南方科技大学

计算机网络实验报告

姓名：王雨童　　学号：**11611808**

专业：计算机科学与技术

实验时间：**2018.09.25**

实验内容：

Assignment 1

Using cURL make GET request to http://httpbin.org/get

Using cURL make POST request to http://httpbin.org/post

Using curl -v to inspect the interaction

Using wiresharkto capture the packet cURL sent.

Write your report.

1. What did you get via cURL?

2. What are the meaning of fields in your request and response headers?

3. Did wiresharkcapture correspond to the cURL request?

Assignment 2

Using asyncio implement a Echo Server, based on Echo Server Multithreading

Assignment 3(CS major only)

Using asyncioimplement a HTTP/1.0 web file browser.

实验步骤：

3.1

1. use cURL to make get request:

```
wangyutongdeMacBook-Pro:~ wangyutong$ curl http://httpbin.org/get
{
  "args": {},
  "headers": {
    "Accept": "*/*",
    "Connection": "close",
    "Host": "httpbin.org",
    "User-Agent": "curl/7.54.0"
  },
  "origin": "116.6.49.95",
  "url": "http://httpbin.org/get"
}
```

2. use cURL to make post request:

```
wangyutongdeMacBook-Pro:~ wangyutong$ curl --data "username=wyt&password=666" -X POST http://httpbin.org/post
{
  "args": {},
  "data": "",
  "files": {},
  "form": {
    "password": "666",
    "username": "wyt"
  },
  "headers": {
    "Accept": "*/*",
    "Connection": "close",
    "Content-Length": "25",
    "Content-Type": "application/x-www-form-urlencoded",
    "Host": "httpbin.org",
    "User-Agent": "curl/7.54.0"
  },
  "json": null,
  "origin": "116.6.49.92",
  "url": "http://httpbin.org/post"
}
```

Use cURL -v to inspect the interaction:

```
wangyutongdeMacBook-Pro:~ wangyutong$ curl --data "username=wyt&password=666" -X POST http://httpbin.org/post -v
Note: Unnecessary use of -X or --request, POST is already inferred.
*   Trying 52.71.139.107...
* TCP_NODELAY set
* Connected to httpbin.org (52.71.139.107) port 80 (#0)
> POST /post HTTP/1.1
> Host: httpbin.org
> User-Agent: curl/7.54.0
> Accept: */*
> Content-Length: 25
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 25 out of 25 bytes
< HTTP/1.1 200 OK
< Connection: keep-alive
< Server: gunicorn/19.9.0
< Date: Sun, 23 Sep 2018 03:20:46 GMT
< Content-Type: application/json
< Content-Length: 412
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Credentials: true
< Via: 1.1 vegur
<
{
  "args": {},
  "data": "",
  "files": {},
  "form": {
    "password": "666",
    "username": "wyt"
  },
  "headers": {
    "Accept": "*/*",
    "Connection": "close",
    "Content-Length": "25",
    "Content-Type": "application/x-www-form-urlencoded",
    "Host": "httpbin.org",
    "User-Agent": "curl/7.54.0"
  },
  "json": null,
  "origin": "116.6.49.98",
  "url": "http://httpbin.org/post"
}
* Connection #0 to host httpbin.org left intact
```

Use wireshark to capture the packet CURL sent

http.host == "httpbin.org" 表达式...

No.	Time	Source	Destination	Protocol	Length	Info
4	0.297457	10.20.61.2	52.22.213.157	HTTP	240	POST /post HTTP/1.1 (application/x-w

```
▶ Transmission Control Protocol, Src Port: 61136, Dst Port: 80, Seq: 1, Ack: 1, Len: 174
▼ Hypertext Transfer Protocol
  ▼ POST /post HTTP/1.1\r\n
    ▶ [Expert Info (Chat/Sequence): POST /post HTTP/1.1\r\n]
      Request Method: POST
      Request URI: /post
      Request Version: HTTP/1.1
    Host: httpbin.org\r\n
    User-Agent: curl/7.54.0\r\n
    Accept: */*\r\n
  ▶ Content-Length: 25\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
    \r\n
    [Full request URI: http://httpbin.org/post]
    [HTTP request 1/1]
    [Response in frame: 6]
    File Data: 25 bytes
▼ HTML Form URL Encoded: application/x-www-form-urlencoded
  ▶ Form item: "username" = "wyt"
  ▶ Form item: "password" = "666"
```

3.2

Coding with python to implement a Echo Server, based on Echo Server Multithreading.

