

Introduction to Computer Networks

Machine Problem 1: HTTP client

Announced: April 1 (Thu), Due: April 19 (Mon)

1. 과제의 목표

Web browser와 같은 역할을 하는 HTTP client 프로그램을 작성한다.

2. 작성해야 할 프로그램

http_client: 웹 서버에 접속하여 HTTP request를 보내고 response를 받아오는 프로그램.

3. 상세설명 (주의 깊게 읽고 지시사항을 따라 구현할 것)

(1) http_client 프로그램은 다음과 같은 형태로 실행한다.

```
./http_client http://hostname[:port][/path/to/file]
```

프로그램 실행 예:

```
./http_client http://127.0.0.1/index.html
./http_client http://www.sogang.ac.kr
./http_client http://12.34.56.78:8888/somfile.txt
./http_client http://localhost:5678/somedir/anotherfile.html
```

- 만약 [:port] 부분이 없으면 default 포트번호는 80번으로 한다.
- 만약 [/path/to/file] 부분이 없으면 default 경로는 / 로 하면 된다.
- 맨 앞에는 반드시 http:// 를 붙여야 한다. 그렇지 않으면 잘못된 포맷이다.

(2) 위에서 보드시피, 프로그램에 들어가는 인자는 한 개이다. 만약에 인자가 없거나 인자가 두개 이상이면, 다음과 같은 에러메시지를 출력하고 프로그램을 종료한다.

```
usage: http_client http://hostname[:port][/path/to/file]
```

(3) 인자 부분에서 http:// 는 반드시 붙여야 한다. 만약 이 부분이 없거나 잘못되었으면 위와 같은 에러메시지를 출력하고 프로그램을 종료한다.

(4) 인자에 대한 parsing이 끝나면, 인자로 주어진 hostname과 port로 TCP 연결요청을 한다. 이 때 hostname 부분을 domain name으로 썼을 경우, DNS에서 해당 domain name에 대한 IP address를 찾지 못하면, 다음과 같은 에러메시지를 출력하고 프로그램을 종료한다.

getaddrinfo: Name or service not known

- 위와 같은 에러메시지 출력은 다음과 같은 코드를 사용하면 된다. 여기 hostname 부분에 domain name을 쓰는 경우, DNS에 IP address를 얻지 못하면 위와 같은 메시지가 출력된다.

```
if ((rv = getaddrinfo(hostname, port, &hints, &servinfo)) != 0) {
    fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(rv));
    return 1;
}
```

(5) IP address는 얻었는데 서버와의 TCP 연결에 실패하는 경우에는 아래와 같은 에러메시지를 출력하고 프로그램을 종료한다.

connect: Connection refused

- 위와 같은 에러메시지 출력은 다음과 같은 코드를 사용하면 된다. 여기 connect 함수의 리턴값이 -1이면 위와 같은 에러메시지가 출력된다.

```
if (connect(sockfd, servinfo->ai_addr, servinfo->ai_addrlen) == -1) {
    close(sockfd);
    perror("connect");
    exit(1);
}
```

(6) 서버와 TCP 연결을 성공적으로 맺고 나면, 그 다음에는 HTTP request message를 보내야 한다. 웹 브라우저에서 주소창에 어떤 URL을 넣으면 아래와 같은 메시지가 전송된다.

```
GET / HTTP/1.1
Host: 163.239.23.130:8080
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/86.0.4240.198 Whale/2.9.115.16 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,
image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
```

- 우리가 만드는 http_client에서는 맨 앞의 두 줄만 서버에 보내는 것으로 한다.

- 첫번째 줄에서는 GET을 쓰고 그 다음에 오브젝트에 대한 경로를 쓰고 그 다음에 HTTP/1.1과 같이 쓰면 된다. 예를 들어 프로그램을 다음과 같이 실행시켰다면,

```
./http_client http://localhost:5678/somedir/anotherfile.html
```

- HTTP request 메시지의 첫번째 줄은 다음과 같이 된다.

GET /somedir/anotherfile.html HTTP/1.1

- 여기서 주의할 것은, HTTP 헤더에서는 줄바꿈이 CRLF이라는 것이다. 그래서 한 줄의 맨 끝에는 `\r\n`을 붙여주어야 한다.

- HTTP request의 두번째 줄은 `Host:` 라고 쓰고 `hostname[:port]` 부분을 써주면 된다.

Host: localhost:5678

- 두번째 줄도 맨 끝에는 `\r\n`으로 끝낸다.
- 세번째 줄은 아무것도 없이 `\r\n`만 있도록 한다. 서버는 `\r\n\r\n` 패턴을 보고 HTTP request가 끝났다는 것을 안다.

(7) HTTP request를 보내고 난 다음에는, 서버로부터 HTTP response를 받아야 한다.

- <http://info.cern.ch> 에 접속하면 다음과 같은 HTTP response 메시지를 받을 수 있다.

HTTP/1.1 200 OK

Date: Wed, 31 Mar 2021 05:59:19 GMT

Server: Apache

Last-Modified: Wed, 05 Feb 2014 16:00:31 GMT

ETag: "286-4f1aadb3105c0"

Accept-Ranges: bytes

Content-Length: 646

Connection: close

Content-Type: text/html

```
<html><head></head><body><header>
<title>http://info.cern.ch</title>
</header>
```

```
<h1>http://info.cern.ch - home of the first website</h1>
<p>From here you can:</p>
<ul>
<li><a href="http://info.cern.ch/hypertext/WWW/TheProject.html">Browse the
first website</a></li>
<li><a href="http://line-
mode.cern.ch/www/hypertext/WWW/TheProject.html">Browse the first website
using the line-mode browser simulator</a></li>
<li><a href="http://home.web.cern.ch/topics/birth-web">Learn about the
birth of the web</a></li>
<li><a href="http://home.web.cern.ch/about">Learn about CERN, the physics
laboratory where the web was born</a></li>
```

```
</ul>
</body></html>
```

- 이 HTTP response 메시지는 헤더 부분과 데이터 부분으로 나뉘는데, 헤더 부분을 아래와 같다. 헤더의 각 줄은 `\r\n`으로 끝나고, 아무것도 없이 빈 줄에 `\r\n`만 있으면 거기가 헤더의 끝이다.

```
HTTP/1.1 200 OK
Date: Wed, 31 Mar 2021 05:59:19 GMT
Server: Apache
Last-Modified: Wed, 05 Feb 2014 16:00:31 GMT
ETag: "286-4f1aadb3105c0"
Accept-Ranges: bytes
Content-Length: 646
Connection: close
Content-Type: text/html
```

- 일단 첫번째 줄은 status code이다. 이 부분은 화면에 출력해준다.
 - 다른 내용은 무시해도 좋지만, `Content-Length`의 값은 기억해야 한다. 위의 예에서 646이라고 되어있는데, 이는 데이터 부분의 크기가 646바이트임을 의미한다. 이것을 알아야 데이터를 제대로 받을 수가 있다. `Content-Length`는 `content-length`인 경우도 있고 `Content-length`인 경우도 있으니 알파벳 대소문자와 상관없이 필드의 이름을 읽어야 한다.
 - 어떤 웹사이트들은 `Content-Length` 필드를 헤더에 제공하지 않는 경우가 있다. 이런 경우에는 `Content-Length not specified.` 라고 화면에 출력하고 종료하면 된다.
- (8) 데이터 부분은 화면에 출력하지 않고 `[학번].out`이라는 파일에 쓰도록 한다.
- 예를 들어, 아래와 같이 프로그램을 실행시켰다고 하자.

```
./http_client http://info.cern.ch/hypertext/WWW/TheProject.html
```

그러면 HTTP response의 data 부분은 TheProject.html의 내용이 될 것이다. 따라서 프로그램이 출력하는 `[학번].out`은 TheProject.html과 완전히 일치해야 한다. (일부가 잘리거나 문자가 추가되어서는 안된다.)

4. 프로그램 실행 예

- (1) 인자의 수가 맞지 않는 경우

```
[실행]
./http_client
```

[결과]

```
usage: http_client http://hostname[:port][/path/to/file]
```

(2) 인자의 수는 맞으나 인자의 포맷이 틀린 경우

[실행]

```
./http_client www.sogang.ac.kr
```

[결과]

```
usage: http_client http://hostname[:port][/path/to/file]
```

(3) 호스트에 잘못된 도메인 이름을 적은 경우

[실행]

```
./http_client http://www.sogang
```

[결과]

```
getaddrinfo: Name or service not known
```

(4) 서버에 연결할 수 없는 경우

[실행]

```
./http_client http://163.239.23.135
```

[결과]

```
connect: Connection refused
```

(5) 서버에 연결하여 파일을 정상적으로 받은 경우

[실행]

```
./http_client http://info.cern.ch/hypertext/WWW/TheProject.html
```

[결과]

```
HTTP/1.1 200 OK
```

```
2217 bytes written to 20190000.out
```

[실행]

```
./http_client http://www.columbia.edu/~fdc/sample.html
```

[결과]

HTTP/1.1 200 OK
30644 bytes written to 20190000.out

(6) 서버에는 연결하였는데 파일이 서버에 없는 경우

[실행]

./http_client http://info.cern.ch/hypertext/WWW/TheProjec.html

[결과]

HTTP/1.1 404 Not Found
226 bytes written to 20190000.out

(7) 서버에는 연결하였는데 파일이 옮겨진 경우

[실행]

./http_client http://google.com

[결과]

HTTP/1.1 301 Moved Permanently
229 bytes written to 20190000.out

(8) 서버에서 HTTP response를 받았는데 헤더에 Content-Length 필드가 없는 경우

[실행]

./http_client http://www.google.com

[결과]

HTTP/1.1 200 OK
Content-Length not specified.

5. 주의 사항

(1) 파일의 크기가 큰 경우 recv 함수 호출 한 번으로 데이터 전체를 받을 수 없는 경우가 있다. 이런 경우에는 여러 번 recv 함수를 호출하여 데이터를 모두 받을 수 있도록 구현해야 한다. 아래 <http://www.columbia.edu/~fdc/sample.html>의 경우 한번의 recv 호출로는 모든 데이터를 받지 못

할 것이다.

[실행]

```
./http_client http://www.columbia.edu/~fdc/sample.html
```

[결과]

```
HTTP/1.1 200 OK
```

```
31373 bytes written to 20190000.out
```

6. 과제 제출

(1) 제출해야 하는 파일은 한 개이며, 그 이름은 아래와 같다. 빨간 색 부분은 자기 학번으로 변경한다.

```
http_client_20190000.cc
```

(2) 이 파일은 cspro에서 g++ 컴파일러로 다음과 같이 컴파일 될 예정이다.

```
g++ -o http_client http_client_20190000.cc
```

- 제출하기 전에 컴파일과 실행이 잘 되는지 테스트해보고 제출한다.

(3) 아이디어는 상의할 수 있으나, 코드를 표절하지 말 것. 수강생들 사이 뿐만 아니라 인터넷에 있는 코드도 동일하게 사용하지 말아야 한다. 표절인 경우에는 0점 처리된다.

(4) 지각 제출은 마감일 후 3일까지 허용하며, 하루에 10%씩 감점한다.