

시스템프로그래밍

Proxy #2-4

담당 교수 : 최상호 교수님(목4)

2021202003 강준우

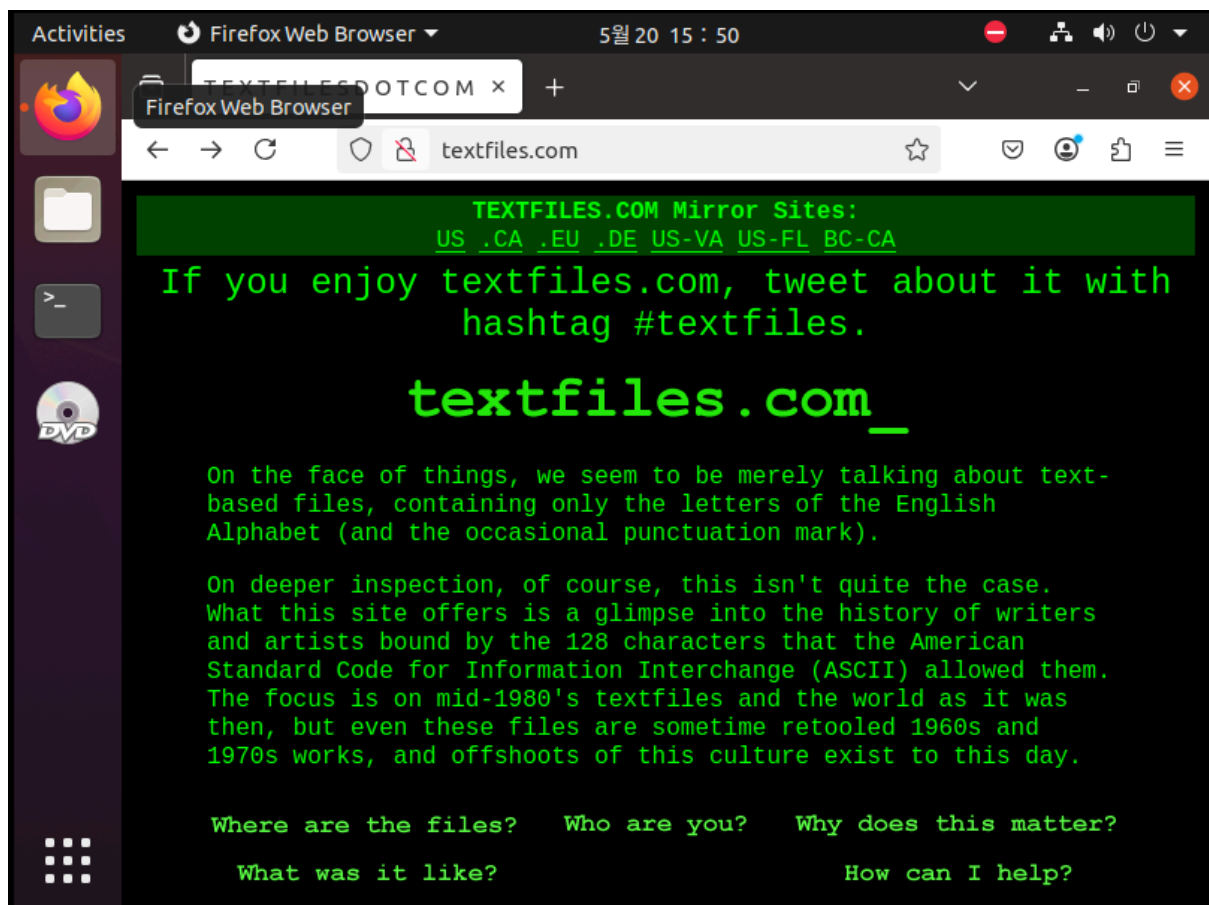
Proxy #2-4

Introduction

이번 과제에서는 기존의 **Proxy#2-3**를 바탕으로 역할에 따른 서버 응답에 더 중점을 두고 구현합니다. 기존의 코드에서 추가적으로 로그파일은 하나의 url에 하나의 줄만 기록하게 설계합니다. 또한 **Miss Hit**에 따른 구현에서, 모든 리스폰스를 서버에서 읽어서 캐시에 해당 내용을 기록하도록 합니다. 반대로 **hit**일때는 페이지 네트워크에 리스폰스에 대한 **write**를 수행합니다.

결과화면

<http://textfiles.com/> 을 통해서 수행했습니다.
프로그램 실행 후 해당 사이트를 접속한 모습입니다.



과제의 캐시 파일 리스폰스 **write**를 수행하는 모습입니다.

