COMP 4621 Lab Tutorial #4

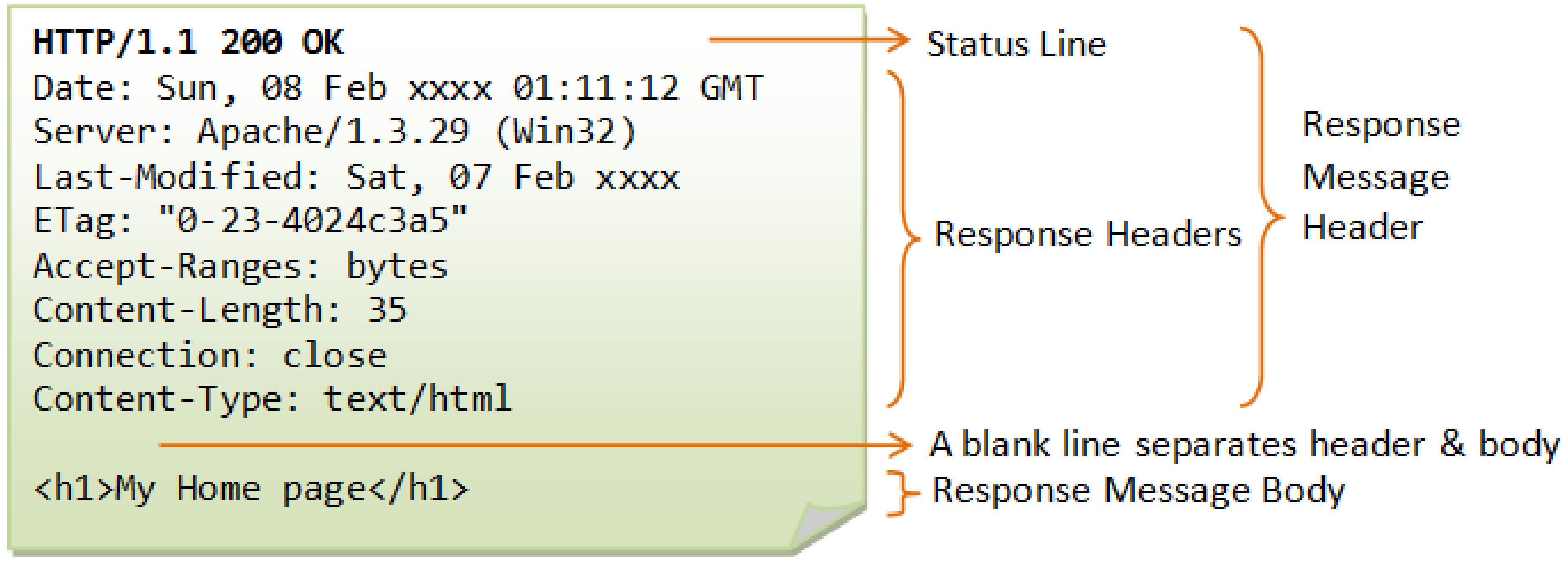
Spring 2015

Quick review

- Lab 1. Basic concepts and UDP socket
- Lab 2. TCP socket
- Lab 3. HTTP programming, multithreading
- Lab 4. HTTP Range Request, programming project

HTTP Response

- Web server attaches web content with HTTP response



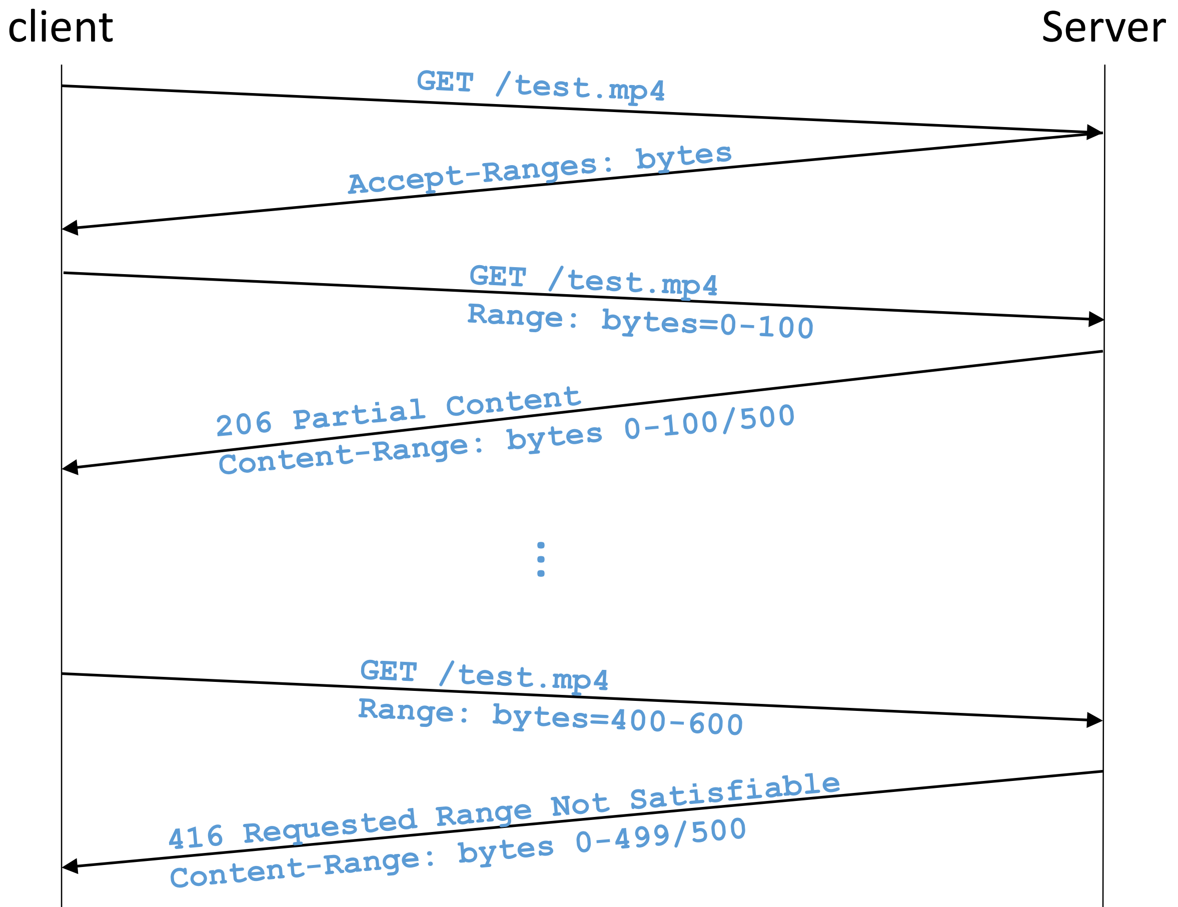
```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```

Some drawbacks…

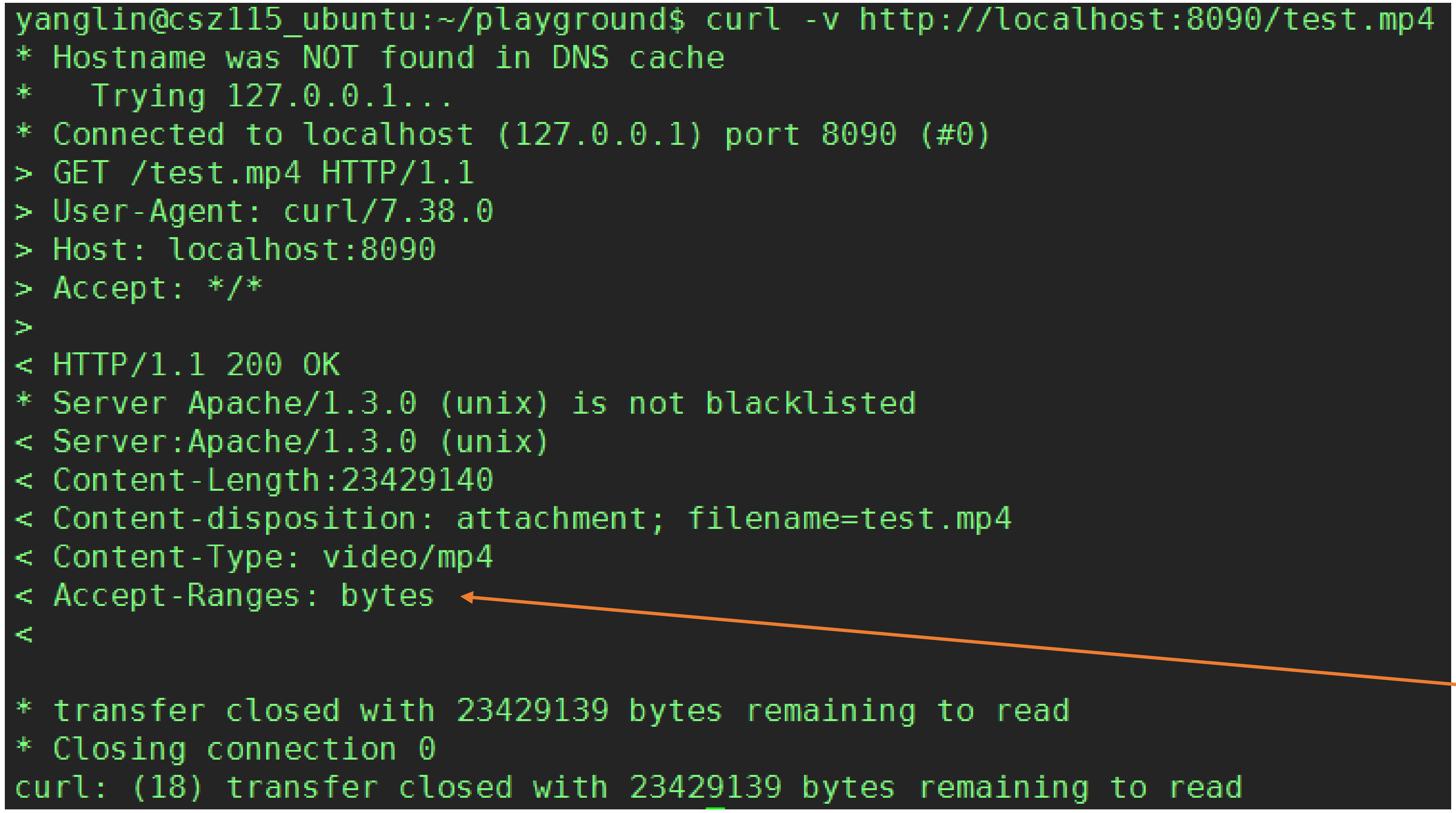
- What if the content is a very large file, *e.g.*, a video
    - Bandwidth issues
    - Bad user experience
- Solutions?
    - HTTP Range Request
