

Network PJ: DNS query and HTTP request

[Network PJ: DNS query and HTTP request](#)

[Introduction](#)

[Expected Output](#)

[RFC Documents](#)

[Phrase 1: URL parsing](#)

[Phrase 2: DNS query](#)

[Phrase 3: HTTP request](#)

[Auto Testing](#)

[Submit](#)

Introduction

In this lab, you should implement a simple tool that can download content from the web using http. This tool should:

1. Send **DNS** query to the **DNS** server, and get the resolved IP address
2. Send **HTTP** request to the **HTTP** server, and bring the result to **stdout**.

Each phrase has basic and extra parts. You **MUST** implement all the basic parts. For the extra parts, you can try some of them. You will get full credit for the lab if you achieve highest points in the class. You will get 60% of the credit for the lab if you implement all the basic parts correctly.

Please fetch the document from

```
ftp://10.132.141.33/classes/14/161 计算机网络/PROJECT
```

or

```
https://github.com/htc550605125/network-pj
```

You can use **any program language** in this lab on **Windows** or **OSX** or **Linux**, but you **should not use any library** for **DNS** or **HTTP** if not specifically indicated. You should implement them with your code only.

You should provide all of your source code and a document describing how to build and run your program. Please also provide the platform you are using, and what you have implemented, in the document.

Generally, your program should be named **labget** and accept one argument, the URL:

```
> labget http://www.zhihu.com
```

If you want to use java, please provide(build) jar, maven is suggested:

```
> java -jar labget.jar http://www.zhihu.com
```

WARNING! PLAGIARISM IS NOT ALLOWED OR YOU WILL GET ZERO CREDIT FOR THIS LAB GUARANTEED. DO NOT SHARE ANY CODE WITH YOUR CLASSMATE.

Expected Output

If there is no error, your program should send the result to **stdout**. You can print your log to **stderr**. Please at least include the **resolved IP address and HTTP request/response header** in **stderr**. The exit code of your program should always be 0.

If anything goes wrong, just leave **stdout** empty and send error message to **stderr**. Please set timeout to **5 seconds** for each network operation.

RFC Documents

Everything you need to know about the **DNS** protocol and **HTTP** protocol is defined in RFC. **Request for Comments (RFC)** is a type of publication from the **Internet Engineering Task Force (IETF)** and the **Internet Society (ISOC)**, the principal technical development and standards-setting bodies for the Internet. They are your friends in this lab. Although they all seem to be very long, you only need a little part of it. Be patient and find out the part you need in RFC while reading.

In this lab, we will touch two protocols: **DNS** and **HTTP**. For **DNS** part, you need [RFC 1035](#) . For **HTTP** part, you need [RFC 7230](#) .

Phrase 1: URL parsing

The HTTP URL scheme is defined in [RFC 7230 Section 2.7](#) . If your program cannot parse the given URL, just print out the error message to **stderr** and exit.

Hint: use regex for parsing.

In this part, your program **MUST** support:

1. (5 points) Handle URL path correctly.
2. (5 points) **HTTP** with default port (80) or custom port.

Extra:

1. (5 - 20 points) **HTTPS** scheme. You can use **TLS** library for this part, but you can get a big bonus if you implement **TLS** yourself. And, if you can, try to implement the server certificate validation.

Hint: Use regex for parsing. Read RFC carefully for URL scheme definition.

Phrase 2: DNS query

Everything is defined in RFC 1035 and RFC 3596(for ipv6). If you are confusing about the RFC document, see an example [here](#) . Please send dns query to 202.120.224.26.

In this part, your program **MUST** support:

1. (30 points) **DNS** query for ipv4 address (A NAME). Please use custom udp socket. If you have difficulty implementing this part, you can use the system dns resolver, but you will not get full points for this part.

Extra:

1. (5 - 20 points) ipv6 support (AAAA NAME). If you want to support ipv6, please try ipv4/ipv6 simultaneously and return the first successful result. Note that there are several conditions: the site doesn't have AAAA record; the site has a AAAA record but cannot connect to the site with ipv6; the running environment doesn't support ipv6.

Note: Almost all the Fudan wireless have ipv6 access. Your dorm's wired port with static ip also have ipv6 access.

Phrase 3: HTTP request

See RFC 7230. Please send request/response **startline** and **headers** to **stderr**.

In this part, your program **MUST** support:

1. (10 points) **HTTP 1.1 GET** request
2. (10 points) **HTTP 1.1** response parsing

Extra:

1. (5 points) **HTTP 1.1 GET** with appropriate headers. If you don't know what is appropriate for a HTTP request, see what chrome sends, and read RFC.
2. (10 - 15 points) Follow **HTTP Redirect** (301, 302). Handle cookie while redirecting.
3. (5 points) Handle chunked transfer coding. See [section 4.1 in RFC 7230](#) .
4. (10 points) Handle gzip coding. See [section 4.2.3 in RFC 7230](#). You can use library for this part.
5. (10 points) Multi-thread downloading with range header. You can detect the total length of the content, if it is bigger than 10M, use multi-thread download to save the content to a temp file, then output the file to **stdout**.

Note: Read the RFC carefully! The behaviour of your code will be concerned on grading.

Auto Testing

This lab provides a python script for testing. It covers some of the functions above. However, the points you get is not only based on this test.

To run the test, install python3, modify `test.py` line 8 to the command that launches your program. Then, run the script using `python test.py`. Your program are not required to pass all the test cases. If you want to skip some of the test cases, add

```
@unittest.skip("Not implemented")
```

before the test function.

Submit

Compress all of your code and documents into a **zip** file named by your student ID. Upload the zip file to

```
ftp://10.132.141.33/classes/14/161 计算机网络/WORK_UPLOAD/PJ1
```

before 2017/1/1 23:59:59.

If you have any question for this lab, feel free to contact me via

- Wechat: htc550605125
- QQ: 550605125
- Telegram: +86 15221920753
- Email: 13302010023@fudan.edu.cn