Firefox Web Browser

TEXTFILES.COM x TEXTFILES.COM x + -

textfiles.com

TEXTFILES.COM Mirror Sites:
[US](#) [.CA](#) [.EU](#) [.DE](#) [US-VA](#) [US-FL](#) [BC-CA](#)

Inspector Console Debugger Network Style Editor Performance

Filter URLs

Disable Cache No Throttling

All HTML CSS JS XHR Fonts Images Media WS Other

Headers Cookies Request Response Timings

Filter Headers

Block Resend

GET http://textfiles.com/

Status 200 OK

Version HTTP/1.1

Transferred 10.86 kB (10.58 kB size)

Request Priority Highest

DNS Resolution System

Response Headers (275 B)

Raw

Accept-Ranges: bytes

Connection: close

Content-Length: 10584

Content-Type: text/html

Date: Tue, 20 May 2025 07:14:40 GMT

Etag: "2958-625f949a948e2"

Last-Modified: Sun, 03 Nov 2024 02:54:33 GMT

8 requests 10.58 kB / 10

Hit와 함께 요구사항에 맞게 모든 내용을 저장해서 해쉬화된 파일에 저장되는 모습입니다.
 HTTP상태와 함께 Date/ Server/ Last modified/ Etag/Accept-rages/ Content length/
 connectitno/ content type이 보이고
 이후에 boty부분이 보입니다.

```

1  HTTP/1.1 200 OK
2  Date: Tue, 20 May 2025 07:14:40 GMT
3  Server: Apache/2.4.58 (FreeBSD) OpenSSL/1.1.1o-freebsd
4  Last-Modified: Sun, 03 Nov 2024 02:54:33 GMT
5  ETag: "2958-625f949a948e2"
6  Accept-Ranges: bytes
7  Content-Length: 10584
8  Connection: close
9  Content-Type: text/html
10
11  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
12  <HTML><TITLE>T E X T F I L E S D O T C O M</TITLE>
13
14  <!-- TEXTFILES.COM is a site dedicated to the Textfiles of the BBS World -->
15  <!-- from the 1980's. Naturally, this role expands into the continuation -->
16  <!-- of the textfile-writing world into the 1990's, and goes back to the -->
17  <!-- 1960's and 1970's as well. If you have any questions, feel free to -->
18  <!-- mail them to jason@textfiles.com. Submissions are always wanted. -->
19
20  <!-- Unlike a lot of sites, this Page Source has a lot of comments. This -->
21  <!-- is because this site's main purpose is education, and a user should -->
22  <!-- have the opportunity to see how this site was created and what the -->
23  <!-- thought process was behind the look. Plus, the site weighs in at a -->
24  <!-- very low size, so giving an explanation with the download is not a -->
25  <!-- very bandwidth-intensive activity, and might inspire someone. -->
26
27  <!-- Since TEXTFILES.COM harkens back to an earlier era, but is being done -->
28  <!-- in the modern era, it is the usual blend of recent features and hacks -->
29  <!-- to simulate older technology. In this case, everything is in green -->
30  <!-- (which, believe it or not, is the light that the eye takes the longest -->
31  <!-- to get tired looking at) on a black screen, and the visual "hook" is -->
32  <!-- the Blinking Cursor, which has disappeared in a lot of mainstream, -->
33  <!-- windowed applications. This is accomplished with small animated GIFs. -->
34
35  <!-- Now, let's make the page the usual black background and green text. -->