3.3

Coding with python to implement a HTTP/1.0 web file browser.

实验结果：

3.2

启动服务器：

```
wangyutongdeMacBook-Pro:lab3 wangyutong$ python3 lab3_2.py
Serving on ('127.0.0.1', 8080)
```

打开两个终端窗口连接到 127.0.0.1 8080:

```
wangyutong — telnet 127.0.0.1 8080 — 80×24
Last login: Tue Sep 25 11:26:29 on ttys000
wangyutongdeMacBook-Pro:~ wangyutong$ telnet 127.0.0.1 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

```
wangyutong — telnet 127.0.0.1 8080 — 80×24
Last login: Tue Sep 25 11:26:55 on ttys002
wangyutongdeMacBook-Pro:~ wangyutong$ telnet 127.0.0.1 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

分别输入，验证了 echo 功能：

```
wangyutong — telnet 127.0.0.1 8080 — 80×24
Last login: Tue Sep 25 11:26:29 on ttys000
wangyutongdeMacBook-Pro:~ wangyutong$ telnet 127.0.0.1 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
cline wyt
cline wyt
```

```
wangyutong — telnet 127.0.0.1 8080 — 80×24
Last login: Tue Sep 25 11:26:55 on ttys002
wangyutongdeMacBook-Pro:~ wangyutong$ telnet 127.0.0.1 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
client lalala
client lalala
```

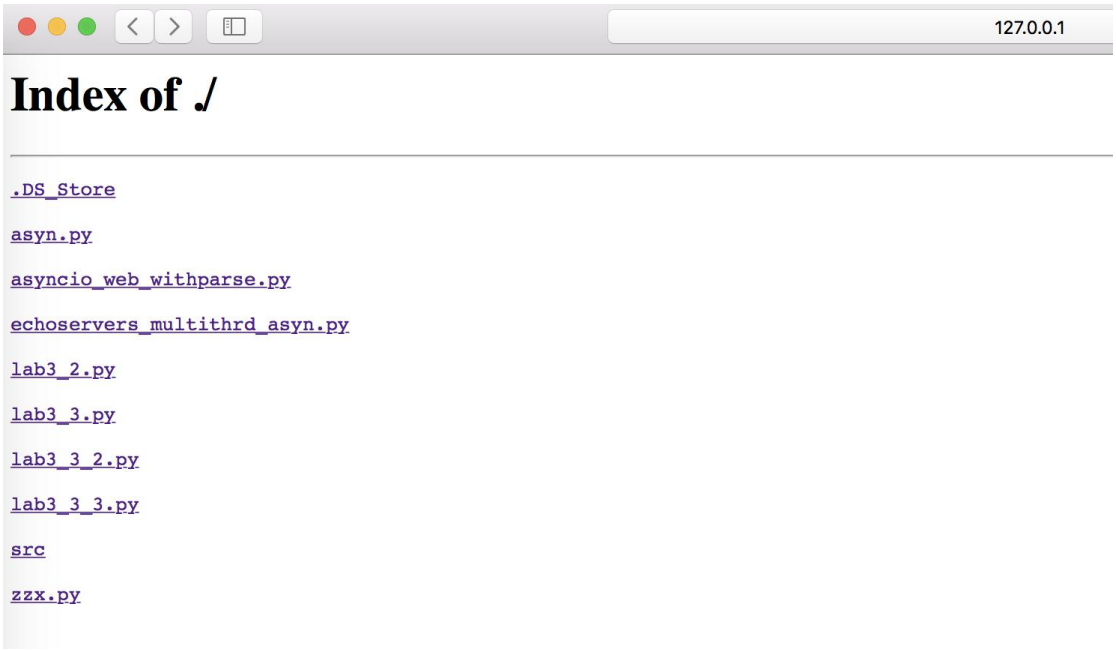
```
lab3 — Python lab3_2.py — 80×24
Last login: Tue Sep 25 11:24:30 on ttys003
wangyutongdeMacBook-Pro:~ wangyutong$ cd /Users/wangyutong/
twork/lab3
wangyutongdeMacBook-Pro:lab3 wangyutong$ python3 lab3_2.py
Serving on ('127.0.0.1', 8080)
b'cline wyt\r\n'
b'client lalala\r\n'
```

