System Programming

Assignment#2-2

06_Proxy 2-2

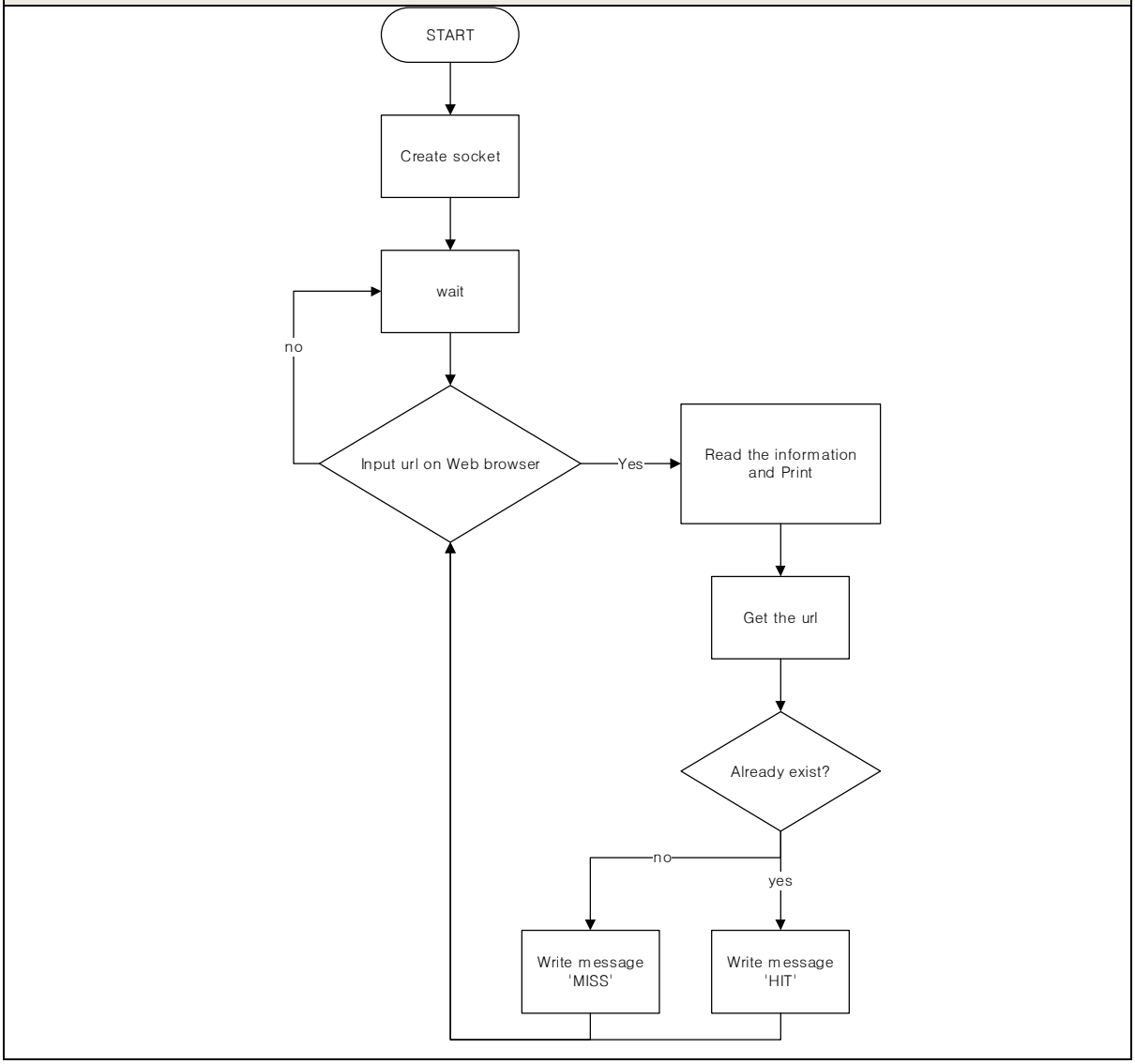
Professor	목 3 4 황호영 교수님
Department	컴퓨터공학과
Student ID	2012722028
Name	장 한 별
Date	2018. 05. 04

Introduction.

시스템 프로그래밍 강의 시간에 배운 proxy server 를 구현하는 것을 목표로 한다.
이번 과제는 Clinet 의 browser 의 모든 HTTP resquest 를 proxy server 에 전송하여 전송된 data 중 host 정보를 추출하고 그 정보로 HIT 인지 MISS 인지 판별해서 Web browser 에 출력하는 것이다. 이때 sprint 함수로 출력할 message를 저장하고, write 함수를 이용해 출력하도록 한다.
반드시 header 정보를 먼저 write 한 후, message 를 write 하도록 한다.

Flow chart.

proxy_cache



위 그림은 이번 과제 proxy_cache 의 flow chart 이다. Server 측에서 socket 을 생성 후, web browser 에서 url 을 입력할 때까지 대기한다. Web browser 에서 url 을 입력시 해당 정보들을 HTTP 형식으로 terminal 에 출력한다. url 로 hashing 한 후, hashed url 로 directory 와 file 들을 생성하는 부분은 생략하겠다. MISS 상태이면 다시 web browser 에서 MISS 를 출력하도록 message 를 write 하고, HIT 일 때도 마찬가지로 message 를 write 한다.

Pseudo code.

proxy_cache

```
Int main(void)
{
    Socket();
    Setsockpot();
    Bind(socket);
    Listen(socket, 5);
    Signal(signal handler);

    While(1)
    {
        Accept();
        Read(client);

        Printf("%s", url information);
        Get url;

        Assignment #1-2;
        If(MISS)
        {
            Message = "MISS";
            Write(client, header);
            Write(client, message);
        }
        Else
        {
            Message = "HIT";
            Write(client, header);
            Write(client, message);
        }
        Close(client);
    }
    Close(socket);
    Return 0;
}
```

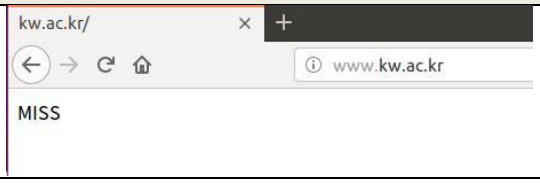
위 그림은 proxy_cache 의 pseudo code 이다. Server 측에서 socket 을 생성 후, web browser 에서 url 을 입력할 때까지 대기한다. Web browser 에서 url 을 입력시 해당 정보들을 HTTP 형식으로 terminal 에 출력한다. url 로 hashing 한 후, hashed url 로 directory 와 file 들을 생성하는 부분은 생략하겠다. MISS 상태이면 다시 web browser 에서 MISS 를 출력하도록 message 를 write 하고, HIT 일때도 마찬가지로 message 를 write 한다. Message를 write 할 때, 반드시 header 의 정보를 먼저 write 를 한 후 message 를 write 하도록 한다.

Result.

2-2.(1)

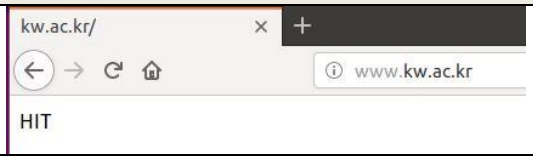
위 그림은 proxy_cache 를 실행 시킨 실행화면이다. Proxy server 인 proxy_cache 를 실행하지 않고 web browser를 사용할 시, Error가 발생한다. 반드시 server를 먼저 실행한 후 작업하도록 해야한다.

2-2.(2)



위 그림은 web browser 에서 www.kw.ac.kr 을 입력했을 때의 화면이다. Proxy_cache를 실행 후 www.kw.ac.kr 이라는 url 을 처음으로 입력했으므로 'MISS' 가 write 된다. Message 가 올바르게 write 되었다는 것을 확인할 수 있다.

2-2.(3)



위 그림은 web browser 에서 www.kw.ac.kr 을 다시 입력했을 때의 화면이다. 2-2.(2) 에서 www.kw.ac.kr. 을 이미 입력 했으므로 해당 url 이 hashing 되어서 그 hashed url 로 directory 와 file들이 생성되는데 2-2.(2) 에서 생성되었으므로 HIT 로 판별된다. 그 후, 'HIT' 가 write 된다. Message 가 올바르게 write 되었다는 것을 확인할 수 있다.

2-2.(4)

```
[127.0.0.1 : 60121] client was connected.
============================================================
Request from [127.0.0.1 : 60121]
GET http://www.kw.ac.kr/ HTTP/1.1
Host: www.kw.ac.kr
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:59.0) Gecko/20100101 Fire
fox/59.0
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
Cookie: WMONID=Djmk-nnZqWA
Connection: keep-alive
Upgrade-Insecure-Requests: 1


============================================================
```

위 그림은 2-2.(1) ~ 2-2.(3)의 과정 후 terminal 에 출력된 결과 화면이다. www.kw.ac.kr 정보가 HTTP 형식에 맞게 출력되었음을 확인할 수 있다.

Conclusion.

이번 과제는 client 인 web browser 와 proxy server 를 connection 후 서로 통신하는 작업을 구현하는 것이 목표였다. Proxy_cache 를 실행하면 proxy server 는 web browser 가 connection request를 할 때 까지 기다리는데 이때, url 을 입력하면 connection 이 된다. 그 후 입력된 url 로 directory 와 file들을 생성하고 HIT 인지 MISS 인지 판별해 그 결과를 web browser로 출력할 수 있도록 message에 write 하도록 했다.

이번에 구현해야 할 것들은 비교적 쉬운 내용이었다. HTTP Request 의 기본적인 format 과 HTTP Response 의 format. 그리고 Assignment #2-1 에서 사용했었던 read(), write() 역시 큰 문제가 없었다. 하지만 Assignment #2-1 에서의 내용을 수정하는 부분에서 지워야 할 것 들을 지우지 않으면 이번 과제에서 문제가 생겼었는데, 그 원인을 찾는 것에 많은 시간이 걸렸다. 지난 과제는 client.c 파일을 따로 만들어 그것과 server와 connection 후 read(), write() 로 통신을 한 반면, 이번에는 web browser 가 client 여서 굳이 반복적으로 read() 를 안해줘도 되는 부분에서 read() 를 해 client port number 를 계속해서 다른 번호로 받아와 web browser 에 message가 write() 되지 않던 문제였다. 구현한 코드를 세세하게 살펴보고 이번에 구현해야 할 점에 대해 이처럼 문제가 되는 부분들을 잘 파악한 후 수정을 해서 별 문제없이 완성할 수 있었다.