

Computer Network Project 3 Simple Proxy Server

CSI4106-01

Fall, 2020

(Difficulty ★★★★★)

Prelim.

The project 3 consists of

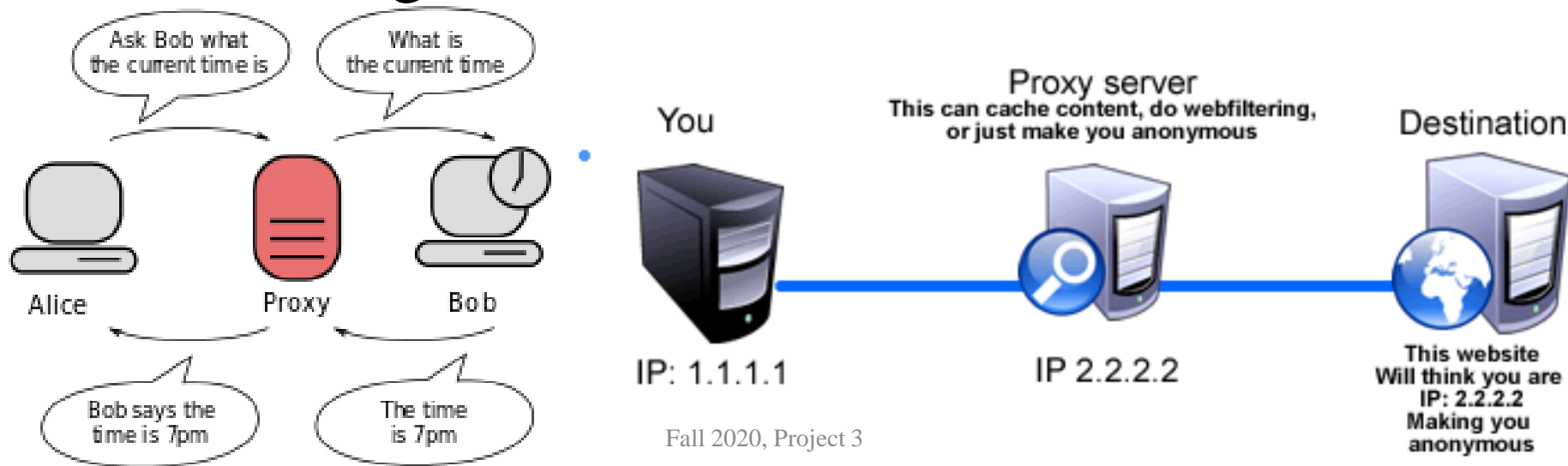
- Mandatory Assignment (100 pts)
- Additional Assignment (+30 pts)
- Bonus score

Prelim.

Before you do this
project, you must be
fully aware of
“Project Policy Notice”

What is a proxy server?

- An intermediary which deals with requests from clients seeking resources of other servers.
- Resides anywhere, between YOU and destination.
- The fundamental function: intercepting and forwarding traffic.

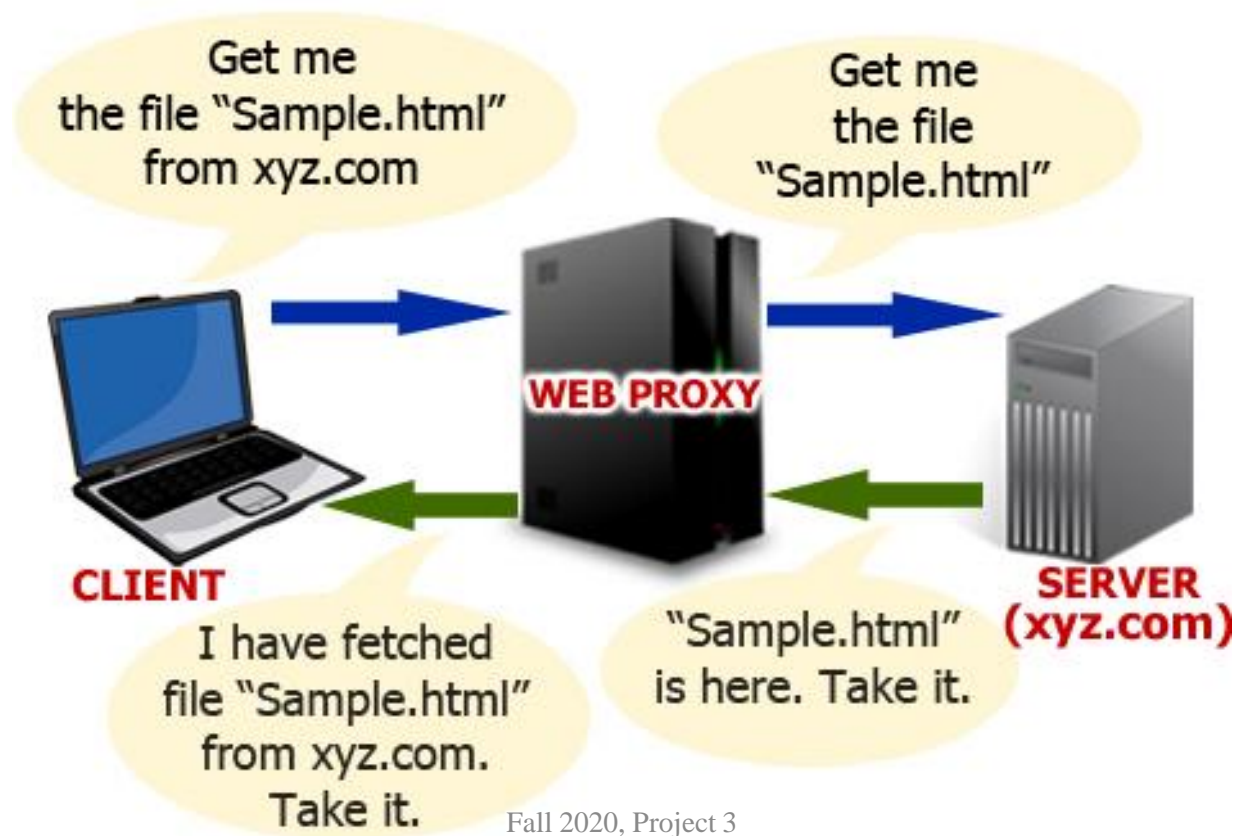


The uses of proxy servers

- Filtering specific data (to block or redirect them)
 - Filtering of encrypted data
 - Bypassing filters / Censorship / Falsification
 - Translation (Localization for different markets)
- Logging or eavesdropping for monitoring traffic
 - Transparent Proxy
- Improving performance (Cache Server)
- *(e.g.) SSL/DNS/CGI/HTTP proxy*
- *(further...) Reverse, Anonymous, Tunneling...*

How a http proxy server works?

- **Transparent Proxy on HTTP**



What is the Cache Proxy server?

- https://en.wikipedia.org/wiki/Proxy_server
- A nice example: CDN Service.
- A cache server can be implemented anywhere.
- Cache Position (Conceptually, Client to Server)
 - (Local – Forward) – Gateway – (Reverse – Server)
- **Intermediary Proxy Server**
 - **Forward / Reverse and explicit servers (CDN's cache servers everywhere)**

Mandatory Assignment (100pts)

- Write a code of http **transparent** proxy server
- Implement Required Functions (1)-(3)
 - *Managing sockets with multi-thread*
 - *URL Filtering*
 - *Image Filtering*
- *Your proxy server is running in background*
- *Your proxy server may be so slow...*
 - *Make proxy server supporting multi-thread or cache*
- **Follow the usage format below**

(Format) `./run.sh port`

(Example) `./run.sh 9001`

Starting proxy server on port 9001

Required Functions

(1) Managing Two Sockets!

- **Question: Why does it need two sockets??**
- One socket for receiving Client Request!
- Another socket for fetching Response of the client request.
- You need to forward Requests/Responses.
- Make your proxy server **multithreading-supported**
- *Hint: Where should requests/responses go?*

Required Functions

(2) URL Filtering and Redirection

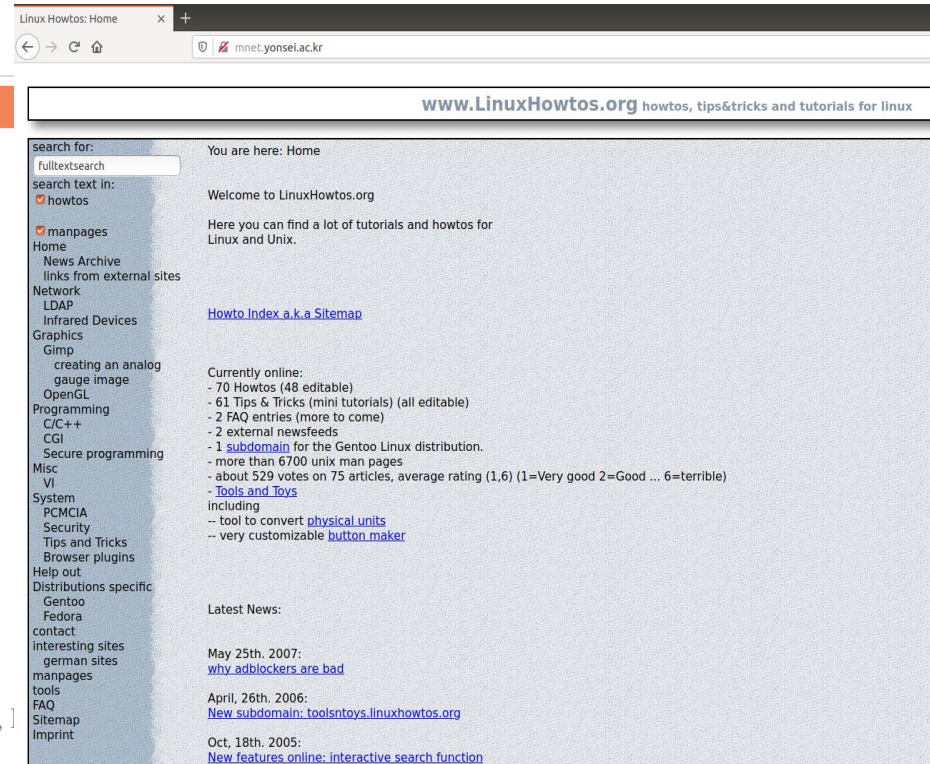
• **Goal: Substitute original URL to designated URL**

▪ Implement it by using flag variable or using a function

Example) Redirect any URL containing “yonsei” to www.linuxhowtos.org

```

[CLI connected to 127.0.0.1:58118]
[CLI ==> PRX --- SRV]
> GET http://mnet.yonsei.ac.kr/
> Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0)
Gecko/20100101 Firefox/83.0
[SRV connected to www.linuxhowtos.org:80]
[CLI --- PRX ==> SRV]
> GET www.linuxhowtos.org
> Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0)
Gecko/20100101 Firefox/83.0
[CLI --- PRX <== SRV]
> 200 OK
> text/html; charset=utf-8 4238bytes
[CLI <== PRX --- SRV]
> 200 OK
> text/html; charset=utf-8 4238bytes
[CLI disconnected]
[SRV disconnected]
```

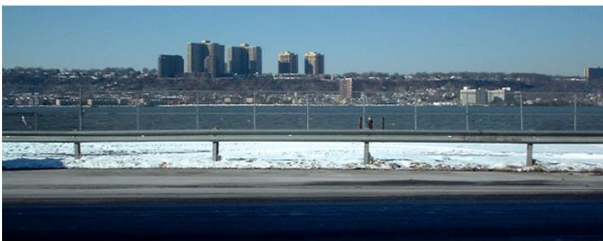


Required Functions

(3) Image filtering

- [http://\[URL\]?image_off](#)
 - With this command, all requests whose domain contains **image** will be dropped.
- To stop it: [http://\[URL\]?image_on](#)

Sample Web Page



A random photo.

[Frank da Cruz](#)

Sat Jan 17 12:07:32 2004

Updated 26 May 2019 for HTML5.

CONTENTS

1. [Creating a Web Page](#)
2. [HTML Syntax](#)
3. [Special Characters](#)
4. [Converting Plain Text to HTML](#)
5. [Effects](#)
6. [Lists](#)
7. [Links](#)
8. [Tables](#)
9. [Viewing your Web page](#)
10. [Installing your Web Page on the Internet](#)
11. [Where to go from here](#)

Sample Web Page

A random photo.

[Frank da Cruz](#)

Sat Jan 17 12:07:32 2004

Updated 26 May 2019 for HTML5.

CONTENTS

1. [Creating a Web Page](#)
2. [HTML Syntax](#)
3. [Special Characters](#)
4. [Converting Plain Text to HTML](#)
5. [Effects](#)
6. [Lists](#)
7. [Links](#)
8. [Tables](#)
9. [Viewing your Web page](#)
10. [Installing your Web Page on the Internet](#)
11. [Where to go from here](#)

You can create a Web page on your desktop computer but nobody can see it but you. If your want other people to be able to see your Web pages, you need an account a computer that has a Web server. Nowadays most people have their own computers on their desks, but normally they don't have Web servers and anyway you don't want the whole world coming into your desktop computer to see your web page because (a) it's not designed for that, and (b) who

Interface

Logging on Command Line

- When a request from a client arrived, your proxy server should print out the precise logs.

Starting proxy server on port 9001

```
-----
1 [Conn:      1/ 20]
[ X ] URL filter | [ X ] Image filter

[CLI connected to 127.0.0.1:40312]
[CLI ==> PRX --- SRV]
> GET http://www.columbia.edu/~fdc/picture-of-something.jpg
> Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101
Firefox/83.0
[SRV connected to www.columbia.edu:80]
[CLI --- PRX ==> SRV]
> GET http://www.columbia.edu/~fdc/picture-of-something.jpg
> Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101
Firefox/83.0
[CLI --- PRX <== SRV]
> 200 OK
> image/jpeg 44926bytes
[CLI <== PRX --- SRV]
> 200 OK
> image/jpeg 44926bytes
[CLI disconnected]
[SRV disconnected]
-----
```

```
2 [Conn:      3/ 20]
[ X ] URL filter | [ X ] Image filter
.....
```

* ? is 0 or X (indicator)

Starting proxy server on port **PORT_NUMBER**

```
-----
NO [Conn:      thread #/  total_thread]
[ ? ] URL filter | [ ? ] Image filter

[CLI connected to CLI_IP:CLI_PORT]
[CLI ==> PRX --- SRV]
> Request_1st_Line_from_Client
> User-Agent_1st_Line_from_Client
[SRV connected to DST_DOMAIN:DST_PORT]
[CLI --- PRX ==> SRV]
> Request_1st_Line_to_Server
> User-Agent_1st_Line_from_Client
[CLI --- PRX <== SRV]
> Response_Status_Code_from_Server
> Response_MIME_Type Response_Sizebytes
[CLI <== PRX --- SRV]
> Response_Status_Code_to_Client
> MIME_Type_to_Client Size_to_Clientbytes
[CLI disconnected]
[SRV disconnected]
```

Background

- TCP (socket programming)
- HTTP Packet (request and response)
- Implementation of Multi-thread in Project 2
- Basic idea of HTML tag
- **Browser' Proxy Setting (How to use proxy)**
 - For this project, you must manually set up your browser's proxy server.(as localhost and custom port)

Directions

- Language: **C or Python3**
- OS: **Ubuntu 18.04** or higher
- You must use only *internal* libraries.
 - ***Any 3rd party framework: NOT ALLOWED***

We will test your code as follows

- Work perfectly during evaluation **with no error** (55pts)
 - Managing sockets with multithread (15pts)
 - URL filtering & Redirection (15pts)
 - Image filtering (15pts)
 - Format of interface (10pts)

If you do not implement the following, you will get a deduction on your score

- **Run with custom Port**
- **Feature socket-reuse (port reuse)**
- **Be terminated by only Ctrl+C**
- **Close the sockets (by netstat)**

Your report must include

- **Introduction/Reference (3pts)**
 - Software environment, programming language you used, version, reference and so on.
- **Flow chart or Diagram (10pts)**
 - Must show the logic of your program
 - Focus on describing how your client and server work.
- **Snapshots for each function** to show your codes are working well. (5pts)

Your report must include (cont'd)

- **Logical explanations block by block in detail. (20pts)**
 - It is different with brief comments in your source code!!!
 - In your report, write what the blocks do and why you implemented those functions.
- **Compare the performance with applying Multi thread and before in chart (7pts)**
 - Performance criterion (Elapsed Time, Memory Use..)

Additional Assignment(+30pts): proxy server with Cache

- This is a **supplementary assignment**
 - If you get poor grades on other assignments, this additional assignment can help
- You need to **add the cache feature to the proxy server** of the Mandatory Assignment.

Cache proxy server

- You already know what a Cache proxy server is through the homework assignment.
- Cache Replacement Policy
 - We use **LRU (Least Recently Used)** in this project
- Follow the usage format below

```
(Format)  ./run.sh port [-size maxSize]
```

```
(Example) ./run.sh 9001 -size 16
```

maxSize: Maximum Total Cache Size (Unit: MiB)

Requirement of cache proxy server

(1) update logging about cache

- In addition to the existing log, your proxy server should also print information about the cache
- Example 1:

```
15 [Conn:      5/ 20] [Cache: 0.07/10MB] [Items: 3]
[ X ] URL filter | [ X ] Image filter

[CLI connected to 127.0.0.1:40312]
[CLI ==> PRX --- SRV]
  > GET http://www.columbia.edu/~fdc/sample.html
  > Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0)
Gecko/20100101 Firefox/83.0
[SRV connected to www.columbia.edu:80]
##### CACHE HIT #####
[CLI <== PRX --- SRV]
  > 200 OK
  > text/html 10147bytes
[CLI disconnected]
[SRV disconnected]
```

