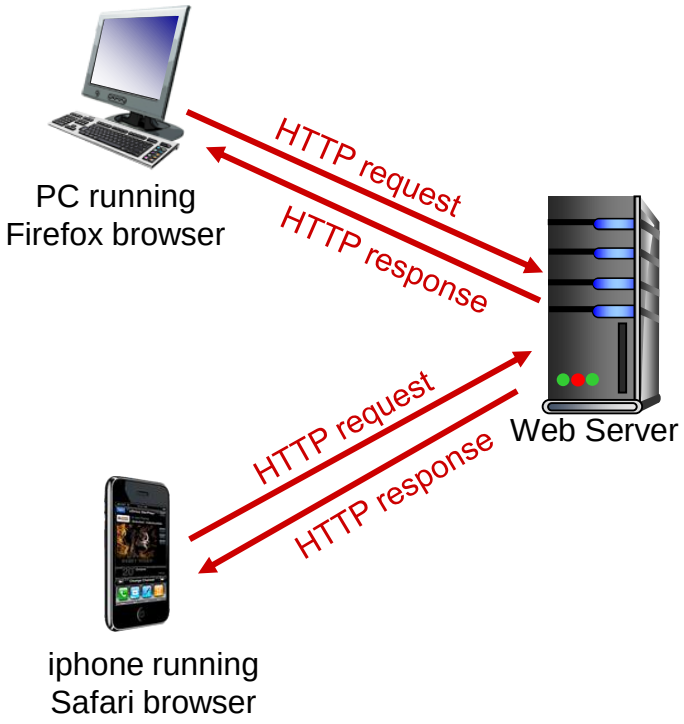
과제 2: 간단한 웹 서버 제작

- **Web Server**
  - WWW 서버
  - HTTP protocol
  - HTML 파일을 포함한 각종 파일 전송



`www.someschool.edu/someDept/pic.gif`

host name　　path name

Non-persistent HTTP

suppose user enters URL:
`www.someSchool.edu/someDepartment/home.index`

(contains text, references to 10 jpeg images)

1a. HTTP client initiates TCP connection to HTTP server (process) at www.someSchool.edu on port 80

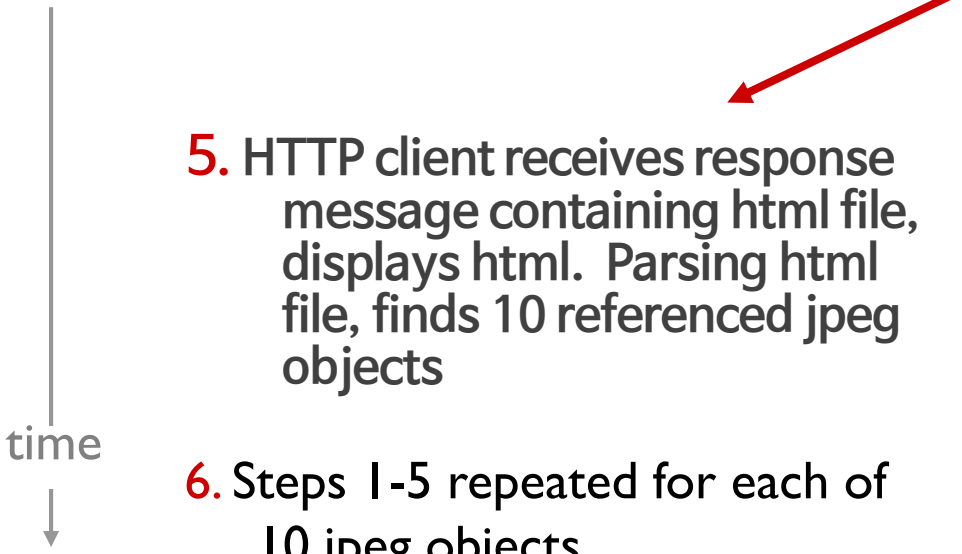
1b. HTTP server at host www.someSchool.edu waiting for TCP connection at port 80. "accepts" connection, notifying client

2. HTTP client sends HTTP *request message* (containing URL) into TCP connection socket. Message indicates that client wants object someDepartment/home.index

3. HTTP server receives request message, forms *response message* containing requested object, and sends message into its socket

time

Non-persistent HTTP (cont.)