    - HTTP progressive download
    - Other Streaming protocols

HTTP Range

- New in HTTP/1.1
- Only a portion of web content is responded from a server to a client.



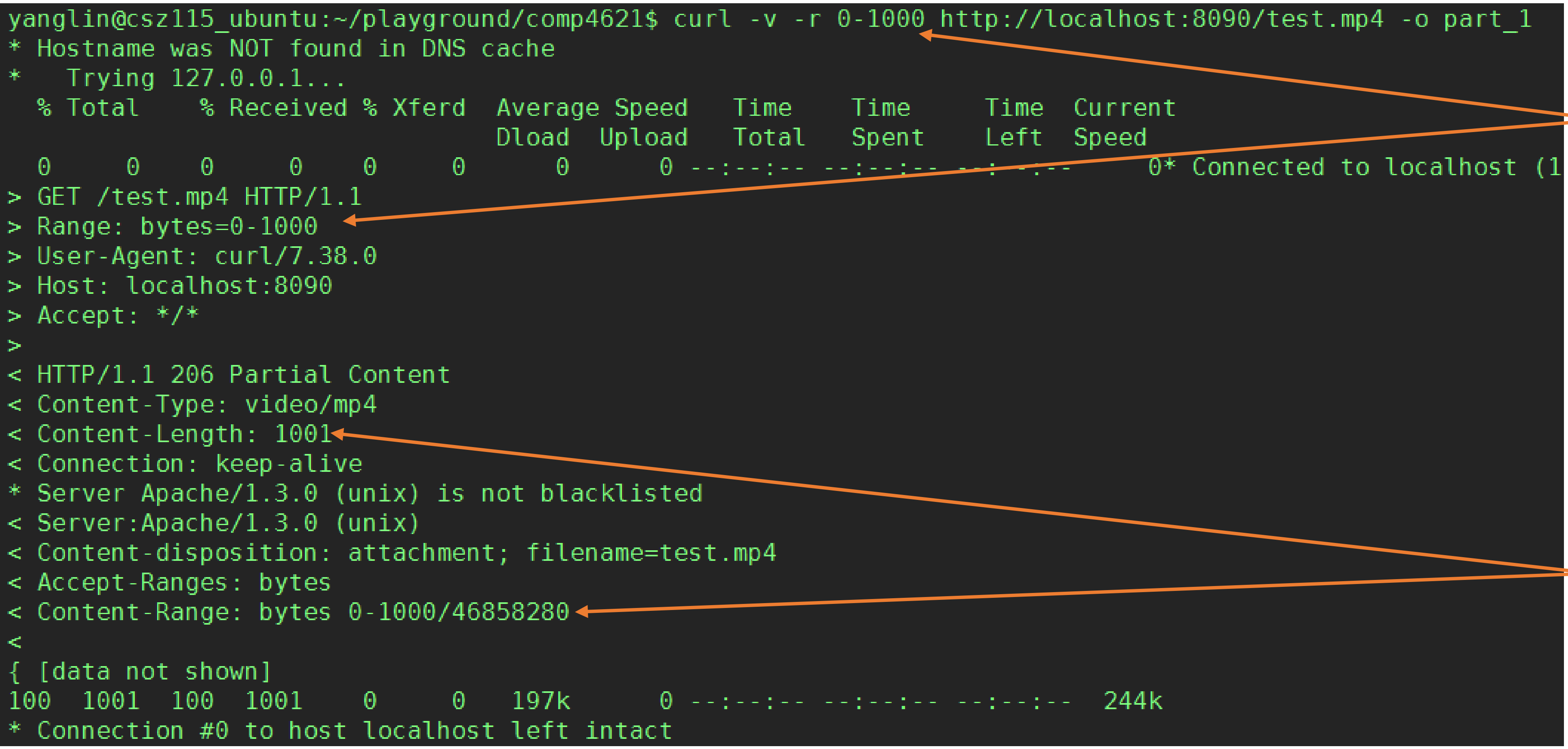
HTTP Range Interaction



```
yanglin@csz115_ubuntu:~/playground$ curl -v http://localhost:8090/test.mp4
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8090 (#0)
> GET /test.mp4 HTTP/1.1
> User-Agent: curl/7.38.0
> Host: localhost:8090
> Accept: */*
>
< HTTP/1.1 200 OK
* Server Apache/1.3.0 (unix) is not blacklisted
< Server:Apache/1.3.0 (unix)
< Content-Length:23429140
< Content-disposition: attachment; filename=test.mp4