• Example 2:

```

18 [Conn:      6/ 20] [Cache: 0.04/5MB] [Items: 6]
[ X ] URL filter | [ X ] Image filter

[CLI connected from 127.0.0.1:40274]
[CLI ==> PRX --- SRV]
  > GET http://help.websiteos.com/websiteos/htmlpage.jpg
  > Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101
Firefox/83.0
[SRV connected to help.websiteos.com:80]
##### CACHE MISS #####
[CLI --- PRX ==> SRV]
  > GET http://help.websiteos.com/websiteos/htmlpage.jpg
  > Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101
Firefox/83.0
[CLI --- PRX <== SRV]
  > 200 OK
  > image/jpeg 34884bytes
##### CACHE REMOVED #####
  > http://help.websiteos.com/websiteos/example_of_a_simple_html_page.htm
0.00MB
  > This file has been removed due to LRU!
##### CACHE REMOVED #####
  > http://help.websiteos.com/websiteos/whtopic.js 0.01MB
  > This file has been removed due to LRU!
##### CACHE ADDED #####
  > http://help.websiteos.com/websiteos/htmlpage.jpg 0.03MB
  > This file has been added to the cache
  > Current 5 items cached,  0.04/5MB
#####
[CLI <== PRX --- SRV] @ 01:40:25.891
  < 200 OK
  < image/jpeg 34884bytes
[SRV disconnected]

```

Requirement of cache proxy server

(2) Save a log file

- **log.txt file stores cache logs.**
- **This is an example format.**
- **You can modify the format if you want.**
- **But this does not mean you can omit detail information!!**

log.txt example

...

14 cache hit

http://help.websiteos.com/websiteos/whtopic.js 0.01MB

start current situation

[Cache: 0.07/0.10MB] [Items: 13]

1:data http://diptera.myspecies.info/sites/all/libraries/mediaelement/build/mediaelement-and-player.min.js?v=2.1.6

size 0.01MB

counter 00:49:50.482

...

13:data http://www.columbia.edu/~fdc/picture-of-something.jpg

size 0.04MB

counter 01:40:24.844

end current situation

Additional Assignment (+30pts)

- **Implementation (20pts)**
 - Update logging about cache (10pts)
 - Save a log file (10pts)
- In your **report..** (10pts)
- **Brief summary of flow charts and block by block explanation about cache part (5pts)**
- **Compare the performance with applying cache and before in chart (5pts)**
 - Performance criterion (Elapsed Time, Memory Use..)

Deliverable

- **Only one zip file of “YourID_p3.zip”**
 - **If your ID is 2020147123, 2020147123_p3.zip should be your deliverable file name.**
- **In the zip file only the two/three files must be included without any folder**
 - **report.pdf**
 - **run.sh**
 - **project.py or project.c**
 - **if you use C language, include `compile.sh` as well**
 - **Files with different format/name will be a deduction [2pts each]**

- **DUE DATE**

12/Dec/2020 23:55:00 KST

No exception for exceeding deadline

- **Delay Policy**

-33%pts for ~13/Dec 23:55:00

-66%pts for ~14/Dec 23:55:00

-100%pts for 14/Dec 23:55:00~

**You agree with the following statement
by submitting your assignment on YSCEC**

**Ctrl+C and Ctrl+V
is not Code Referencing,**

Plagiarizing = 0pts = Fail

No exception for any kinds of cheating and copying

Score Policy : *Max. 100pts*

1	Not submitted / not working / missing files	0 pts
2	Overdue → Delay	-33% pts/day
3	The rules or directions whose scores are not specified/ are not followed	-10 pts/rule
4	Any 3 rd party framework is used	0 pts
5	Plagiarizing / Over-implementation (Any kinds of Suspicion of Code-copy)	0 pts
6	Impolite Report / Lack of Comments	0 pts / -50 <u>u</u> % pts

Bonus Score Policy: *+30pts*

1	Not submitted / not working / missing files	0 pts
2	Overdue → Delay	-33% pts/day
3	Any 3 rd party framework is used	0 pts
4	Plagiarizing / Over-implementation (Any kinds of Suspicion of Code-copy)	0 pts
5	Impolite Report / Lack of Comments	0 pts / -50 <u>u</u> % pts