시스템 프로그래밍 실습 Report

실습 제목: Proxy #1-2

실습일자: 2022년 3월 28일 (월)

제출일자: 2022년 4월 06일 (수)

학 과: 컴퓨터정보공학부

담당교수: 최상호 교수님

실습분반: 목요일 7,8교시

학 번: 2018202065

성 명: 박 철 준

- Introduction

1. 제목

Proxy 1-2

2. 목적

가. 시스템으로부터 현재시간을 구해서 프로그램에 사용하여야 한다.

나. Log파일을 생성하고 Hit과 Miss를 판별하여 Log파일에 현재시간과 함께 저장 할 수 있어야 한다.

3. 배경지식

가. time_t : time()함수를 통해 초단위의 데이터를 받아오기 위한 자료형

나. ctime함수: 초단위 시간 정보를 문자열로 변환

다. gmtime함수: 초 단위 시간 정보를 struct tm 형으로 변환 UTC를 기준으로 시간 할당

라. localtime함수: 초 단위 시간 정보를 struct tm 형으로 변환 시스템 로컬 시간을 기준으로 시간을 할당

마. asctime함수: struct tm 시간 정보를 문자열로 변환

바. mkttime함수: tm 시간 정보를 초 단위시간 정보로 변환

사. time함수: UTC부터 흐른 시간을 반환

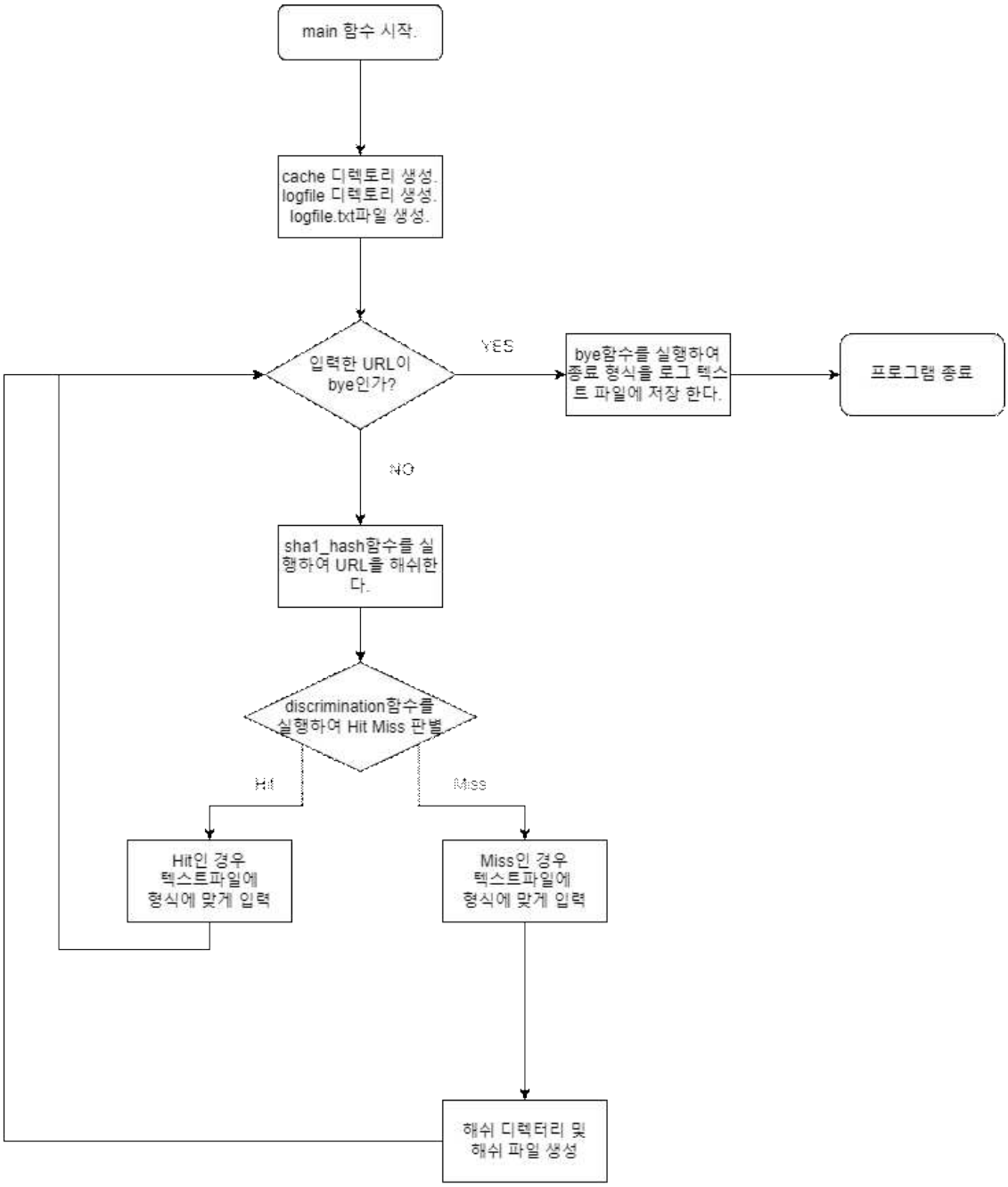
아. opendir함수: 특정경로에 특정하위 디렉토리를 여는 함수

자. readdir함수: 특정경로에 특정하위 디렉토리 혹은 파일이 있는지 확인하는 함수

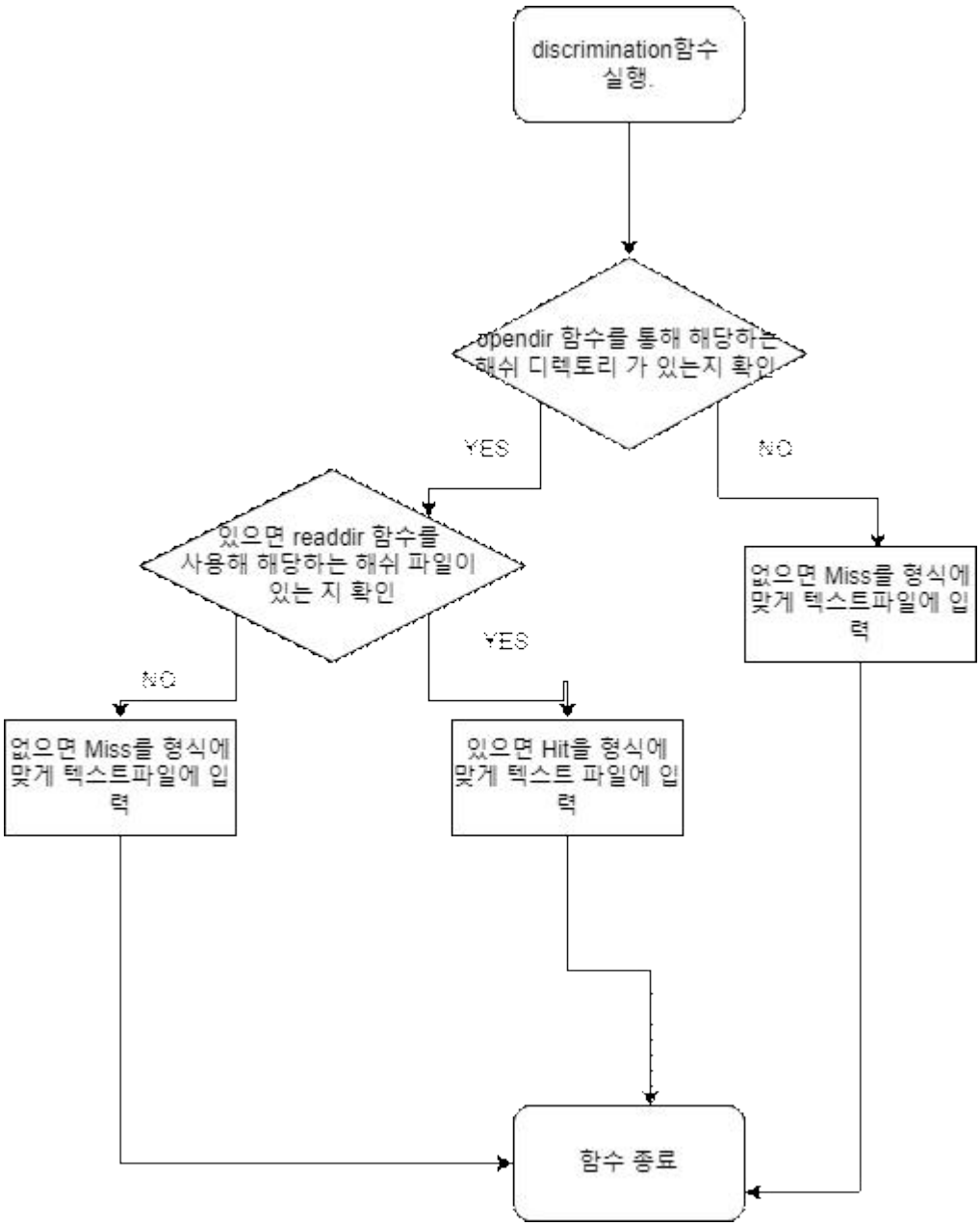
● 실습 결과

1 proxy #1-2

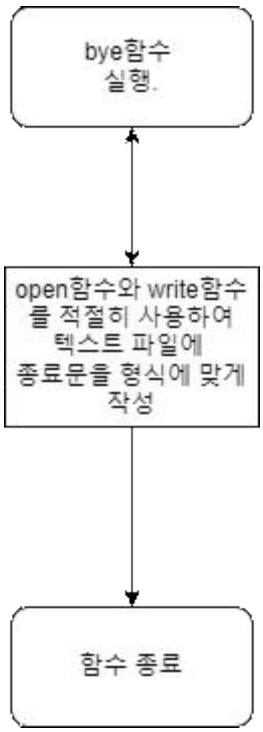
-Flow Chart
main 함수
1-1에서 작성한 main함수의 플로우 차트를 기반으로 1-2의 main의 플로우차트를 만들었고 1-1에서의 중점을 두었던 과정은 축소하여 표현하였고 1-2의 중점사항을 부각시켜 표현하였다. 이는 아래와 같다.



discrimination함수



bye함수


bye함수
실행.
open함수와 write함수
를 적절히 사용하여
텍스트 파일에
종료문을 형식에 맞게
작성
함수 종료

-Pseudo code

main함수

```
int main()
{
        문자열 변수 URL, HS_URL 선언
        해쉬된 문자열을 저장하기 위한 first_HS_URL, second_HS_URL 변수 선언
        디렉토리 위치를 저장할 문자열 변수 dire선언
        miss횟수와 hit회수를 기록할 인트형 변수 misscount, hitcount선언
        프로그램 시작 시간과 종료 시간을 저장할 time_t타입 변수 start와 end
        선언
        getHomeDir함수를 실행해서 dire에 홈디렉터리 위치 저장
        strcat 함수로 /cahe/를 dire에 붙이기
        umask 함수로 초기 설정된 umask값 변경
        mkdir함수로 홈디렉터리위치에 cache폴더 생성
        mkdir함수로 홈디렉터리위치에 logfile디렉토리 생성
        open함수로 logfile.txt파일 생성
        while(1){
                변수 URL에 사용자의 URL입력받아 저장하기
                if(strcmp함수로 변수 URL이 "bye"인지 확인)
                {
                        result = (int)(end - start);//시작 시간과 종료시간 빼서 인트형 변수
                        에 넣어준다
                        맞으면 result와 hitcount, misscount 를 인자로 받는 bye함수실행
                        후
                        프로그램 종료
                ]
                sha1_hash함수를 실행하여 HS_URL변수에 해쉬된 URL저장
                해쉬된 URL에서 앞 3글자를 가져와 first_HS_URL에 저장
                해쉬된 URL에서 앞 3글자를 제외하고 모두 가져와 second_HS_URL에 저장
                if(discrimination함수를 실행하고 반환값이 1이면)
                        misscount 1증가
                else
                        hitcount 1증가
                getHomeDir함수를 통해 dire변수에 홈디렉토리 위치 저장
                strcat으로 dire에 /cache/와 first_HS_URL변수를 붙여줌
                if(mkdir함수의 리턴값을 확인하여 같은 이름을 가진 디렉토리가 있는지 확
                인)
                {
                        없으면
                        strcat함수를 통해 dire에 생성할 파일이름을 저장함
                        open함수를 통해 파일 생성
```

```
                }
                else(디렉토리가 있으면)
                {
                        strcat함수를 통해 dire에 생성할 파일이름을 저장함
                        if(같은 이름의 파일이 없으면)
                        {
                                open함수를 통해 파일 생성
                                close함수를 통해 종료
                        }
                        else(있으면)
                        close함수를 통해 종료
                }
        }
}
```

discrimination 함수

int discrimination(char* first_HS_URL(디렉토리 이름), char* second_HS_URL(파일 이름),char * URL(입력 URL))

{

홈디렉토리의 위치를 저장하기위한 문자형 변수 dire2선언

getHomeDir함수를 통해 홈 디렉토리의 위치를 dire2에 저장

해쉬된 디렉토리가 있는 위치를 dire2변수를 strcat함수로 통해 저장

디렉토리를 열기위해 DIR*형 변수 dp 선언

해쉬 디렉토리안에 해쉬 파일을 열기위해 dirent* 구조체 변수 entry 선언

if 해쉬 디렉토리가 없으면

{

형식에 맞춰 Miss문을 텍스트 파일에 작성한다.

```
getHomeDir(dire2);
strcat(dire2, "/logfile/");
strcat(dire2, "logfile.txt");
int fd3;
fd3 = open(dire2, O_WRONLY | O_APPEND, 0777);
write(fd3, "[Miss]", strlen("[Miss]"));
write(fd3, URL, strlen(URL));
write(fd3, "-[", strlen("-["));
time_t now;
struct tm* ltp;
time(&now);
ltp = localtime(&now);
char st[100];
memset(st, 0, 100);
sprintf(st, "%d", (ltp->tm_year+1900));
write(fd3, st, strlen(st));
write(fd3, "/", strlen("/"));
sprintf(st, "%d", (ltp->tm_mon));
write(fd3, st, strlen(st));
write(fd3, "/", strlen("/"));
sprintf(st, "%d", (ltp->tm_mday));
write(fd3, st, strlen(st));
write(fd3, ", ", strlen(", "));
sprintf(st, "%d", (ltp->tm_hour));
write(fd3, st, strlen(st));
write(fd3, ":", strlen(":"));
sprintf(st, "%d", (ltp->tm_min));
write(fd3, st, strlen(st));
write(fd3, ":", strlen(":"));
```

```
            sprintf(st, "%d", (ltp->tm_sec));
            write(fd3, st, strlen(st));
            write(fd3, "]₩n", strlen("]₩n"));
            close(fd3);
            closedir(dp);
            Miss인 경우 1을 반환
            함수종료
}
해쉬 디렉토리가 있으면 while문을 통해 디렉토리내 파일 확인
{
            if 디렉토리내부의 해쉬파일이 입력받은 URL과 같은지 확인
            {
                        해쉬파일이 같으면 형식에 맞춰 출력
                        getHomeDir(dire2);
                        strcat(dire2, "/logfile/");
                        strcat(dire2, "logfile.txt");
                        int fd4;
                        fd4 = open(dire2, O_WRONLY | O_APPEND, 0777);
                        write(fd4, "[Hit]", strlen("[Hit]"));
                        write(fd4, first_HS_URL, strlen(first_HS_URL));
                        write(fd4, "/", strlen("/"));
                        write(fd4, second_HS_URL, strlen(second_HS_URL));
                        write(fd4, "-[", strlen("-["));
                        time_t now2;
                        struct tm* gtp;
                        time(&now2);
                        gtp = localtime(&now2);
                        char st2[100];
                        memset(st2, 0, 100);
                        sprintf(st2, "%d", (gtp->tm_year + 1900));
                        write(fd4, st2, strlen(st2));
                        write(fd4, "/", strlen("/"));
                        sprintf(st2, "%d", (gtp->tm_mon));
                        write(fd4, st2, strlen(st2));
                        write(fd4, "/", strlen("/"));
                        sprintf(st2, "%d", (gtp->tm_mday));
                        write(fd4, st2, strlen(st2));
                        write(fd4, ", ", strlen(", "));
                        sprintf(st2, "%d", (gtp->tm_hour));
                        write(fd4, st2, strlen(st2));
                        write(fd4, ":", strlen(":"));
```

```
                    sprintf(st2, "%d", (gtp->tm_min));
                    write(fd4, st2, strlen(st2));
                    write(fd4, ":", strlen(":"));
                    sprintf(st2, "%d", (gtp->tm_sec));
                    write(fd4, st2, strlen(st2));
                    write(fd4, "]₩n", strlen("]₩n"));
                    write(fd4, "[Hit]", strlen("[Hit]"));
                    write(fd4, URL, strlen(URL));
                    write(fd4, "₩n", strlen("₩n"));
                    close(fd4);
                    closedir(dp);
                    Hit인 경우 2을 반환
                    함수종료
            }
}
```

같은 이름의 해쉬 디렉토리는 있었지만 해쉬파일이 없는 경우 while문을 빠져나온다 그리고 이 경우도 Miss임으로 Miss문을 형식에 맞게 파일에 입력

```
getHomeDir(dire2);
strcat(dire2, "/logfile/");
strcat(dire2, "logfile.txt");
int fd5;
fd5 = open(dire2, O_WRONLY | O_APPEND, 0777);
write(fd5, "[Miss]", strlen("[Miss]"));
write(fd5, URL, strlen(URL));
write(fd5, "-[", strlen("-["));
time_t now3;
struct tm* etp;
time(&now3);
etp = localtime(&now3);
char st3[100];
memset(st3, 0, 100);
sprintf(st3, "%d", (etp->tm_year + 1900));
write(fd5, st3, strlen(st3));
write(fd5, "/", strlen("/"));
sprintf(st3, "%d", (etp->tm_mon));
write(fd5, st3, strlen(st3));
write(fd5, "/", strlen("/"));
sprintf(st3, "%d", (etp->tm_mday));
write(fd5, st3, strlen(st3));
write(fd5, ", ", strlen(", "));
sprintf(st3, "%d", (etp->tm_hour));
```

```
            write(fd5, st3, strlen(st3));
            write(fd5, ":", strlen(":"));
            sprintf(st3, "%d", (etp->tm_min));
            write(fd5, st3, strlen(st3));
            write(fd5, ":", strlen(":"));
            sprintf(st3, "%d", (etp->tm_sec));
            write(fd5, st3, strlen(st3));
            write(fd5, "]₩n", strlen("]₩n"));
            close(fd5);
            closedir(dp);
            Miss인 경우 1을 반환
            프로그램 종료

}
```

bye 함수

void bye(int hitcount(hit개수), int misscount(miss개수), int result(프로그램 런타임 시간을 저장함))

{

홈 디렉토리 위치를 저장할 문자형 배열 변수 dire3선언

getHomeDir 함수와 strcat함수를 적절히 사용해 로그텍스트파일 위치를 dire3에 저장

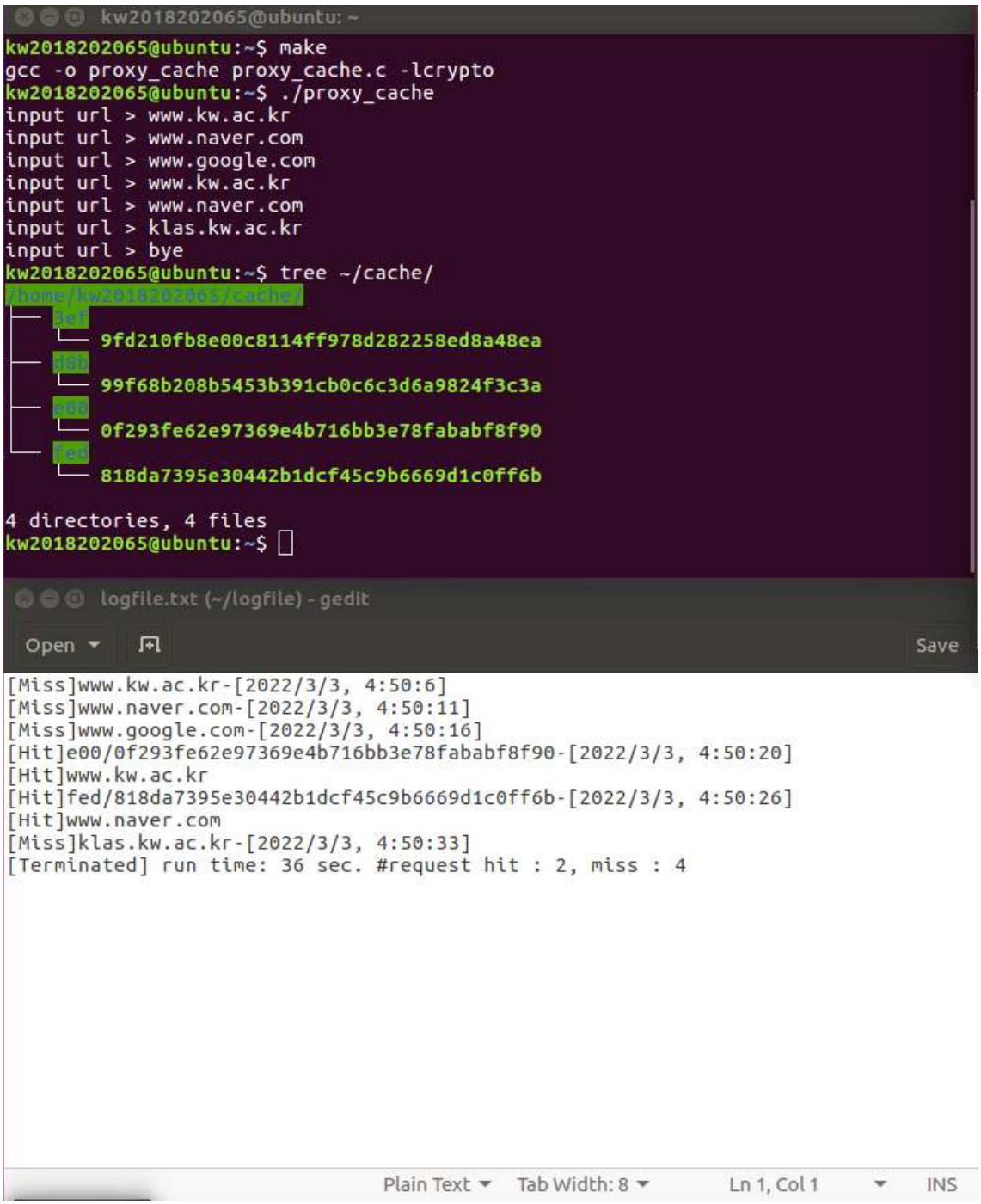
로그텍스트파일 오픈

// 아래는 종료문을 형식에 맞게 입력하는 과정

```
write(fd5, "[Terminated] ", strlen("[Terminated] "));
write(fd5, "run time: ", strlen("run time: "));
char st3[100];
sprintf(st3, "%d", result);
write(fd5, st3, strlen(st3));
write(fd5, " sec. ", strlen(" sec. "));
write(fd5, "#request hit : ", strlen("#request hit : "));
sprintf(st3, "%d", hitcount);
write(fd5, st3, strlen(st3));
write(fd5, ", ", strlen(", "));
write(fd5, "miss : ", strlen("miss : "));
sprintf(st3, "%d", misscount);
write(fd5, st3, strlen(st3));
write(fd5, "₩n", strlen("₩n"));
close(fd5);
return;
```

}

-결과 화면



컴파일 후 프로그램을 실행한 뒤

1. www.kw.ac.kr -> miss인 경우
2. www.naver.com -> miss인 경우
3. www.google.com -> miss인 경우
4. www.kw.ac.kr -> hit인 경우
5. www.naver.com -> hit인 경우
6. klas.kw.ac.kr -> miss인 경우

7. bye -> 프로그램 종료
7줄의 입력에 따른 로그 텍스트파일에 제대로 입력이 됨을 알 수 있으며
tree명령어를 통해 확인한 cache디렉토리의 내부는 miss의 개수인 4개의 디렉토리가 생성됨을 알 수 있다.

- 고찰

이번 과제를 진행하면서 프로그램의 설계에 있어 main함수에서 모든 과정을 다 수행하는 알고리즘 보다는 여러 함수를 사용하여 수행하는 알고리즘으로 설계하였고 Hit, miss를 판별하는 discrimination함수와 bye함수를 통해 프로그램을 구현하였다 이를 통해 1-1과제에서 구성한 main에 많은 변형을 주는 것이 아닌 조금의 변형만으로 프로그램을 구동할 수 있었다.