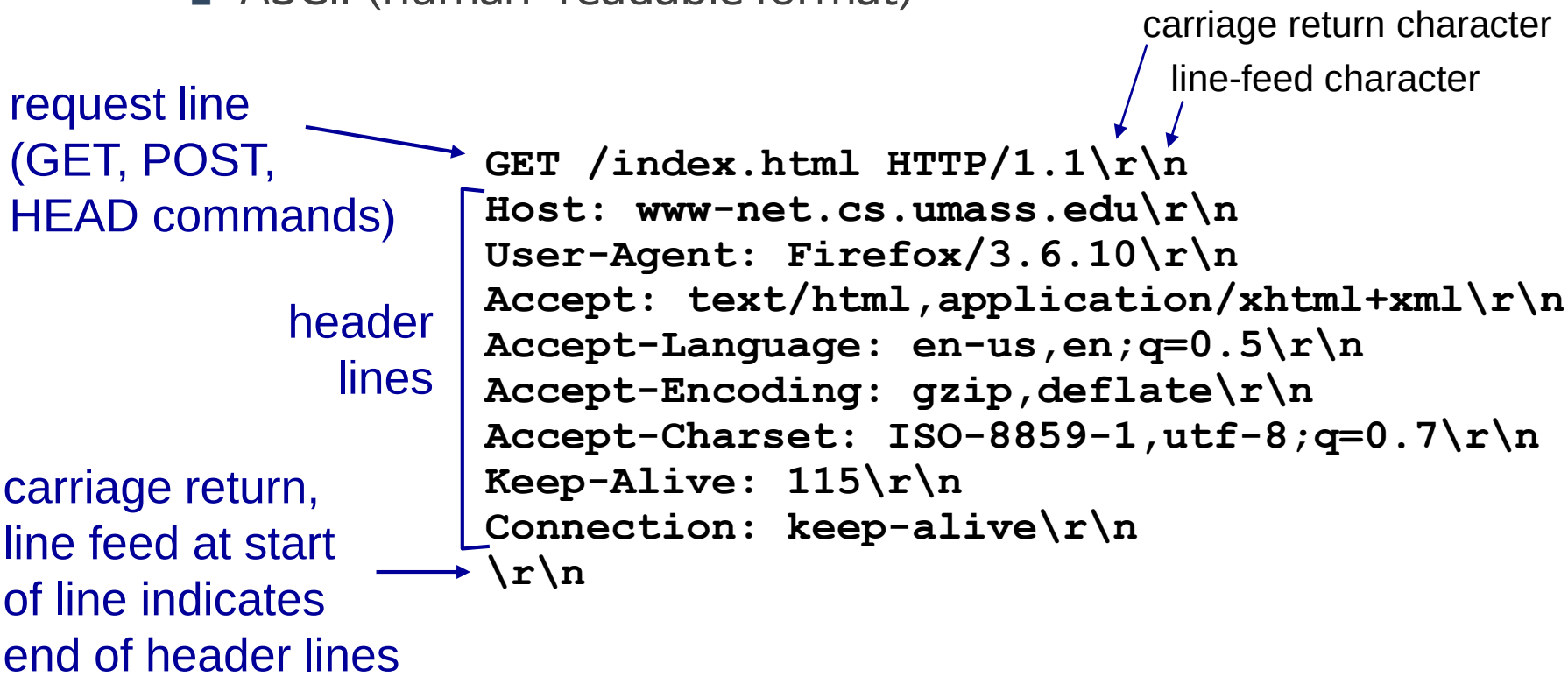
4. HTTP server closes TCP connection.

