Multi-Threaded Web Proxy

A project report for the course

CSCE 5580 Computer Networks

Spring 2015

By

Group - 1

Ramachandra Reddy B

Anuj Bhalotiya

Jeremy Henry

Under the guidance of

Dr. Enkh – Amgalan Evan Baatarjav

University of North Texas

Denton, TX

April 2015

Abstract

We designed a web proxy server that receives an HTTP request for an object from a browser, it generates a new HTTP request for the same object and sends it to a remote server that is hosting the requested object. The proxy will be muti-threaded, so it will be able to handle multiple requests at the same time. The proxy also has URL blacklisting, content filtering and Cache functionalities.

Contents

List of Figures

1 Introduction or Problem Definition

Our aim is to implement a multi-threaded web proxy server, that receives a HTTP request from the browser creates a new request for the same object and sends it to the remote server that hosts the object. The response from the remote server is processed based on requirements and the result is displayed in the browser.

1.1 System Functionalities

- Filter URL’s that are blacklisted
- Filter words that are considered as inappropriate
- Cache the requested URLs’

1.2 System Architecture

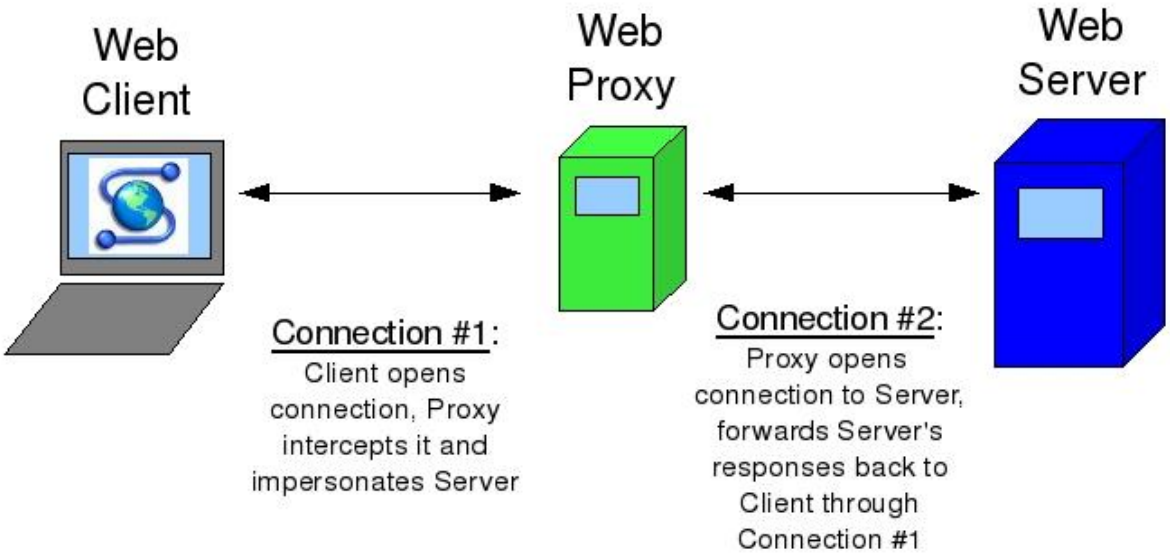

Web Client
Web Proxy
Web Server
Connection #1: Client opens connection, Proxy intercepts it and impersonates Server
Connection #2: Proxy opens connection to Server, forwards Server's responses back to Client through Connection #1

Figure 1: Web Proxy Architecture

The Figure 1 depicts the overview of a web proxy architecture.

2 Implementation

As project is based on client server model, communication takes place through socket between client and server .For simultaneous multiple client communication multi-threading is used and each thread takes care of individual clients. Each request URL is parsed for allowing only valid requests. Requests are verified against blacklists and the responses are filtered for avoiding inappropriate content.