3.3

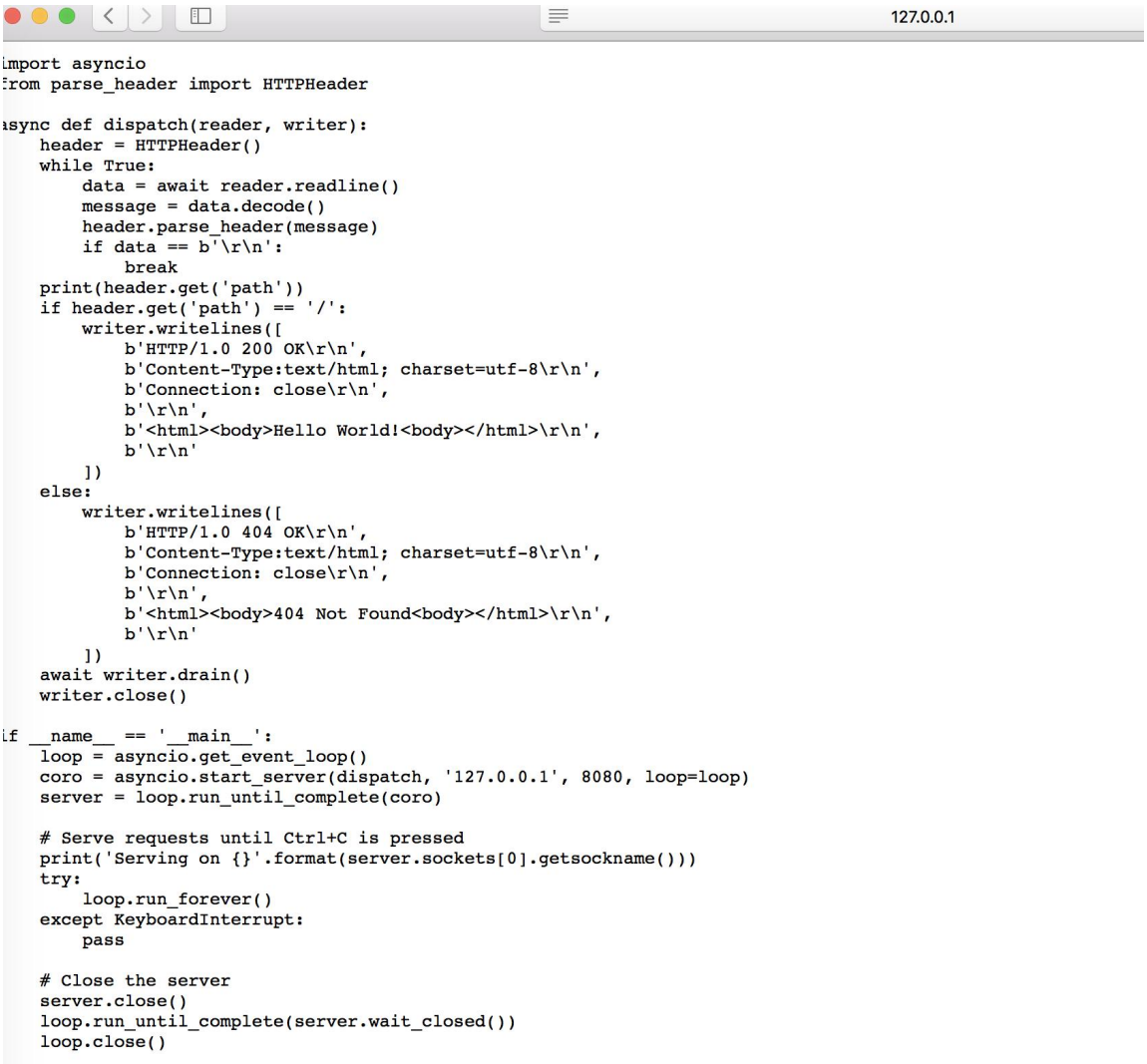
启动服务器:

