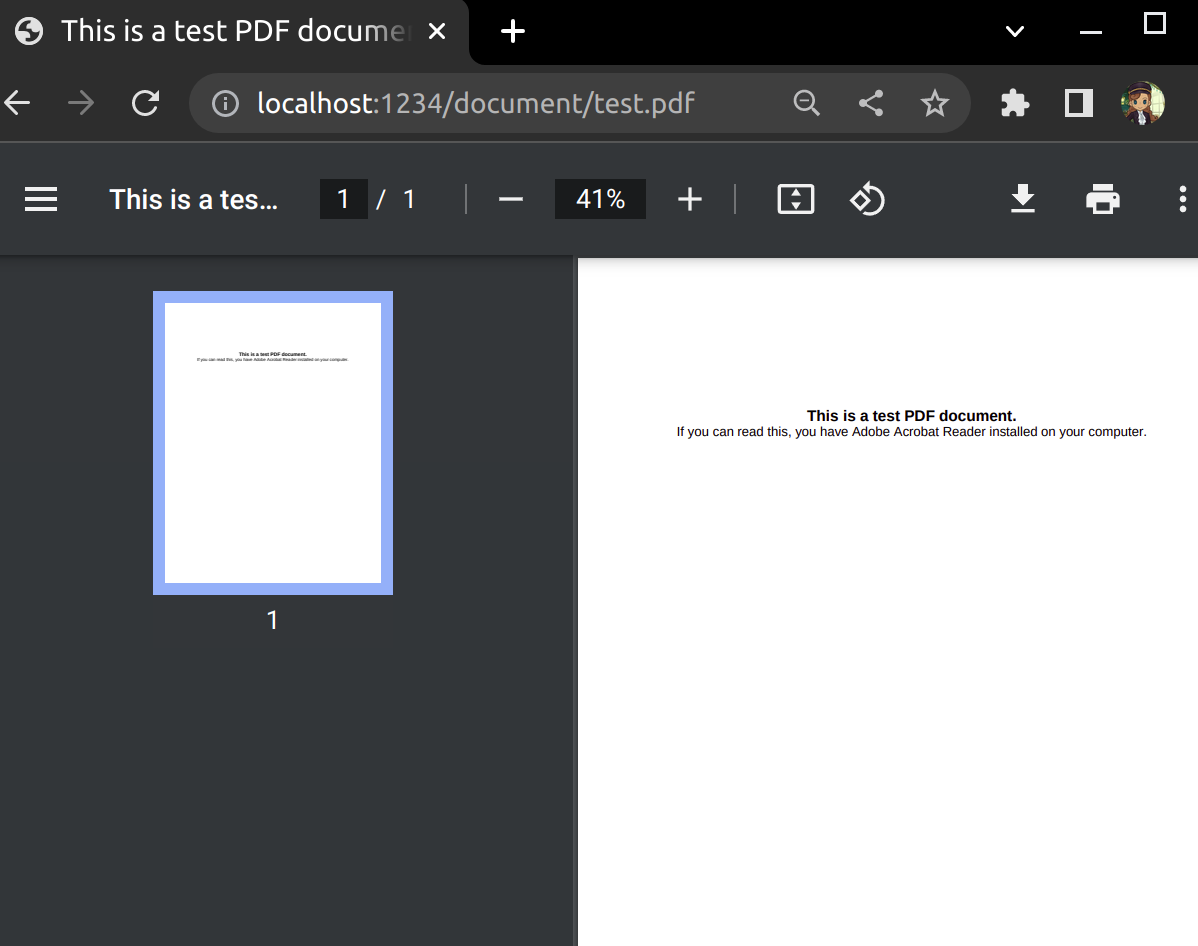
readme.txt에 테스트 할 수 있는 출력 값들에 대해 적어놓았으나, 이 레포트에도 설명한다. 기본적으로 서버의 터미널에서는 클라이언트의 http request message를 볼 수 있다.

a) 사진들: localhost:{port\_num}/img/{file\_name}.{file\_format}으로 액세스 할수 있다.

b)음악: localhost:{port\_num}/audio/{file\_name}.mp3로 액세스 할 수 있다.



c)PDF file: localhost:{port\_num}/document/{file\_name}.pdf로 액세스 할 수 있다. 뷰어 필요.



d)html file: localhost:{port\_num}/{filename}.html로 액세스할 수 있다.

