

시스템프로그래밍

Proxy #2-2

담당 교수 : 최상호 교수님(목4)

2021202003 강준우

Proxy #2-2

Introduction

이번 과제에서는 기존의 Proxy#2-1를 바탕으로 직접 **firefox**를 통해서 소통을 시도해봅니다. 이전 과제에서는 클라이언트와 서버의 파일을 분리하고 수행했던 내용을 **firefox**를 사용하는 클라이언트와 해당 내용을 기반으로 소통하는 서버를 구현 및 연결하여 해당 사이트에 대한 접속 기록 기반 **hit miss**를 판별합니다.

결과화면

각각에서는 같은 경로의 주소를 처음에는 **MISS**를 반환하지만 이후 **HIT**로 바뀌어 반환하는 모습입니다.

MISS

192.168.161.141:53346

<http://info.kw.ac.kr/>

kw2021202003

```
[192.168.161.141 : 53346] client was connected
=====
Request from [192.168.161.141 : 53346]
GET http://info.kw.ac.kr/ HTTP/1.1
Host: info.kw.ac.kr
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:136.0) Gecko/20100101
Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i

=====
[192.168.161.141 : 53346] client was disconnected
```

```
kw2021202003@ubuntu:~/cache$ tree
└── 4d2
    └── 59e95a0f7aae0ff67b0626fb9818a9025a42b

1 directory, 1 file
```

해당 파일 디렉토리로 MISS한번을 통해 파일이 생긴것을 확인할 수 있습니다.

The terminal window shows a log entry from a web server. The log entry indicates a client connection from IP 192.168.161.141 port 45702. The client sent an HTTP GET request for the root URL. The browser screenshot shows a simple page with the word "HIT". Below the browser, the server log continues with the client disconnecting.

```
[192.168.161.141 : 45702] client was connected
=====
Request from [192.168.161.141 : 45702]
GET http://info.kw.ac.kr/ HTTP/1.1
Host: info.kw.ac.kr
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:136.0) Gecko/20100101
Firefox/136.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i

=====
[192.168.161.141 : 45702] client was disconnected
```

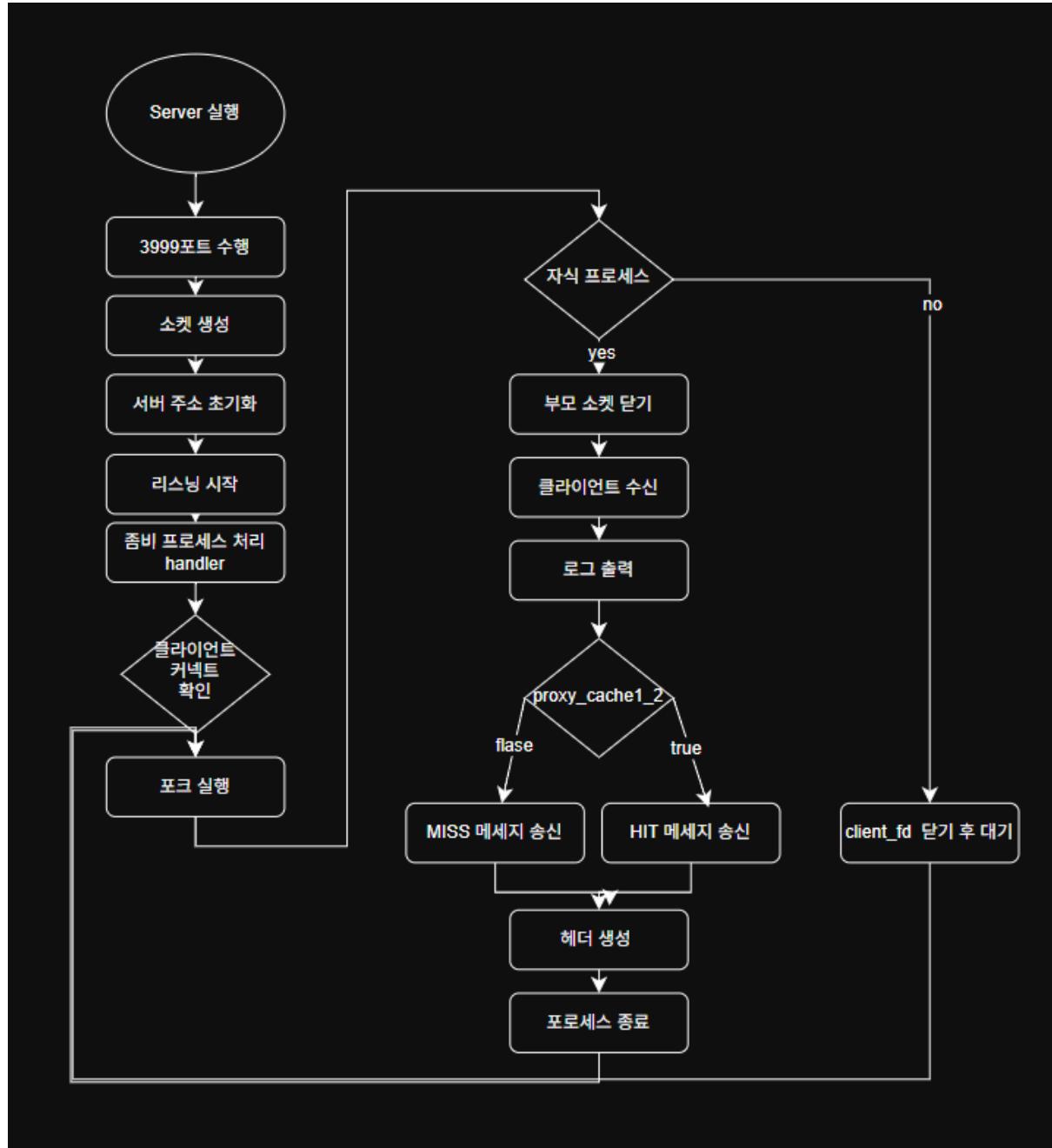
이후 디렉토리를 통해서 HIT를 확인하고 결과를 반환한 모습입니다.