< Content-Type: video/mp4
< Accept-Ranges: bytes
<

* transfer closed with 23429139 bytes remaining to read
* Closing connection 0
curl: (18) transfer closed with 23429139 bytes remaining to read
```

Web server support range

HTTP Range Interaction



```
yanglin@csz115_ubuntu:~/playground/comp4621$ curl -v -r 0-1000 http://localhost:8090/test.mp4 -o part_1
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connected to localhost (1
> GET /test.mp4 HTTP/1.1
> Range: bytes=0-1000
> User-Agent: curl/7.38.0
> Host: localhost:8090
> Accept: */*
>
< HTTP/1.1 206 Partial Content
< Content-Type: video/mp4
< Content-Length: 1001
< Connection: keep-alive
* Server Apache/1.3.0 (unix) is not blacklisted
< Server:Apache/1.3.0 (unix)
< Content-disposition: attachment; filename=test.mp4
< Accept-Ranges: bytes
< Content-Range: bytes 0-1000/46858280
<
{ [data not shown]
100  1001  100  1001    0     0   197k      0 --:--:-- --:--:-- --:--:--  244k
* Connection #0 to host localhost left intact
```

Request 0~1000 bytes

Reply with first 1K bytes
Available range: 23MB

A Useful tool: curl

- curl is a client to get documents/files from or send documents to a server, using any of the supported protocols (HTTP, HTTPS, FTP, GOPHER, DICT, TELNET, LDAP or FILE).