```

이후 결과에서는 해당 경로에 접속 후 내용들을 로그로 기록한 모습이며, 이후 재접속 했을 때 hit를

반환하는 모습도 기록되었습니다.

```

[MISS]http://textfiles.com/-[2025/05/20, 16:14:40]
[HIT]565/73643331b10dd1a32919b722a5982e85e6662-[2025/05/20, 16:14:57]
[HIT]http://textfiles.com/

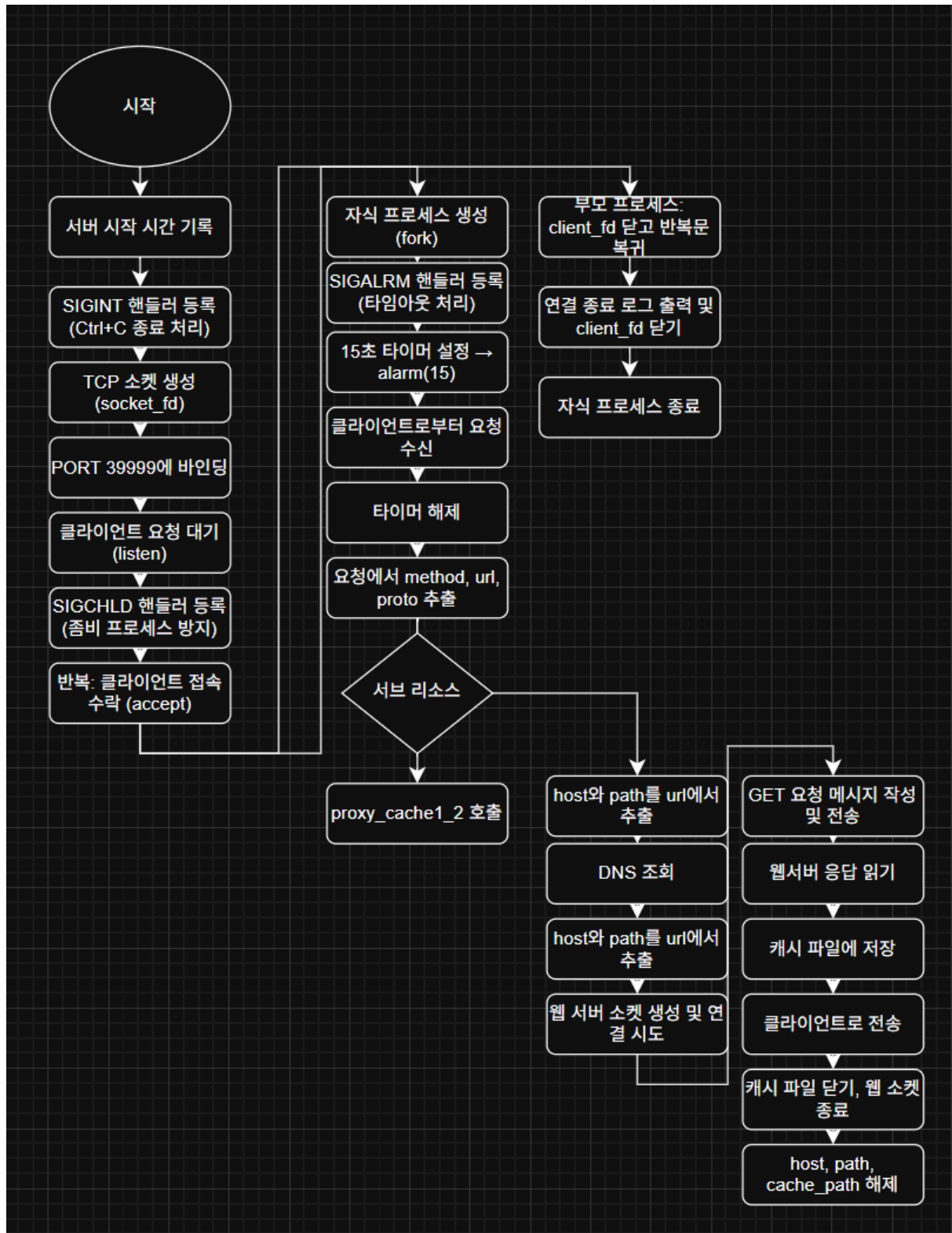
```

▼ cache/565

🔍 73643331b10dd1a32919b722a5982e8...

파일 생성도 이전과 같이 한번만 되는 모습입니다.

Algorithm – Flow Chart



pseudo code

서버 시작 시간 기록

SIGINT 핸들러 등록 (서버 종료 시 로그 작성)

SIGCHLD 핸들러 등록 (좀비 프로세스 방지)

socket()으로 TCP 소켓 생성
bind()로 IP, PORT 설정
listen()으로 클라이언트 요청 대기 시작

while (true):

client 연결 수락 (accept)

fork()로 자식 프로세스 생성

if 자식 프로세스:

SIGALRM 핸들러 등록 (타임아웃 15초 설정)

read()로 클라이언트 요청 수신

if 요청 실패:

연결 종료 및 종료

요청에서 method, url, proto 파싱

if 메인 URL이 아닌 경우:

연결 종료 및 종료

캐시 경로 및 HIT 여부 확인 (proxy_cache1_2)

if HIT:

캐시 파일 읽어서 클라이언트에 전송

else:

host와 path 파싱

gethostbyname()으로 IP 조회

웹 서버로 TCP 연결 시도

연결 실패 시 종료

HTTP GET 요청 구성 후 웹 서버에 전송

웹 서버 응답을 캐시에 저장하며 클라이언트에 전송

메모리 정리 및 소켓 종료

로그 출력 (클라이언트 연결 해제)

프로세스 종료

else:

부모는 client 소켓 닫고 다음 요청 대기

고찰

이번 수행을 위해서 이전에 구현한 함수들에 많은 수정이 필요했습니다. 특히 1-2함수에서 히트 미스에 따른 구현이 다시 추가되었기 때문에 캐시파일에 대한 경로 뿐만 아니라 히트 미스를 위한 논리형 연산자도 돌아왔습니다. 뿐만아니라 요구사항에 따라, 로그 파일을 기록할 때 gif ico 등의 로그가 남지 않게 하려고 함수를 통해 제한하는 과정이 추가되었습니다.

Reference

2025-1_SPLab_proxy_Assginment2-4