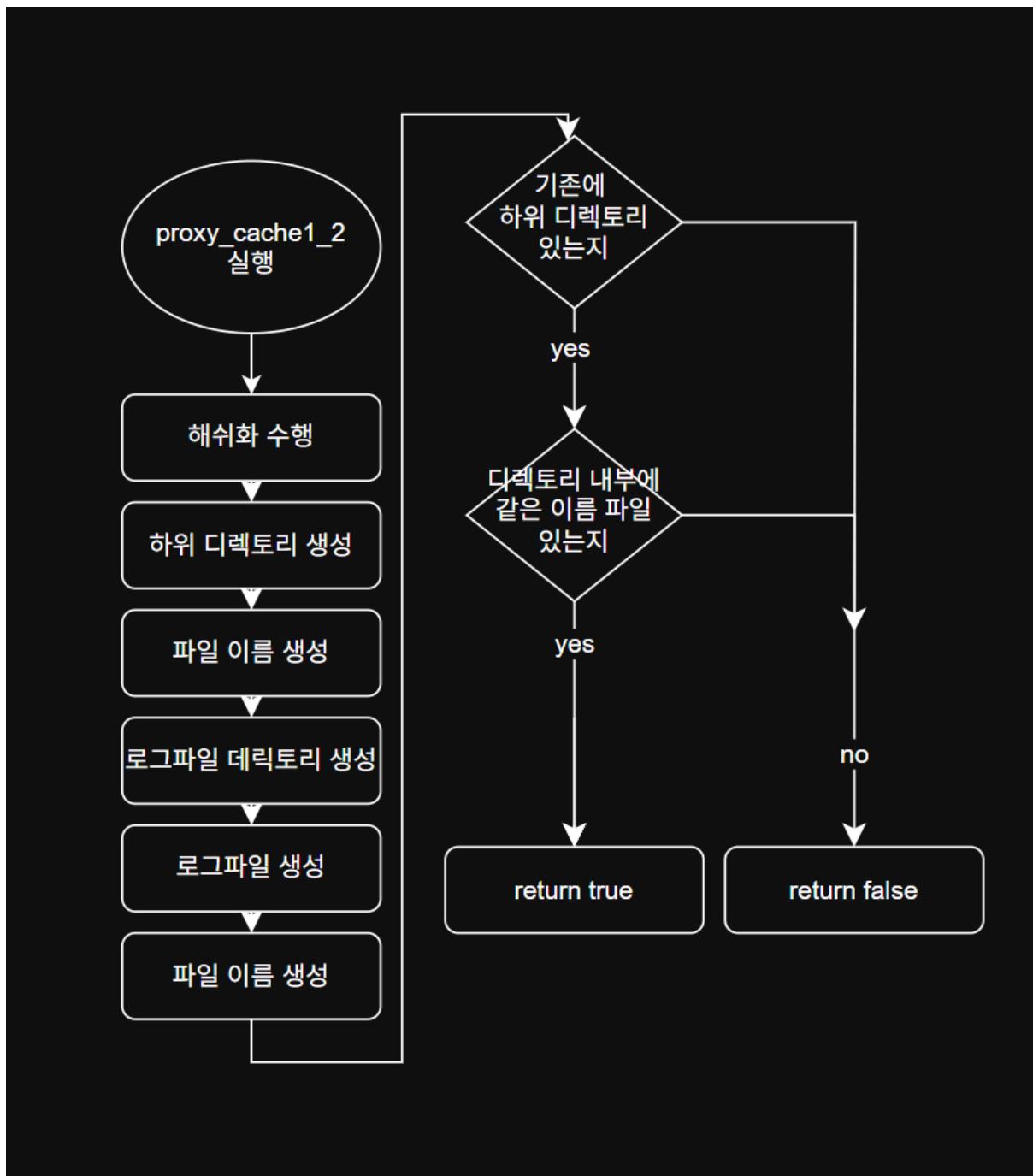
A screenshot of a Linux desktop environment. At the top, there is a dock with icons for 'Activities' (purple), 'Text Editor' (blue), 'Dash' (orange), and 'Ubuntu Software' (dark blue). A terminal window titled 'Text Editor' is open, showing the contents of a file named 'logfile.txt' located at '~/.logfile'. The terminal displays the following log entries:

```
1 [Miss]http://info.kw.ac.kr/-[2025/05/08, 21:34:16]
2 [Hit]4d2/59e95a0f7aae0ff67b0626fb9818a9025a42b-[2025/05/08, 21:34:21]
3 [Hit]http://info.kw.ac.kr/
4
```

로그파일도 해당텍스트 처럼 보여줍니다. 초기 세팅 로그를 생략하고 핵심 내용만 발췌한 모습입니다.

Algorithm – Flow Chart





Pseudo Code

서버 소켓을 생성

서버 주소 구조체를 초기화하고 IP(PORT)를 설정

`bind()`를 통해 서버 소켓과 주소를 연결

`listen()` 호출로 클라이언트 연결을 대기

SIGCHLD 시그널 핸들러를 등록하여 좀비 프로세스를 방지

루프

클라이언트 연결을 accept()

fork()를 통해 자식 프로세스를 생성한

자식 프로세스 (fork() == 0)인 경우

부모 소켓을 닫기기

클라이언트 IP와 포트를 출력

클라이언트 요청 메시지를 읽기

요청 로그를 출력

HTTP 요청에서 method, url, proto를 파싱

proxy_cache1_2(url)을 호출하여 캐시 HIT 여부를 판단
HIT인 경우:

<h1>HIT</h1>
IP:PORT
URL
학번
 형식으로 본문 구성

MISS인 경우:

<h1>MISS</h1>
IP:PORT
URL
학번
 형식으로 본문 구성

HTTP 응답 헤더 생성

응답 헤더와 본문을 클라이언트에게 전송

연결 종료 로그 출력 후 클라이언트 소켓을 닫고 종료

부모 프로세스인 경우

클라이언트 소켓을 닫고 다음 연결 대기

main 종료 시 소켓 닫고 종료

고찰

이번 과제에서는 좀더 실전같이 파이어폭스 사이트와 연결을 통해서 이전 과제를 구현해보았습니다. 하면서 점점 기존에 코드의 사용이 줄어든다는 생각이 들기도 했습니다 1-2과제의 코드등의 재사용율이 줄기도 했으며, 또한 클라이언트 부분은 이제 파이어폭스에서 수행하다보니 사용이 줄어든 모습도 확인할 수 있었습니다. 결과적으로

이렇게 개발이 넘어가다보면 프론트엔드 부분과 백엔드 분야의 구현으로 분할되겠다는 생각을 하였습니다.

Reference

2025-1_SPLab_proxy_Assginment2-2