- Some examples

```
curl -v http://localhost:8090/
curl -v -r 0-1000 http://localhost:8090/test.mp4 -o part_1
```

url to get

Verbose, show all request and reply

Range, only request a range of file

- *More can be find in curl --help*

Practice

- Implement HTTP range based on the webserver in tutorial 3

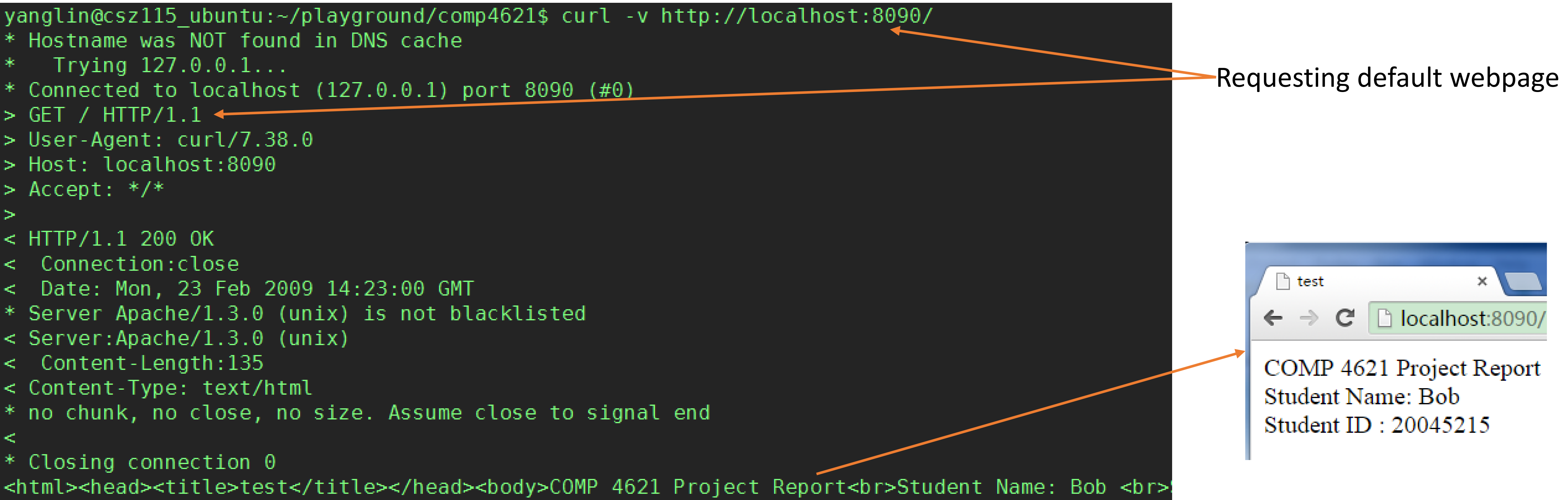
- More references about HTTP range request:
  - http://www.cyberciti.biz/cloud-computing/http-status-code-206-commad-line-test/
  - http://benramsey.com/blog/2008/05/206-partial-content-and-range-requests/

Programming Project

- A simple Web Server:
  - Handle basic request of web page (tutorial #1-3)
  - Handle multiple requests simultaneously (tutorial #3)
  - Handle request of downloading large file with HTTP Range(tutorial #4)

Programming Project

- If client requests a default page, then replies with a simple web page, which shows your name and student ID.



```
yanglin@csz115_ubuntu:~/playground/comp4621$ curl -v http://localhost:8090/
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8090 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.38.0
> Host: localhost:8090
> Accept: */*
>
< HTTP/1.1 200 OK
<  Connection:close
<  Date: Mon, 23 Feb 2009 14:23:00 GMT
* Server Apache/1.3.0 (unix) is not blacklisted
< Server:Apache/1.3.0 (unix)
<  Content-Length:135
< Content-Type: text/html
* no chunk, no close, no size. Assume close to signal end
<
* Closing connection 0
<html><head><title>test</title></head><body>COMP 4621 Project Report<br>Student Name: Bob <br>
```

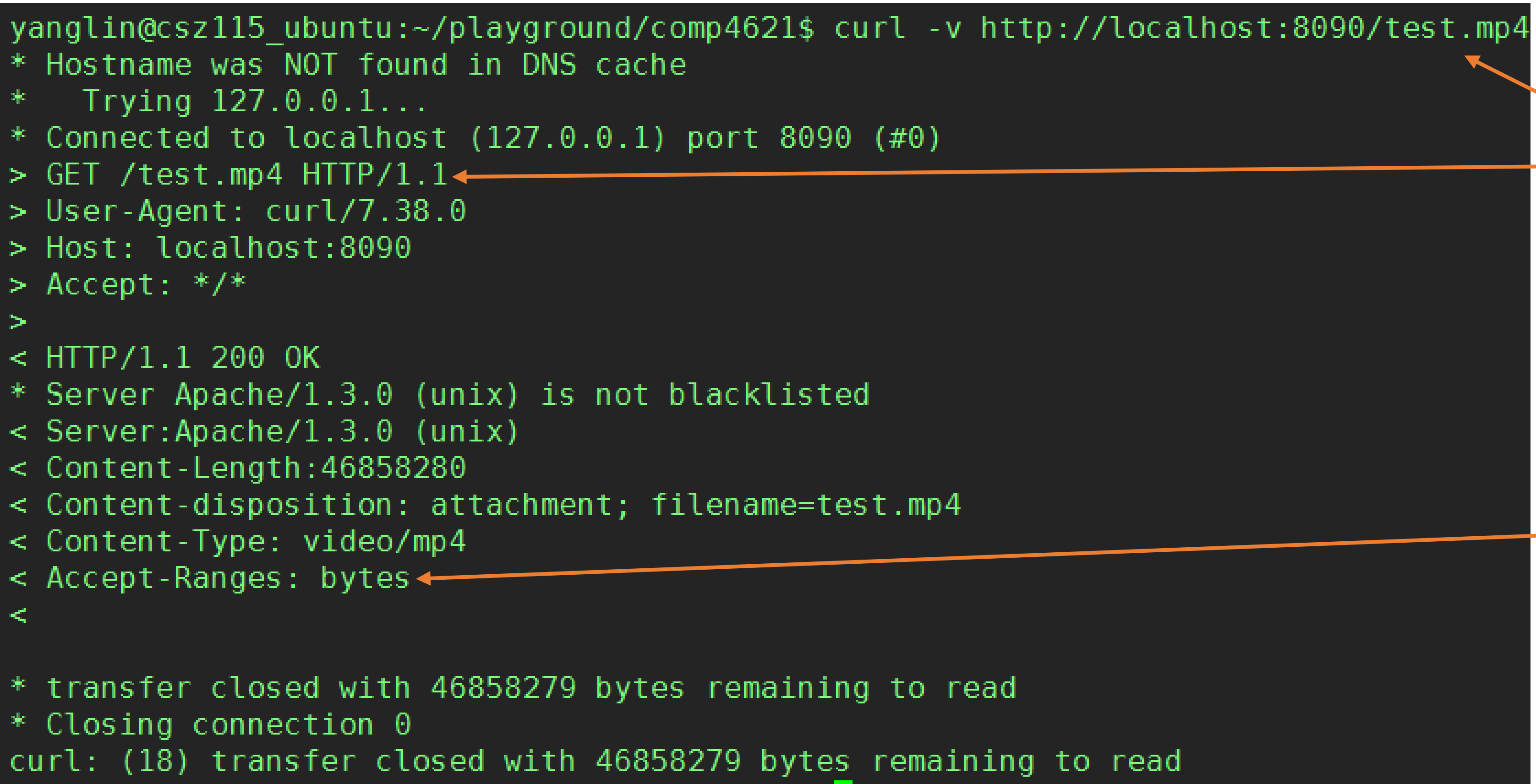
test

localhost:8090/

COMP 4621 Project Report
Student Name: Bob
Student ID : 20045215

Programming Project …

- If the client requests **"test.mp4"** *(hardcode this name in your code),* then show support of HTTP range



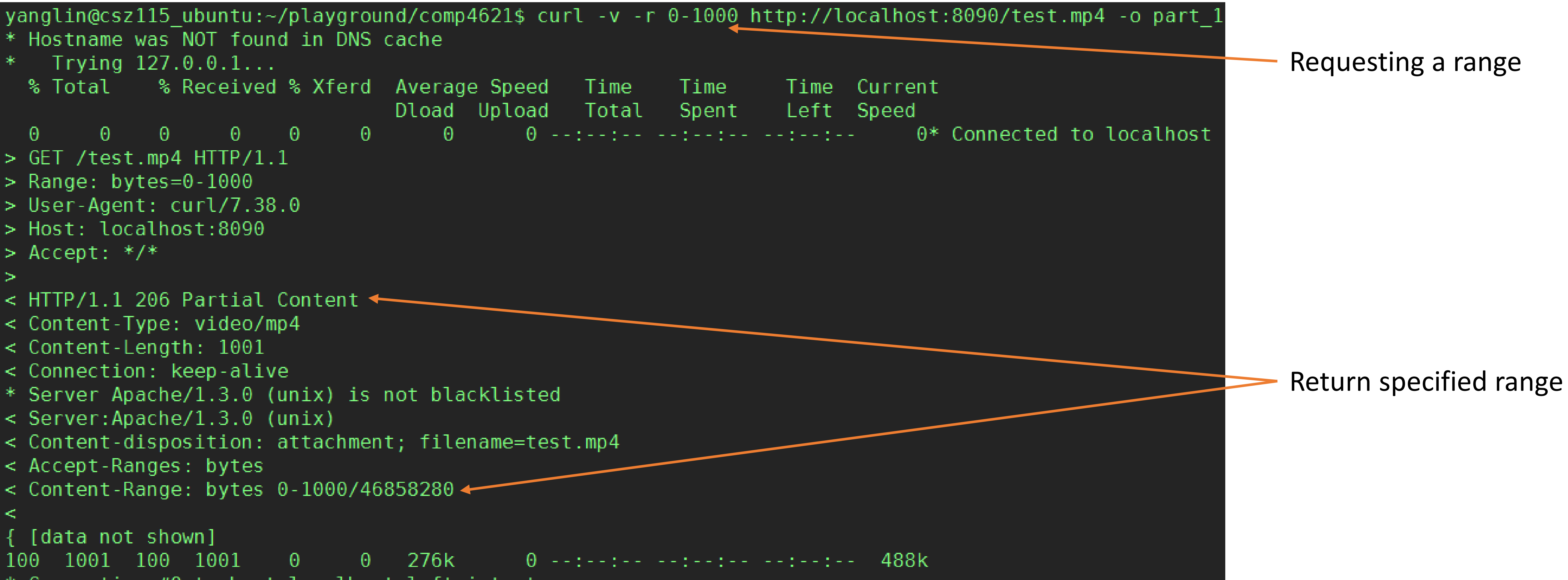
```
yanglin@csz115_ubuntu:~/playground/comp4621$ curl -v http://localhost:8090/test.mp4
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8090 (#0)
> GET /test.mp4 HTTP/1.1
> User-Agent: curl/7.38.0
> Host: localhost:8090
> Accept: */*
>
< HTTP/1.1 200 OK
* Server Apache/1.3.0 (unix) is not blacklisted
< Server:Apache/1.3.0 (unix)
< Content-Length:46858280
< Content-disposition: attachment; filename=test.mp4
< Content-Type: video/mp4
< Accept-Ranges: bytes
<

* transfer closed with 46858279 bytes remaining to read
* Closing connection 0
curl: (18) transfer closed with 46858279 bytes remaining to read
```

Requesting a video file

Indicating support of HTTP range

Programming Project …

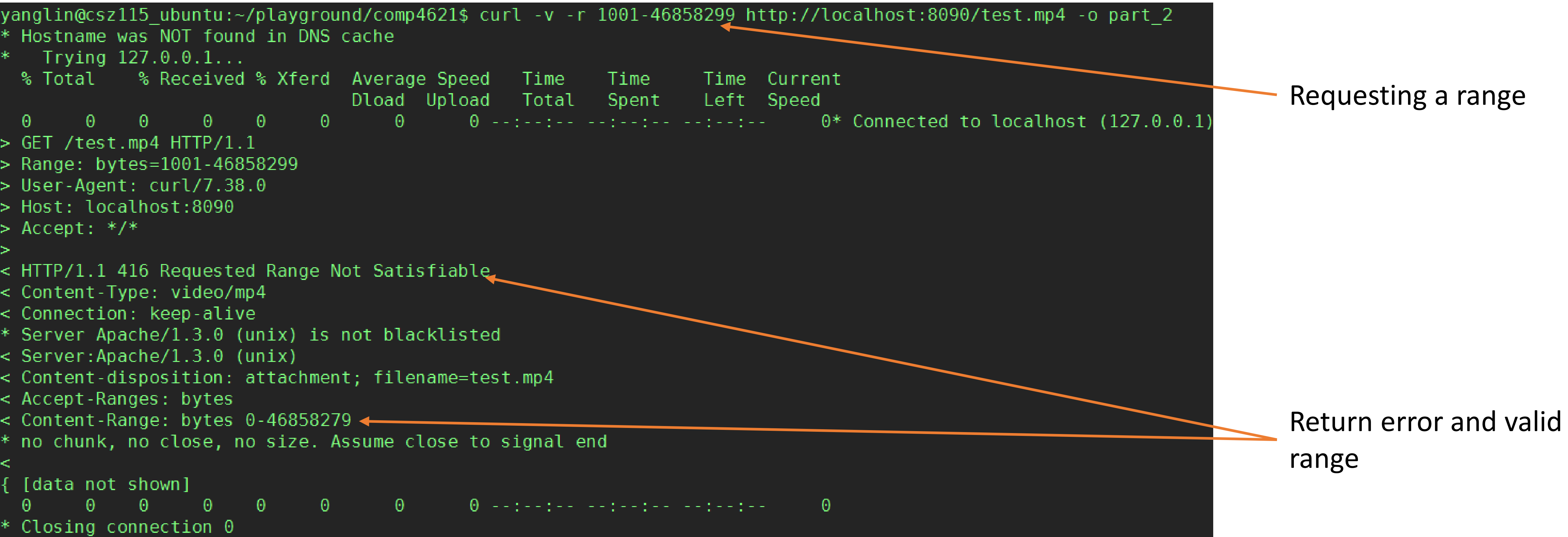
- If the client requests “test.mp4” with a valid range request, then replies specified data range



```
yanglin@csz115_ubuntu:~/playground/comp4621$ curl -v -r 0-1000 http://localhost:8090/test.mp4 -o part_1
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connected to localhost
> GET /test.mp4 HTTP/1.1
> Range: bytes=0-1000
> User-Agent: curl/7.38.0
> Host: localhost:8090
> Accept: */*
>
< HTTP/1.1 206 Partial Content
< Content-Type: video/mp4
< Content-Length: 1001
< Connection: keep-alive
* Server Apache/1.3.0 (unix) is not blacklisted
< Server:Apache/1.3.0 (unix)
< Content-disposition: attachment; filename=test.mp4
< Accept-Ranges: bytes
< Content-Range: bytes 0-1000/46858280
<
{ [data not shown]
100  1001  100  1001    0     0   276k      0 --:--:-- --:--:-- --:--:--  488k
```

Programming Project …

- If client request an invalid range, then return 416 Requested Range Not Satisfiable, use *Content-Range header* to indicate valid range



```
yanglin@csz115_ubuntu:~/playground/comp4621$ curl -v -r 1001-46858299 http://localhost:8090/test.mp4 -o part_2
* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connected to localhost (127.0.0.1)
> GET /test.mp4 HTTP/1.1
> Range: bytes=1001-46858299
> User-Agent: curl/7.38.0
> Host: localhost:8090
> Accept: */*
>
< HTTP/1.1 416 Requested Range Not Satisfiable
< Content-Type: video/mp4
< Connection: keep-alive
* Server Apache/1.3.0 (unix) is not blacklisted
< Server:Apache/1.3.0 (unix)
< Content-disposition: attachment; filename=test.mp4
< Accept-Ranges: bytes
< Content-Range: bytes 0-46858279
* no chunk, no close, no size. Assume close to signal end
<
{ [data not shown]
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
* Closing connection 0
```

Submission Requirement

- A folder named as “name_studentID”, which should include
  - All source code (.java)
  - A two-page project report (Times New Roman, 11pts, single column)
  - No need to submit test.mp4, just hardcode this file name in your source code, *e.g.,*
    ```
    if (strReqTarget.equals("/test.mp4")) { … }
    ```

- The project report should explain:
  - Structure of your code, give explanation for import functions
  - How to handle multiple HTTP requests simultaneously?
  - How to handle HTTP request of webpage?
  - How to handle HTTP Range requests?

Please paste your test result with curl in your report

Marking scheme

- Handle request of web page correctly (20%)
- Handle multiple requests simultaneously (20%)
- Handle request of Range appropriately (30%)
    - Support of range (10%)
    - Handle valid range request (10%)
    - Handle invalid range request (10%)
- Project report (30%)

Implementation

- You may start with webserver example we have implemented in lab tutorial 3
- Some more functions need to implement:
    - Parse HTTP Request
    - Parse Range-related header
    - Implement HTTP range

- **Note:**
    - When you access the test.mp4, please use a relative path, i.e., ./test.mp4
    - Do not package your code

Q&A