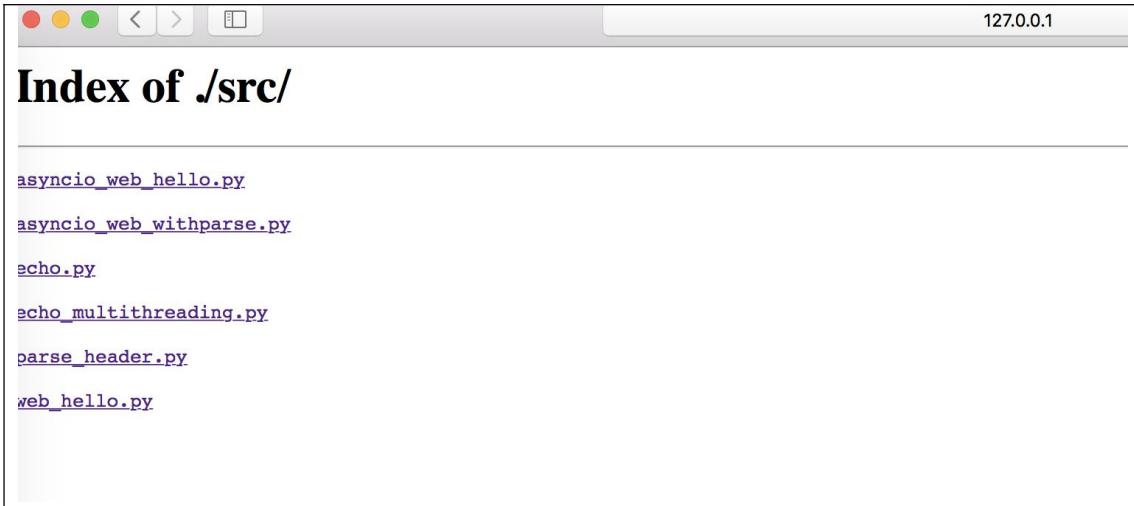
```
wangyutongdeMacBook-Pro:lab3 wangyutong$ python3 lab3_3.py
Serving on ('127.0.0.1', 8080)
```

在浏览器中打开 127.0.0.1 8080:



验证打开文件、文件夹功能：





在终端验证：

i. 传递文件路径 发送 get 请求：

```
wangyutongdeMacBook-Pro:~ wangyutong$ curl http://127.0.0.1:8080/lab3_2.py/
import asyncio

async def echo(reader,writer):
    while True:
        data = await reader.readline()
        print(data)
        if(data == b'\r\n'):
            break
        writer.write(data)
        await writer.drain()
    writer.close()

if __name__ == '__main__':
    loop = asyncio.get_event_loop()
    coro = asyncio.start_server(echo, '127.0.0.1', 8080, loop=loop)
    server = loop.run_until_complete(coro)
    # Serve requests until Ctrl+C is pressed
    print('Serving on {}'.format(server.sockets[0].getsockname()))
    try:
        loop.run_forever()
    except KeyboardInterrupt:
        pass
    # Close the server
    server.close()
    loop.run_until_complete(server.wait_closed())
wangyutongdeMacBook-Pro:~ wangyutong$
```

```
wangyutongdeMacBook-Pro:~ wangyutong$ curl http://127.0.0.1:8080/lab3_2.py/ -v
*   Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to 127.0.0.1 (127.0.0.1) port 8080 (#0)
> GET /lab3_2.py/ HTTP/1.1
> Host: 127.0.0.1:8080
> User-Agent: curl/7.54.0
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Content-Type:text/x-python
< Content-Length:718
< charset=utf-8
< Connection: close
<
import asyncio

async def echo(reader,writer):
    while True:
        data = await reader.readline()
        print(data)
        if(data == b'\r\n'):
            break
        writer.write(data)
        await writer.drain()
    writer.close()

if __name__ == '__main__':
    loop = asyncio.get_event_loop()
    coro = asyncio.start_server(echo, '127.0.0.1', 8080, loop=loop)
    server = loop.run_until_complete(coro)
    # Serve requests until Ctrl+C is pressed
    print('Serving on {}'.format(server.sockets[0].getsockname()))
    try:
        loop.run_forever()
    except KeyboardInterrupt:
        pass
    # Close the server
    server.close()
    loop.run_until_complete(server.wait_closed())
* Closing connection 0
    loop.close()wangyutongdeMacBook-Pro:~ wangyutong$
```

ii. 传递文件路径 发送 head 请求：

```
wangyutongdeMacBook-Pro:~ wangyutong$ curl http://127.0.0.1:8080/lab3_3_3.py/ --HEAD
HTTP/1.0 200 OK
Content-Type:text/x-python
Content-Length:5521
charset=utf-8
Connection: close
```

```
wangyutongdeMacBook-Pro:~ wangyutong$ curl http://127.0.0.1:8080/lab3_3_3.py/ --HEAD -v
*   Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to 127.0.0.1 (127.0.0.1) port 8080 (#0)
> HEAD /lab3_3_3.py/ HTTP/1.1
> Host: 127.0.0.1:8080
> User-Agent: curl/7.54.0
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
HTTP/1.0 200 OK
< Content-Type:text/x-python
Content-Type:text/x-python
< Content-Length:5521
Content-Length:5521
< charset=utf-8
charset=utf-8
< Connection: close
Connection: close

<
* Closing connection 0
```

