Read Me

How to use:

1. Set Firefox proxy

Go to ‘Edit’’Preference’’Advanced’’Setting’

Select ‘Manual Proxy Configuration’, enter the host name as ‘localhost’, and use the same port number you will specify in the command line. Click the little box bellow.

1. Complie command:

Gcc webproxy.c –o webproxy –lssl –lcrypto -lpthread

Run the code with following command:

./webproxy port

or ./webproxy timeout\_value port

(\*Note\*: If a timeout value is specified, the cached file will no longer be accessed after the timeout that you defined in the command line. Otherwise, it will use the default timeout value of 60s.)

1. You can either test the proxy in the browser or using telnet.
2. If you test the proxy in the browser, just enter the http website you want to visit.

\*Note\*: My proxy works well with the following webpages, but not guarantee other websites because some are too complicated with many contents so I didn’t try them.

[www.bing.com/](http://www.bing.com/)

[www.baidu.com/](http://www.baidu.com/)

1. If you use telnet, open another terminal, type:

telnet localhost portnumber

After something comes out automatically, type:

GET <http://www.bing.com/> HTTP/1.0 (I’m just using bing for example.)

Then you will see the content of the webpage is shown in the terminal. Simultaneously, the webpage will be cached.

Functions and design decisions:

Multi-thread

I use pthread to achieve multi-thread. To test multi-thread:

1. If using Firefox, for example, type:

[www.bing.com/](http://www.bing.com/)

You can open another tag, type another webpage:

[www.baidu.com/](http://www.baidu.com/)

You will see the webpages are shown normally in the browser.

1. If using telnet, just open multiple terminals, and using the same command mentioned before.

Caching

I use a link list to store the md5 hash value of the host+url, as well as the current time (assuming named t1) a node is added to the link list. Each time when I’m checking if a file is cached or not, it will get the current time again (assuming named t2), computing t2-t1 and check if it is greater than or smaller than the timeout value. If greater, it means timeout occurs, it will just ignore the node and create a new node at the beginning of the link list and get everything from the socket. If timeout not occurred, it will read everything from the local file and return the content to the client. To test caching, after you done everything above, disconnect the internet, open a new tag and enter the webpage you visited before, it will be shown in the new tag. Remember to do this before timeout occurred. You can also see the files in proxy\_cached folder.

Link Prefetching

\*Very Important Note\*:

My link prefetching only works well when you run with gdb. So, I comment my link prefetching at line <312>(The statement is:

read\_file(host, url, ( void \*) (long) s\_server, serv\_port);

). It is because if you want to use link prefetching run the code normally, you can open the first webpage in Firefox, and you can see the content of every link is cached in your directory. But when you try to open the second webpage, it keeps running without the content shown in the browser, I think it is because this is a tiny proxy but most webpages are too complicated which contains too many links and this will cause the number of threads exceeds the max number of threads allowed. I comment that line because link-prefetching will affect multi-thread in some sense, but my link prefetching actually works in the following circumstance:

To test how link-prefetching works, cancel the comment of that specific line, \*must run the code using gdb\*, the comment is:

gdb webproxy

After something comes out, type:

Run portnumber

And then open the browser, enter the webpage, for example:

[www.bing.com/](http://www.bing.com/)

And the you can try to open another tag, and enter:

[www.baidu.com/](http://www.baidu.com/)

And you can see the content of each link is cached in the folder. If you click the link in the browser after disconnecting the internet, it may not show the content because I think it is the problem of the website, when I check the file it cached in the directory, it is 203 found error and this is why nothing shows in the browser. But it can open the link sometimes. My proxy do achieve link prefetch because the content of each link is stored under the directory. (You can check this with a simple webpage, example.com, there is only one link in the page, and you will see more than one page cached besides the main page).

When I visit the main page, as soon as I got the cached file, I read the content from that file, parsing the links to extract all the links. Then I parse the link to get host and url, I create another thread to create a socket using the host and url and port number to connect to the actual server.

Error handling:

The proxy can handle error: 400 Bad Request, 500 Internal Server Error, 501 Not Implemented and 502 Bad Gateway.

Parse the request:

This is the request client send to the proxy:

GET <http://www.bing.com/>HTTP/1.0

After parsing and reconstruct, this is the request the proxy send to the server:

GET /url HTTP/1.0\r\n

Host: www.bing.com\r\n\r\n