

COMPSCI 356: Computer Network Architecture

Spring 2015

[Home](#)[Syllabus](#)[Assignments](#)[Exams](#)[Piazza](#)

COMPSCI 356 Computer Network Architectures: Proxy

Overview

In this assignment, you will implement a Web proxy with Caching that lies between your browser and the Internet. You must write your own code, and may not copy code from any other source.

All HTTP requests sent from your browser to any Web site's origin server should go through your proxy, which will then forward any responses from origin servers back to the browser. If you implement your proxy correctly, then after changing your browser settings to indicate that you would like to use your proxy as a Web proxy, you should be able to browse the Web and view Web sites just as you could when you were not using your proxy. Your program must be written in C and must use the pthreads library to create a separate thread to process each request that the proxy receives.

The basic work flow of your proxy should be as follows:

1. From the command line, start your proxy so that it listens on a certain port given as a command-line argument. Your proxy accepts two arguments: the first the port, the second the size of the cache in MBs.
2. In your browser, when you type a URL such as <http://www.cnn.com>, the request should be forwarded to your proxy. This can be done by going into your browser settings and indicating that you want to use a Web proxy. You need to specify the host and the port of the proxy. Then your browser will then send all HTTP and HTTPS requests to your proxy.
3. All the connections that your proxy creates will be TCP connections.
4. Upon receiving a request for a new connection, the proxy should spawn a thread to process it. By reading from the connection, the thread learns whether the request is GET for HTTP. You only need to worry about GET requests.
5. In the case of GET, the request includes a URL that contains both a host/port and a path. The proxy checks to see if the object is in its cache,
 - a) if so the content is served from cache. Note, you will need to generate your own HTTP headers.
 - b) Else, if not in the cache the proxy uses the URL to fetch the content from the origin web server, and as proxy receives the content, the proxy passes it on to the client and stores the content in its cache. Before storing content in your Cache, you may need to remove HTTP headers.
6. Implement a Least Recently Used Cache eviction policy. (You learned this in OS class)

7. When there is no more data to be transferred, close the connection to the web server and the connection to the browser.
8. Note that you are required to support persistent connections. A persistent connection between the proxy and a Web server should be kept open only as long as a persistent connection is kept open between a client and the proxy.
9. You will not be required to support HTTP PUT and POST methods.
10. when the client closes the connection, the proxy should also close the connection.

Code base

You will start from scratch. We do not provide you with any files. You can use your assignment 1 as a starting point.

Some implementation detail

Here are some hints that will help you get your cache-proxy work correctly.

1. Many things can go wrong when establishing a network connection or in the midst of reading and writing across a network connection. Your proxy should check for errors and handle them gracefully. In particular, your proxy should not crash or exit just because something goes wrong with a network connection. Be careful about which library functions you use to ensure that you handle errors gracefully.
2. Make sure that your proxy ignores SIGPIPE signals by installing the ignore function as handler. This can be done by using `signal(SIGPIPE, SIG_IGN)`.
3. Sockets are created using IP-addresses. However, the URL will be in URL format. You will need to convert from URL to IP-Addresses. To do this, you will need to use the `getaddrinfo()`, please see consult Beej's Guide: <http://beej.us/guide/bgnet/output/html/multipage/getaddrinfo.man.html>
4. Primer for HTTP format can be found in slides: L20. The exact information can be found here: <http://www.w3.org/Protocols/rfc2616/rfc2616.html>

Submitting

Submit your source code with a readme file on Sakai.

Collaboration policy

For this project, you can form a team of two students to complete the assignment.

In particular, you may not copy code from Web sites, from books, from other students who are taking the class now or who have taken the class in the past, or from any other person.