5. HTTP client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

6. Steps 1-5 repeated for each of 10 jpeg objects

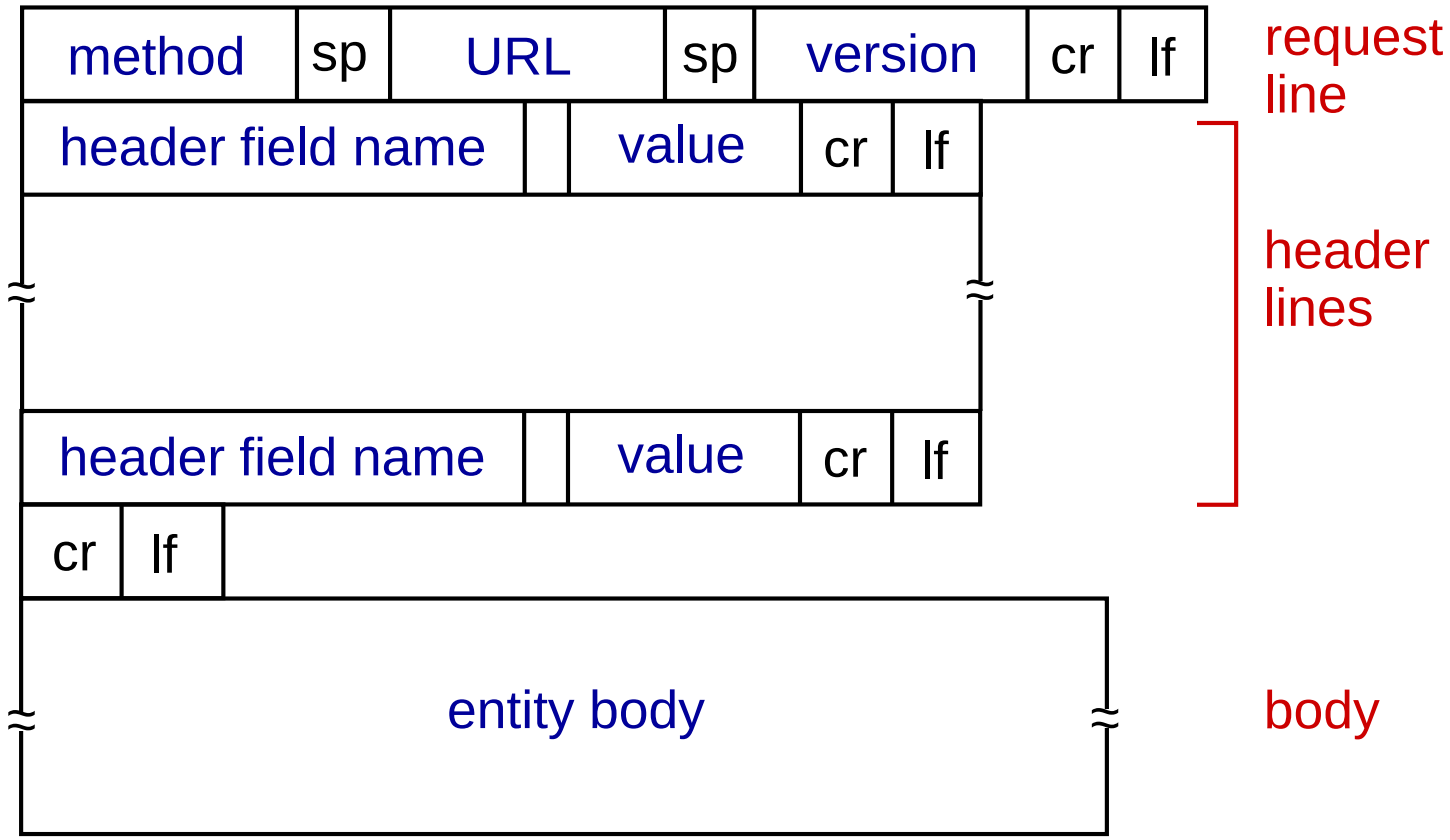
HTTP request message

- HTTP message type: request, response
- HTTP request message:
  - ASCII (human-readable format)



HTTP request message: general format

- HTTP request message format



HTTP response message

- HTTP response message:
  - ASCII (human-readable format)

status line (protocol status code status phrase)

header lines

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02 GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-1\r\n
\r\n
data data data data data ...
```

data, e.g., requested HTML file

과제 2: 간단한 웹 서버 제작

Web Server의 제작

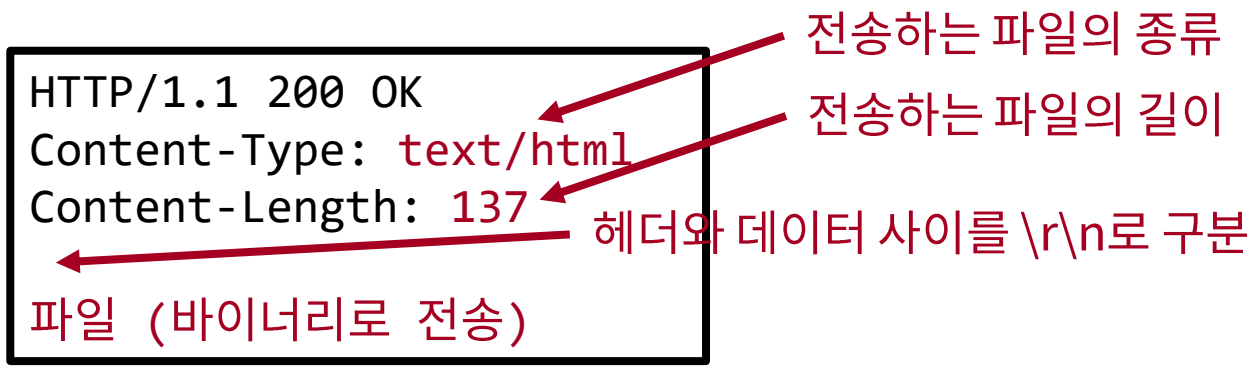
- TCP접속 후 클라이언트로부터 HTTP Request Message 받음

```
GET /test.html HTTP/1.1
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/66.0.3359.139 Safari/537.36
Accept: Accept-Encoding: gzip, deflate
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
```

- GET 다음에 요청 파일 경로 추출
  - / 일 경우, /index.html 와 같이 취급

과제 2: 간단한 웹 서버 제작

- 해당 파일을 HTTP response message 로 보내줌



- 해당 파일이 없을 경우

```
HTTP/1.1 404 Not Found
Content-Type: text/html
Content-Length: 137

에러 파일 (바이너리로 전송)
```

과제 2: 간단한 웹 서버 제작

- 파일의 확장자에 따라 Content-Type을 바꿔야 함
  - .html : text/html
  - .jpg : image/jpg
  - .png : image/png
- 파일의 크기에 따라 Content-Length 설정
- 파일은 binary 형태로 전송
- 전송 후, persistent HTML이기 때문에 다음 요청 기다림

과제 2: 간단한 웹 서버 제작

구현해야 하는 것들

- 다중 접속 허용 (fork 를 이용한 multiprocess 구조)
- html, jpg, png 등의 파일 처리
- 해당 파일이 없을 경우, 404 에러 발생
- 포트 번호 변경 가능

실행

- 포트 번호가 80일 경우

```
# ./web_server 80
```

구동 확인

- 실제 웹브라우저로 접속 (크롬, 익스플로러, 엣지 중 선택)
- 요청된 페이지를 제대로 보여줘야 함
- 없는 파일을 요청할 경우 에러 페이지 보여줌

과제 2: 간단한 웹 서버 제작

제출 방법

- 소스 코드 (.h, .c 파일)
  - 파일을 여러 개로 나눠 제출해도 됨
  - 여러 개로 나눌 경우 Makefile 함께 제출

- 테스트용 웹페이지 (.html, .jpg, .png)

- 구현 내용 작성 파일 (.pdf .doc, .hwp)
  - 접속이나 데이터 전송 등을 어떻게 구현했는지 작성
  - 어떠한 파일들을 처리할 수 있는지 작성
  - 간단히 작성

- 위의 파일들을 스마트 캠퍼스의 과제 항목에 업로드

제출 기한

- 11월 9일 23:59 까지