2.1 Programming Tools

Programming Language: C
Communication between two machines: Sockets, multi-threading.
IDE: XCode.
Version Control: GitHub

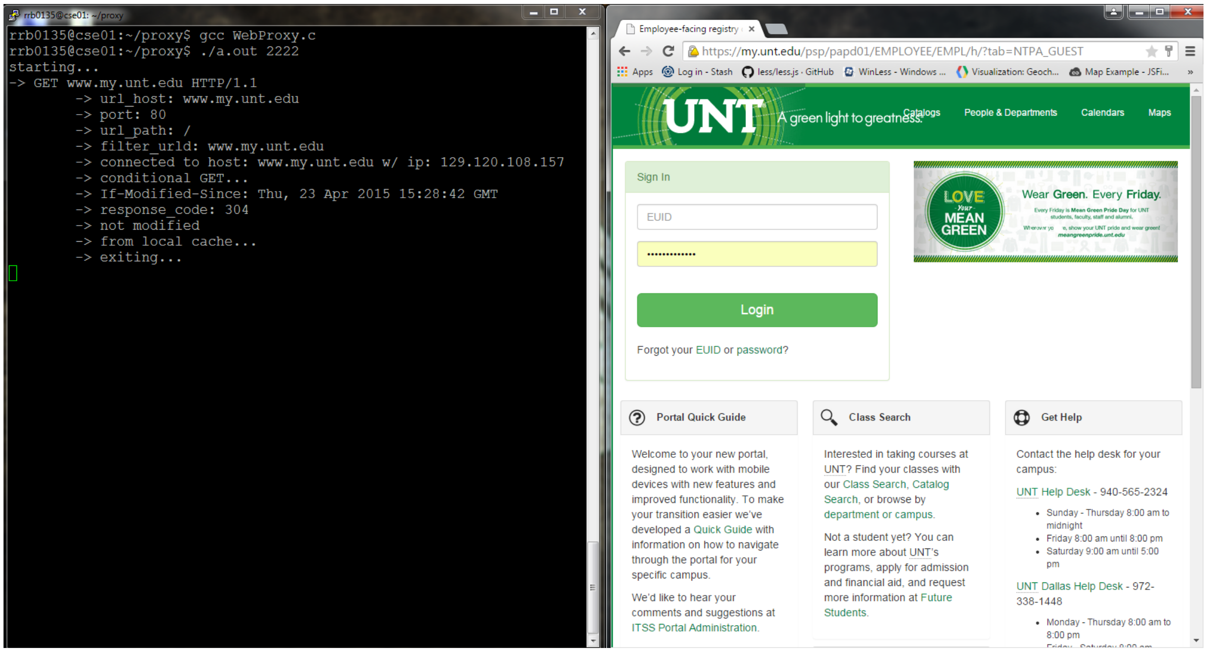
2.2 Execution setup

For execution of the project kindly refer **CNP README.txt** attached with the package.

2.3 Approach for each functionality

Below is the demonstration for each functionality that is implemented.

2.3.1 Proxy Server



```
rrb0135@cse01:~/proxy$ gcc WebProxy.c
rrb0135@cse01:~/proxy$ ./a.out 2222
starting...
-> GET www.my.unt.edu HTTP/1.1
        -> url_host: www.my.unt.edu
        -> port: 80
        -> url_path: /
        -> filter_urld: www.my.unt.edu
        -> connected to host: www.my.unt.edu w/ ip: 129.120.108.157
        -> conditional GET...
        -> If-Modified-Since: Thu, 23 Apr 2015 15:28:42 GMT
        -> response_code: 304
        -> not modified
        -> from local cache...
        -> exiting...
```

https://my.unt.edu/psp/papd01/EMPLOYEE/EMPL/h/?tab=NTPA_GUEST

Forgot your EUID or password?

Welcome to your new portal, designed to work with mobile devices with new features and improved functionality. To make your transition easier we've developed a Quick Guide with information on how to navigate through the portal for your specific campus.

We'd like to hear your comments and suggestions at ITSS Portal Administration.

Interested in taking courses at UNT? Find your classes with our Class Search, Catalog Search, or browse by department or campus.

Not a student yet? You can learn more about UNT's programs, apply for admission and financial aid, and request more information at Future Students.

Contact the help desk for your campus:

UNT Help Desk - 940-565-2324

- Sunday - Thursday 8:00 am to midnight
- Friday 8:00 am until 8:00 pm
- Saturday 9:00 am until 5:00 pm

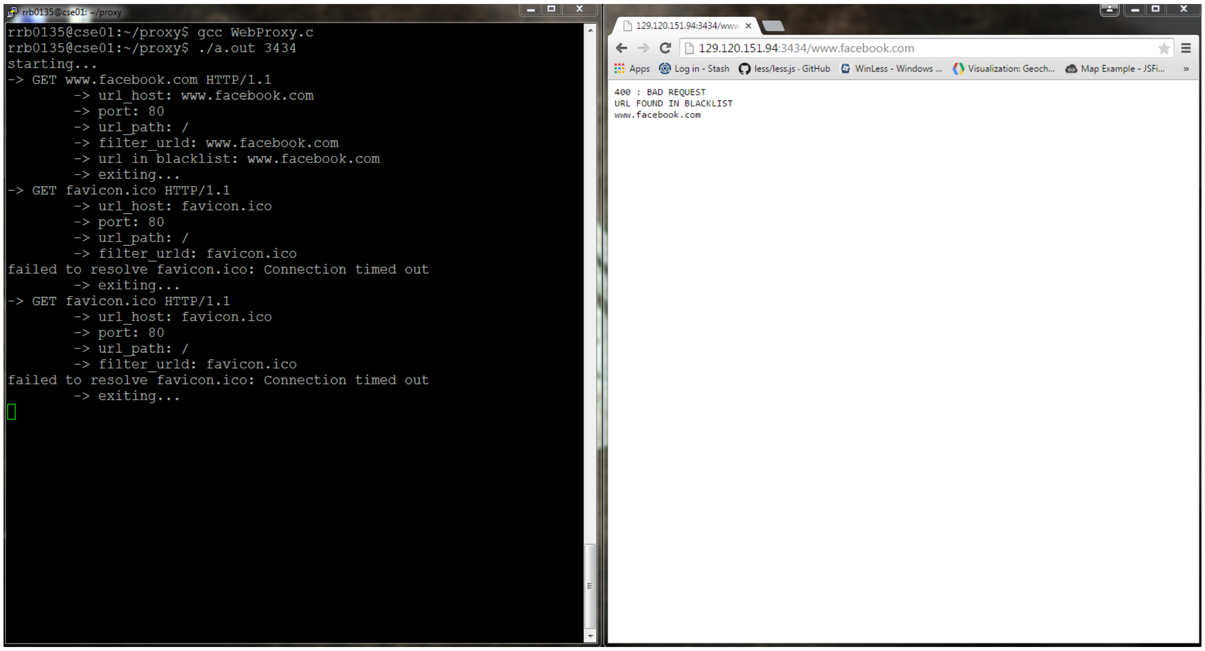
UNT Dallas Help Desk - 972-338-1448

- Monday - Thursday 8:00 am to 8:00 pm

Figure 2 Web Proxy

2.3.2 URL Blocking

The request is parsed and checked with an array of blacklisted URLs'. If the requested URL is found in the array it is blocked and request is not sent. An error message is displayed on the browser as depicted in figure 3



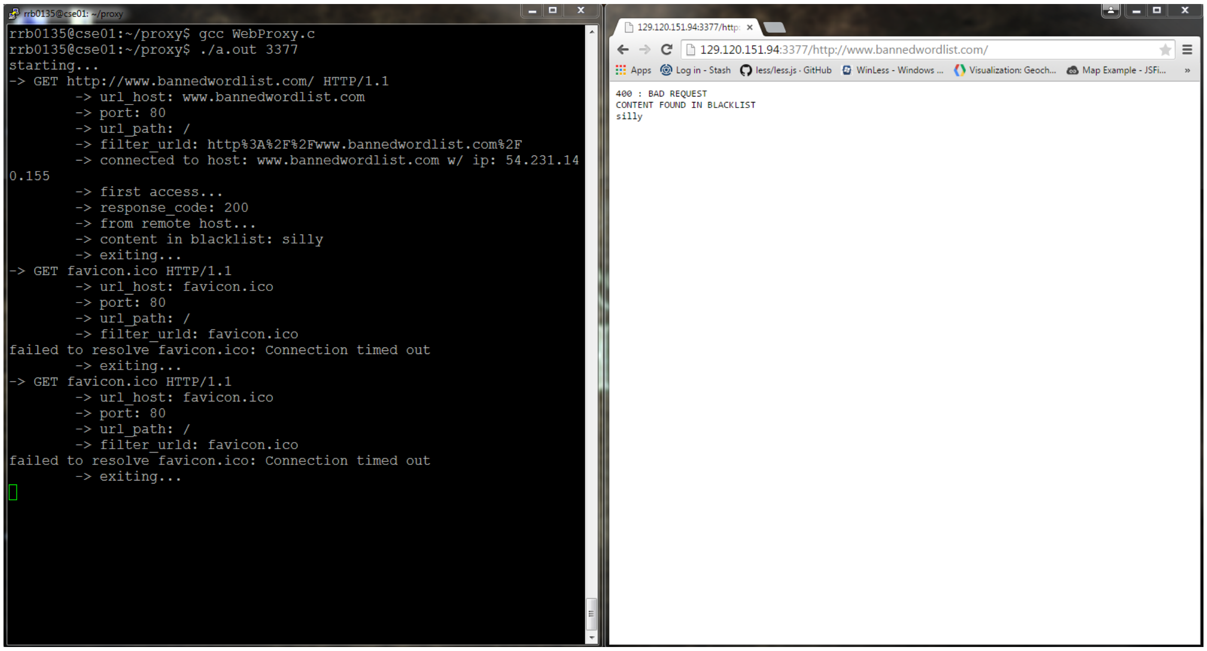
```
rrb0135@cse01:~/proxy$ gcc WebProxy.c
rrb0135@cse01:~/proxy$ ./a.out 3434
starting...
-> GET www.facebook.com HTTP/1.1
        -> url_host: www.facebook.com
        -> port: 80
        -> url_path: /
        -> filter_urld: www.facebook.com
        -> url in blacklist: www.facebook.com
        -> exiting...
-> GET favicon.ico HTTP/1.1
        -> url_host: favicon.ico
        -> port: 80
        -> url_path: /
        -> filter_urld: favicon.ico
failed to resolve favicon.ico: Connection timed out
        -> exiting...
-> GET favicon.ico HTTP/1.1
        -> url_host: favicon.ico
        -> port: 80
        -> url_path: /
        -> filter_urld: favicon.ico
failed to resolve favicon.ico: Connection timed out
        -> exiting...
```

```
400 : BAD REQUEST
URL FOUND IN BLACKLIST
www.facebook.com
```

Figure 3 URL Filtering

2.3.3 Content Filtering

The response is parsed and checked with an array of blacklisted words. If the response content is found to match with any word in the blacklisted words array blocked and response is not displayed. An error message is displayed on the browser as depicted in figure 4.



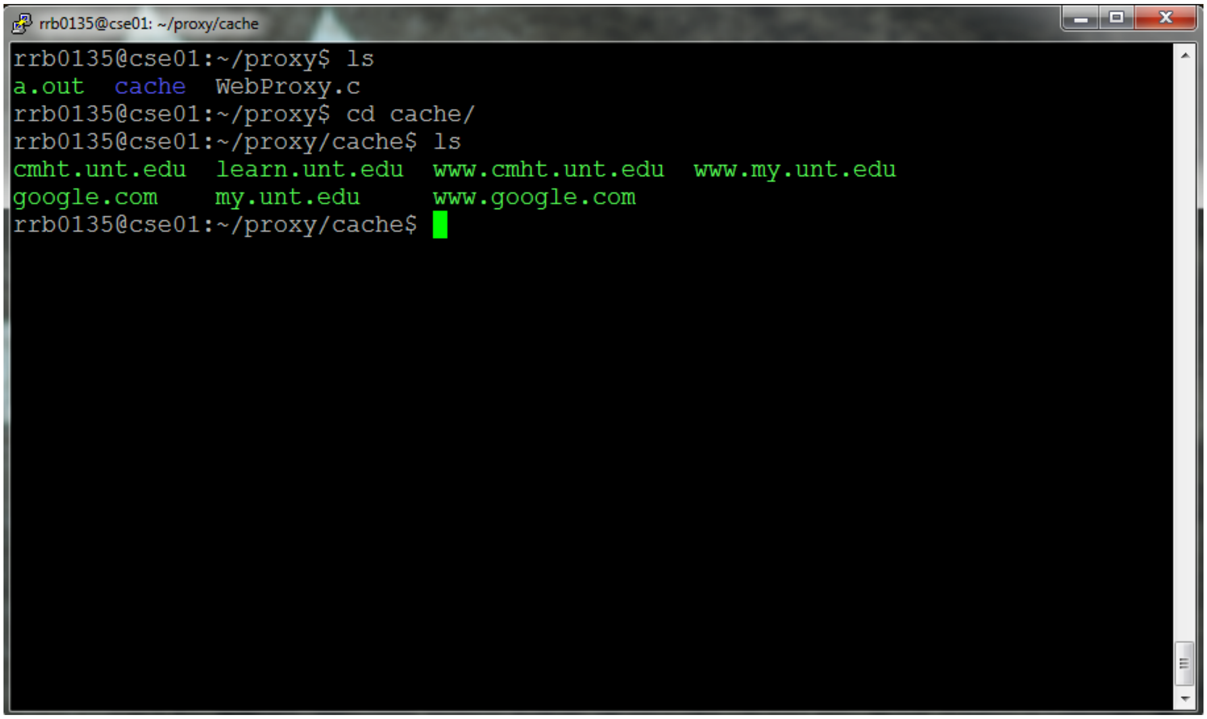
```
rrb0135@cse01:~/proxy$ gcc WebProxy.c
rrb0135@cse01:~/proxy$ ./a.out 3377
starting...
-> GET http://www.bannedwordlist.com/ HTTP/1.1
        -> url_host: www.bannedwordlist.com
        -> port: 80
        -> url_path: /
        -> filter_urld: http%3A%2F%2Fwww.bannedwordlist.com%2F
        -> connected to host: www.bannedwordlist.com w/ ip: 54.231.14
0.155
        -> first access...
        -> response_code: 200
        -> from remote host...
        -> content in blacklist: silly
        -> exiting...
-> GET favicon.ico HTTP/1.1
        -> url_host: favicon.ico
        -> port: 80
        -> url_path: /
        -> filter_urld: favicon.ico
failed to resolve favicon.ico: Connection timed out
        -> exiting...
-> GET favicon.ico HTTP/1.1
        -> url_host: favicon.ico
        -> port: 80
        -> url_path: /
        -> filter_urld: favicon.ico
failed to resolve favicon.ico: Connection timed out
        -> exiting...
```

```
400 : BAD REQUEST
CONTENT FOUND IN BLACKLIST
silly
```

Figure 4 Content Filtering

2.3.4 <u>Cache</u>

A cache of all successful request/responses are stored in a cache directory.

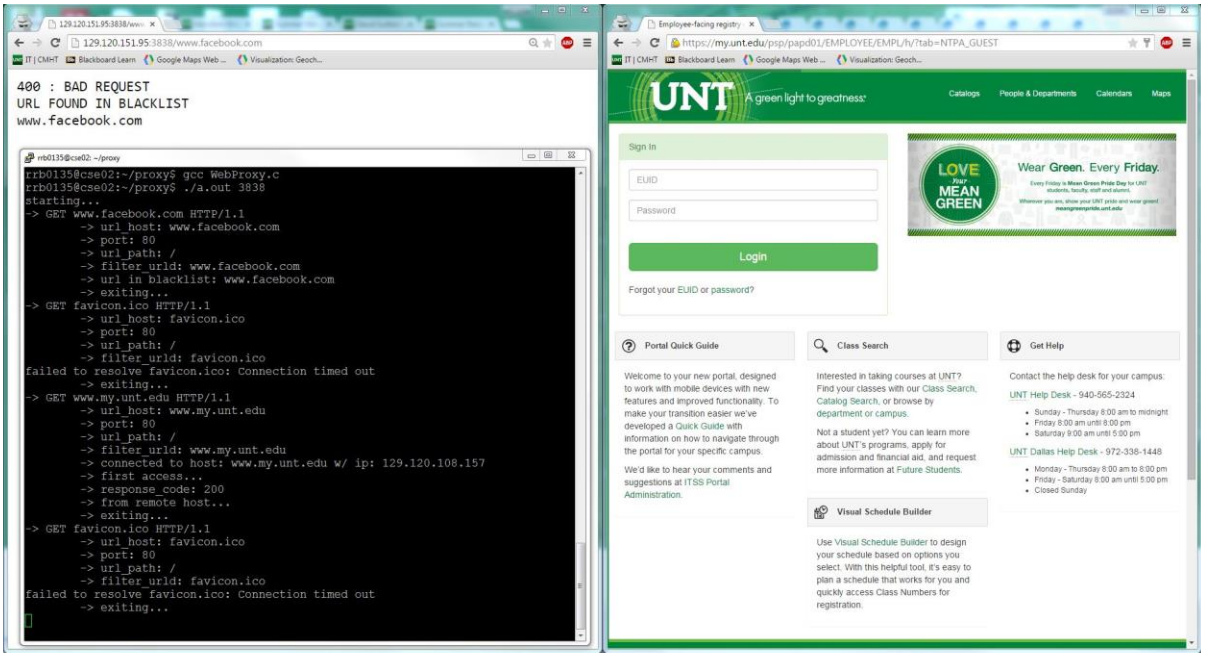


```
rrb0135@cse01:~/proxy$ ls
a.out  cache  WebProxy.c
rrb0135@cse01:~/proxy$ cd cache/
rrb0135@cse01:~/proxy/cache$ ls
cmht.unt.edu  learn.unt.edu  www.cmht.unt.edu  www.my.unt.edu
google.com    my.unt.edu     www.google.com
rrb0135@cse01:~/proxy/cache$
```

Figure 5 Cache

2.3.5 Multithreaded Web Proxy

Threads are used to handle requests from multiple clients. The server processes each request and applies all required filters on each request/response in separate threads.



```
400 : BAD REQUEST
URL FOUND IN BLACKLIST
www.facebook.com
```

```
rrb0135@cse02:~/proxy$ gcc WebProxy.c
rrb0135@cse02:~/proxy$ ./a.out 3838
starting...
-> GET www.facebook.com HTTP/1.1
        -> url_host: www.facebook.com
        -> port: 80
        -> url_path: /
        -> filter_urld: www.facebook.com
        -> url in blacklist: www.facebook.com
        -> exiting...
-> GET favicon.ico HTTP/1.1
        -> url_host: favicon.ico
        -> port: 80
        -> url_path: /
        -> filter_urld: favicon.ico
failed to resolve favicon.ico: Connection timed out
        -> exiting...
-> GET www.my.unt.edu HTTP/1.1
        -> url_host: www.my.unt.edu
        -> port: 80
        -> url_path: /
        -> filter_urld: www.my.unt.edu
        -> connected to host: www.my.unt.edu w/ ip: 129.120.108.157
        -> first access...
        -> response_code: 200
        -> from remote host...
        -> exiting...
-> GET favicon.ico HTTP/1.1
        -> url_host: favicon.ico
        -> port: 80
        -> url_path: /
        -> filter_urld: favicon.ico
failed to resolve favicon.ico: Connection timed out
        -> exiting...
```

Employee-facing registry
https://my.unt.edu/psp/papd01/EMPLOYEE/EMPL/h/?tab=NTPA_GUEST
IT | CMHT
Blackboard Learn
Google Maps Web ...
Visualization: Geoch...
UNT
A green light to greatness.
Catalogs
People & Departments
Calendars
Maps
Sign In
EUID
Password
Login
LOVE Your MEAN GREEN
Wear Green. Every Friday.
Every Friday is Mean Green Pride Day for UNT students, faculty, staff and alumni.
Wherever you are, show your UNT pride and wear green!
meangreenpride.unt.edu

Forgot your EUID or password?

Portal Quick Guide

Welcome to your new portal, designed to work with mobile devices with new features and improved functionality. To make your transition easier we've developed a Quick Guide with information on how to navigate through the portal for your specific campus.

We'd like to hear your comments and suggestions at ITSS Portal Administration.

Class Search

Interested in taking courses at UNT? Find your classes with our Class Search, Catalog Search, or browse by department or campus.

Not a student yet? You can learn more about UNT's programs, apply for admission and financial aid, and request more information at Future Students.

Visual Schedule Builder

Use Visual Schedule Builder to design your schedule based on options you select. With this helpful tool, it's easy to plan a schedule that works for you and quickly access Class Numbers for registration.

Get Help

Contact the help desk for your campus:

UNT Help Desk - 940-565-2324

- Sunday - Thursday 8:00 am to midnight
- Friday 8:00 am until 8:00 pm
- Saturday 9:00 am until 5:00 pm

UNT Dallas Help Desk - 972-338-1448

- Monday - Thursday 8:00 am to 8:00 pm
- Friday - Saturday 8:00 am until 5:00 pm
- Closed Sunday

Figure 6 Proxy Server Handling Multiple Requests

3 References

3.1 Books

[1] Computer Networks, 5th edition, Peterson, Morgan Kaufman

[2] UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI.

[3] Let Us C 5th Edition by Yashwant Kanetkar

3.2 Websites

[4] http://www.linuxhowtos.org/C_C++/socket.htm

[5] http://beej.us/guide/bgnet/

[6] https://help.github.com/articles/set-up-git/

3.3 GitHub Repository URL

[7] https://github.com/Ramchandra3/Multithreaded-WebProxy.git