

시스템 프로그래밍 (월요일)

최상호 교수님

Proxy 1-1

컴퓨터정보공학부 2018202076 이연걸

mkdir() (Proxy는 후술)

mkdir은 others에게 w권한을 주지 않는다. 하지만 과제에서 제시된 코드는 other에게 모든 권한을 주는 S_IRWXO를 사용하였으므로 권한 문제가 발생한다. 이를 해결하기 위해선 umask를 사용해 권한을 제거해 주어야 한다.

문제를 해결한 소스코드:

```
#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(002);

    mkdir(argv[1], S_IRWXU | S_IRWXG | S_IRWXO);
}
```

문제 해결 장면:

```
kw2018202076@ubuntu:~/sslabs$ ls
a.out  makeDIR.c  proxy
kw2018202076@ubuntu:~/sslabs$ ./a.out testDir
kw2018202076@ubuntu:~/sslabs$ ls -l
total 24
-rwxrwxr-x 1 kw2018202076 kw2018202076 8704 Mar 29 05:58 a.out
-rw-rw-r-- 1 kw2018202076 kw2018202076 214 Mar 29 05:58 makeDIR.c
drwxrwxr-x 3 kw2018202076 kw2018202076 4096 Mar 30 01:08 proxy
drwxrwxr-x 2 kw2018202076 kw2018202076 4096 Mar 30 01:16 testDir
```

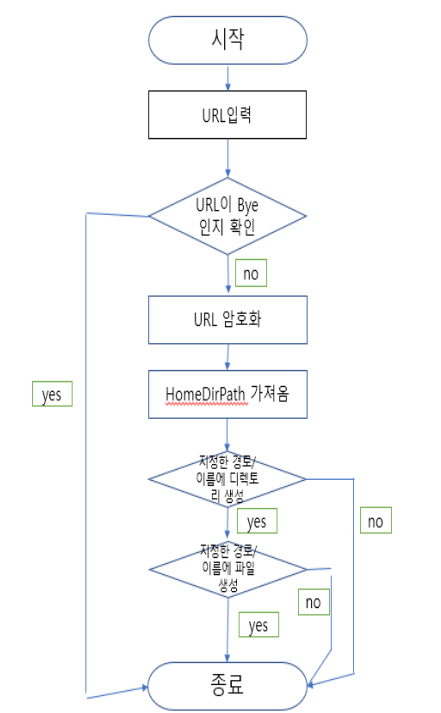
Proxy

Introduction:

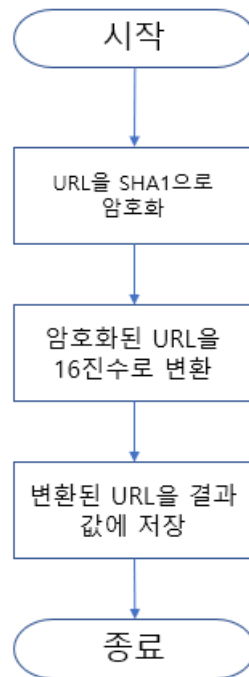
이번 과제는 사용자로부터 입력 받은 url을 암호화하고 암호화된 url 중 일부를 이름으로 사용해 디렉토리와 파일을 만드는 과제다. 본 과제를 수행하기 위해 가장 중요한 것은 url의 암호화다. 이를 위해 SHA1이 사용된다. SHA1은 최대 2^{64} 크기의 입력 값을 160bit의 출력 값으로 바꿔주는 암호화 알고리즘인데 openssl/sha.h header file을 include하여 사용한다. 암호화된 url중 첫 3자리를 사용해 Directory를 만들고 나머지 자리들을 사용해 만들어진 Directory내에 File을 만든다.

지금까지의 설명해서 알 수 있듯 문자열을 변환하고 그 값을 다시 빈 문자열에 담아주고, 문자열의 원소를 추출해 Directory와 File을 만들기 때문에 문자열을 다뤄야 하고, 이를 쉽게 하기 위한 라이브러리(header file)이 필요한 과제이다.

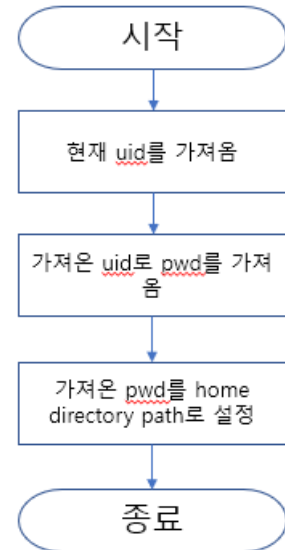
Flow Chart:



Main



sha1_hash



getHomeDir

Pseudo code:

프로그램 시작

Bye가 입력되거나 에러가 생기기 전까지 반복

입력 URL, 암호화된 URL 동적할당

URL 입력

URL이 bye인지 확인

bye라면 프로그램 종료

URL 암호화

Home Directory Path 경로 추출

암호화된 URL 첫 3글자를 이름으로 DIR 생성 후 성공여부 확인

실패라면 프로그램 종료

암호화된 URL의 4글자 이후 부분으로 DIR 하위에 파일 생성 후 성공여부 확인

실패라면 프로그램 종료

메모리 동적할당 해제

프로그램 종료

결과화면:

```
kw2018202076@ubuntu:~/sslslab/proxy$ ./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
kw2018202076@ubuntu:~/sslslab/proxy$ tree ~/cache/
/home/kw2018202076/cache/
├── 44b
│   └── fe3530e8aff9a4a38f9697e9b19dd29c83fd4
├── 4c2
│   └── 90a42e60308cc5413bac7534f3ec15b5d5817
├── 962
│   └── 180d9e0445fe2cae2017f8b8fa8fa83bc0cc4
└── a8c
    └── a7f78085b833812898caea609cf12d5e1a11b

4 directories, 4 files
kw2018202076@ubuntu:~/sslslab/proxy$
```

고찰:

이번 과제에서는 제공된 암호화 알고리즘 보다는 문자열, 포인터, 그리고 메모리가 중요했던 것 같다. URL을 입력 받고 암호화를 하고 배열을 쪼개서 이름으로 사용하고 이런 과정들을 고정된 메모리로 진행하였더니 overflow가 계속 발생해 결국 메모리 동적할당으로 해결할 수 있었다. 그리고 문자열의 끝에 삽입되는 종결문자가 계속 파일 이름으로 사용되어서 이를 제거해주는 과정 역시 문자열에 대한 이해가 필요했다. 메인 함수를 포함한 3개의 함수에서 string.h 헤더파일이 쓰였는데, 이 헤더파일이 없었다면 한 줄로 끝낼 수 있는 코드를 반복문과 조건문을 섞어서 여러 줄로 끝냈을 것이다. 중요한 점은 위와 같지만 새롭게 알 수 있던 점은 cache와 proxy의 개념이다. Backend 서비스를 개발하면서도 용어만 알지 개념에 대해서는 정확히 알지 못했는데 수업과 강의 자료를 통해 이해가 높아졌다.