iii. 传递文件夹路径 发送 get 请求：

```
wangyutongdeMacBook-Pro:~ wangyutong$ curl http://127.0.0.1:8080/src/
<html><head><title>Index of ./src/</title></head>
<body bgcolor="white">
<h1>Index of ./src/</h1><hr>
<pre>
<a href="asyncio_web_hello.py/">asyncio_web_hello.py</a><br>
<a href="asyncio_web_withparse.py/">asyncio_web_withparse.py</a><br>
<a href="echo.py/">echo.py</a><br>
<a href="echo_multithreading.py/">echo_multithreading.py</a><br>
<a href="parse_header.py/">parse_header.py</a><br>
<a href="web_hello.py/">web_hello.py</a><br>
</pre>
</hr>
</body></html>
```

```
wangyutongdeMacBook-Pro:~ wangyutong$ curl http://127.0.0.1:8080/src/ -v
*   Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to 127.0.0.1 (127.0.0.1) port 8080 (#0)
> GET /src/ HTTP/1.1
> Host: 127.0.0.1:8080
> User-Agent: curl/7.54.0
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Content-Type:text/html; charset=utf-8
< Connection: close
<
<html><head><title>Index of ./src/</title></head>
<body bgcolor="white">
<h1>Index of ./src/</h1><hr>
<pre>
<a href="asyncio_web_hello.py/">asyncio_web_hello.py</a><br>
<a href="asyncio_web_withparse.py/">asyncio_web_withparse.py</a><br>
<a href="echo.py/">echo.py</a><br>
<a href="echo_multithreading.py/">echo_multithreading.py</a><br>
<a href="parse_header.py/">parse_header.py</a><br>
<a href="web_hello.py/">web_hello.py</a><br>
</pre>
</hr>
</body></html>
* Closing connection 0
```

iv. 传递文件夹路径 发送 head 请求：

```
wangyutongdeMacBook-Pro:~ wangyutong$ curl http://127.0.0.1:8080/src/ --HEAD
HTTP/1.0 200 OK
Content-Type:text/html
charset=utf-8
Connection: close
```

```
wangyutongdeMacBook-Pro:~ wangyutong$ curl http://127.0.0.1:8080/src/ --HEAD -v
*   Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to 127.0.0.1 (127.0.0.1) port 8080 (#0)
> HEAD /src/ HTTP/1.1
> Host: 127.0.0.1:8080
> User-Agent: curl/7.54.0
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
HTTP/1.0 200 OK
< Content-Type:text/html
Content-Type:text/html
< charset=utf-8
charset=utf-8
< Connection: close
Connection: close
* Closing connection 0
wangyutongdeMacBook-Pro:~ wangyutong$
```

实验分析（包括回答问题）：

- Write your report.
  - What did you get via cURL?
  - What are the meaning of fields in your request and response headers?
  - Did wireshark capture correspond to the cURL request?

3.1

1.Via cURL I get a JSON file including headers, form, url and some other fiels in the file.

2.

i. Request headers：

Accept：指定客户端能够接受的内容类型，这种内容类型用 MIME 类型表示；

*/*代表任意类型

Connection：当 client 和 server 通信时对于长链接如何进行处理；

Connection = close 表明当前正在使用的 tcp 链接在请求处理完毕后会被断掉。

Host：指定请求的服务器的域名和端口号为 httpbin.org

Content-length：请求的内容长度 25 字节

Content-type：请求的与实体对应的 MIME 信息为 application/x-www-form-urlencoded

User-Agent：告诉 http 服务器，客户端使用的操作系统和浏览器的名称、版本为：

curl/7.54.0

ii. Response headers：

Connection：为 keep-alive，告诉对方请求响应完成后 TCP 连接不要关闭，下一次还用该连接继续交流

Server：对方服务器为 gunicorn/19.9.0

Date：日期为 2018 年 9 月 23 日，周日，03:20:46

Content-type: 返回内容的 MIME 类型为 application/json

Content-length：响应体内容的长度为 412 字节

3.

The content which wireshark captures matches with the cURL request. The content of host, user-agent, accept, content-length, content-type are the same, and the content in form are the same.

小结及感悟:

1. 通过第一题，我了解了基本 get、post 请求，并通过查阅资料理解了 http header 中各种属性值的含义。

2. 在写第三题的时候，起初毫无思路，后来仔细分析了样例代码，明白了大体框架。

备注: