

시스템 프로그래밍 실습 Report

실습 제목: Proxy #1-1

실습일자: 2022년 3월 24일 (목)

제출일자: 2022년 3월 30일 (수)

학 과: 컴퓨터정보공학부

담당교수: 최상호 교수님

실습분반: 목요일 7,8교시

학 번: 2018202065

성 명: 박 철 준

- Introduction

1. 제목

Proxy 1-1

2. 목적

가. mkdir()을 이용하여 디렉터리 생성시 발생하는 권한 문제를 파악 후 발생 이유와 해결법을 알아낸다.

나. Proxy 서버를 구현하기 위한 요구사항 중 가장 첫 번째로 URL을 받고 이를 sha1_hash 함수를 통해 Hashed URL로 변환한다. 두번째로 getHomedir 함수를 사용하여 Home directoty path를 얻고 Home diretor에 cache 디렉토리를 생성한다. 프로그램은 Bye command를 입력으로 받았을 때 종료한다. 마지막으로 디렉토리의 이름은 Hashed URL의 앞 세글자로 한다. 파일 생성시 다양한 함수의 사용이 가능하다.

● 실습 결과

1. mkdir()을 이용하여 디렉터리 생성 시 발생하는 권한 문제

가장 먼저 test1.c라는 파일을 생성하여 아래와 같이 mkdir 함수에 모든 권한 777을 준다고 가정하여 코드를 작성하였다.

[illegible]

그 후 test1.c를 컴파일 하여 practice1이라는 디렉터리를 만들었고 ls -al을 통해 권한을 확인 하면 아래와 같은데

```
kw2018202065@ubuntu: ~/practice
kw2018202065@ubuntu:~/practice$ ./test1 practice1
kw2018202065@ubuntu:~/practice$ ls
Makefile  practice1  test1  test1.c  test2.c
kw2018202065@ubuntu:~/practice$ ls -al
total 36
drwxrwxrwx  3 kw2018202065 kw2018202065 4096 Mar 26 07:15 .
drwxr-xr-x 18 kw2018202065 kw2018202065 4096 Mar 26 07:13 ..
-rw-rw-r--  1 kw2018202065 kw2018202065   96 Mar 26 07:09 Makefile
drwxrwxr-x  2 kw2018202065 kw2018202065 4096 Mar 26 07:15 practice1
-rwxrwxr-x  1 kw2018202065 kw2018202065 8656 Mar 26 07:13 test1
-rw-----  1 kw2018202065 kw2018202065  207 Mar 26 04:14 test1.c
-rw-----  1 kw2018202065 kw2018202065  232 Mar 26 04:34 test2.c
kw2018202065@ubuntu:~/practice$
```

사용자, 그룹, other에게 모든 권한을 주었지만 생성된 디렉토리는 775의 값을 가지며 other에게는 w권한은 빠진 것을 알 수 있다.

이는 mkdir()을 사용할 때 2번째 인자 값은 현재 시스템의 umask 설정값에 의해 조정되어 지는데 mkdir의 mode 값을 0777로 지정함에도 불구하고 0775의 권한으로 디렉토리가 생성된다면 시스템의 umask의 값이 0002로 설정되어 있기 때문이다.(때론 0022로 설정 되어 있을 수 도 있다.) 이를 해결하기 위해서는 시스템의 umask 설정값을 바꿔주어야 하고 이는 아래에서 사진과 함께 설명하겠다

```
kw2018202065@ubuntu: ~/practice
#include<stdio.h>
#include<sys/types.h>
#include<sys/stat.h>
void main(int argc, char* argv[])
{
    if (argc < 2) {
        printf("error\n");
        return;
    }
    umask(0000);

    mkdir(argv[1], S_IRWXU | S_IRWXG | S_IRWXO);
}
~
~
~
~
~
~
"test2.c" [dos] 16L, 232C
```

mkdir()함수를 사용하기 전에 umask(0000)의 함수를 사용함으로써 umask는 생성파일의 권한을 빼앗은 역할을 하지만 umask(0000)은 설정값을 0으로 하기 때문에 권한을 빼앗는 역할을 하지 못한다.

```
kw2018202065@ubuntu: ~/practice
kw2018202065@ubuntu:~/practice$ gcc -o test2 test2.c
kw2018202065@ubuntu:~/practice$ ls
Makefile practice1 test1 test1.c test2 test2.c
kw2018202065@ubuntu:~/practice$ ./test2 practice2
kw2018202065@ubuntu:~/practice$ ls -al
total 52
drwxrwxrwx  4 kw2018202065 kw2018202065 4096 Mar 26 07:31 .
drwxr-xr-x 18 kw2018202065 kw2018202065 4096 Mar 26 07:30 ..
-rw-rw-r--  1 kw2018202065 kw2018202065   96 Mar 26 07:09 Makefile
drwxrwxr-x  2 kw2018202065 kw2018202065 4096 Mar 26 07:15 practice1
drwxrwxrwx  2 kw2018202065 kw2018202065 4096 Mar 26 07:31 practice2
-rwxrwxr-x  1 kw2018202065 kw2018202065 8656 Mar 26 07:13 test1
-rw-----  1 kw2018202065 kw2018202065  207 Mar 26 04:14 test1.c
-rwxrwxr-x  1 kw2018202065 kw2018202065 8704 Mar 26 07:30 test2
-rw-----  1 kw2018202065 kw2018202065  232 Mar 26 04:34 test2.c
kw2018202065@ubuntu:~/practice$
```

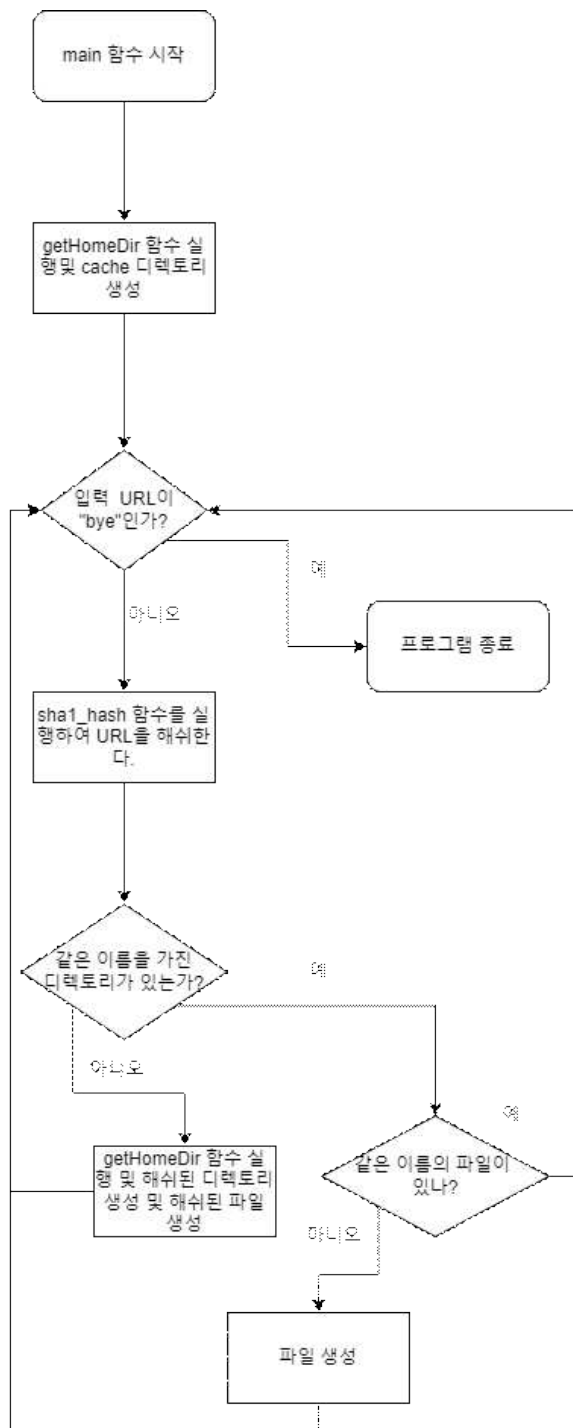
test2.c파일을 컴파일한 후 test2실행파일을 실행하여 practice2디렉토리를 생성하면 ls -al을

통해 권한을 보면 모든 권한을 받아온 것을 확인 할 수 있다.

2. proxy #1-1

-Flow Chart

먼저 proxy_cache프로그램의 main함수 플로우 차트는 아래와 같다.



-Pseudo code

먼저 proxy_cache프로그램의 main함수의 Pseudo code는 아래와 같다.

```
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
    getHomeDir함수를 실행해서 dire에 홈디렉토리 위치 저장
    strcat 함수로 /cahe/를 dire에 붙이기
    umask 함수로 초기 설정된 umask값 변경
    mkdir함수로 홈디렉토리위치에 cache폴더 생성
    while(1){
        변수 URL에 사용자의 URL입력받아 저장하기
        if(strcmp함수로 변수 URL이 "bye"인지 확인)
        {
            맞으면 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
        getHomeDir함수를 통해 dire변수에 홈디렉토리 위치 저장
        strcat으로 dire에 /cache/와 first_HS_URL변수를 붙여줌
        if(mkdir함수의 리턴값을 확인하여 같은 이름을 가진 디렉토리가 있는지 확인)
        {
            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
            }
            else(있으면)
            {아무것도 안함}
        }
    }
}
```

-결과 화면

```
kw2018202065@ubuntu:~$ make
gcc -o proxy_cache proxy_cache.c -lcrypto
kw2018202065@ubuntu:~$ ./proxy_cache
input url > www.kw.ac.kr
input url > www.google.com
input url > www.naver.com
input url > klas.kw.ac.kr
input url > bye
kw2018202065@ubuntu:~$ tree ~/cache/
/home/kw2018202065/cache/
├── 0c3
│   └── bd0638bffc16ea24ad61ae0db8ea48feb43b5
├── 83f
│   └── dc9358608064df2f9e490717a3f7a3bcdcd298
├── 917
│   └── bdfffeb7aca51e8c7dde9a99c0bb336f546e0
└── 123
    └── f8d6b6bd62a6cd49204b11c63765b1c6cf919

4 directories, 4 files
kw2018202065@ubuntu:~$
```

proxy_cache.c를 컴파일 하고 proxy_cach를 실행하였고 총 4개의 사이트를 입력하였다. 후에 tree ~/cache/를 통해 디렉토리와 파일을 확인한 모습이다. 올바른 결과 값을 가짐을 알 수 있다.

- 고찰

이번 과제를 진행하면서 프로그램의 main함수에서 getHomeDir함수와 sha1_hash함수가 어떤 알고리즘으로 사용되어야 과제의 목표를 이룰 수 있는지 생각해 보는 시간이 가장 중요했고 이를 해결하기 위해 플로우 차트를 그려보며 가장 최적의 구조를 설계를 하여 구현할 수 있었다. 이를 통해 어떠한 시스템 프로그램을 설계함에 있어서 다양한 함수들이 사용될 것이며 각 함수들은 알맞은 파트에 최적의 구조를 가지게 설계해야하는 중요함을 느낄 수 있었다.