
System Programming

Assignment#1-1

02_Proxy 1-1

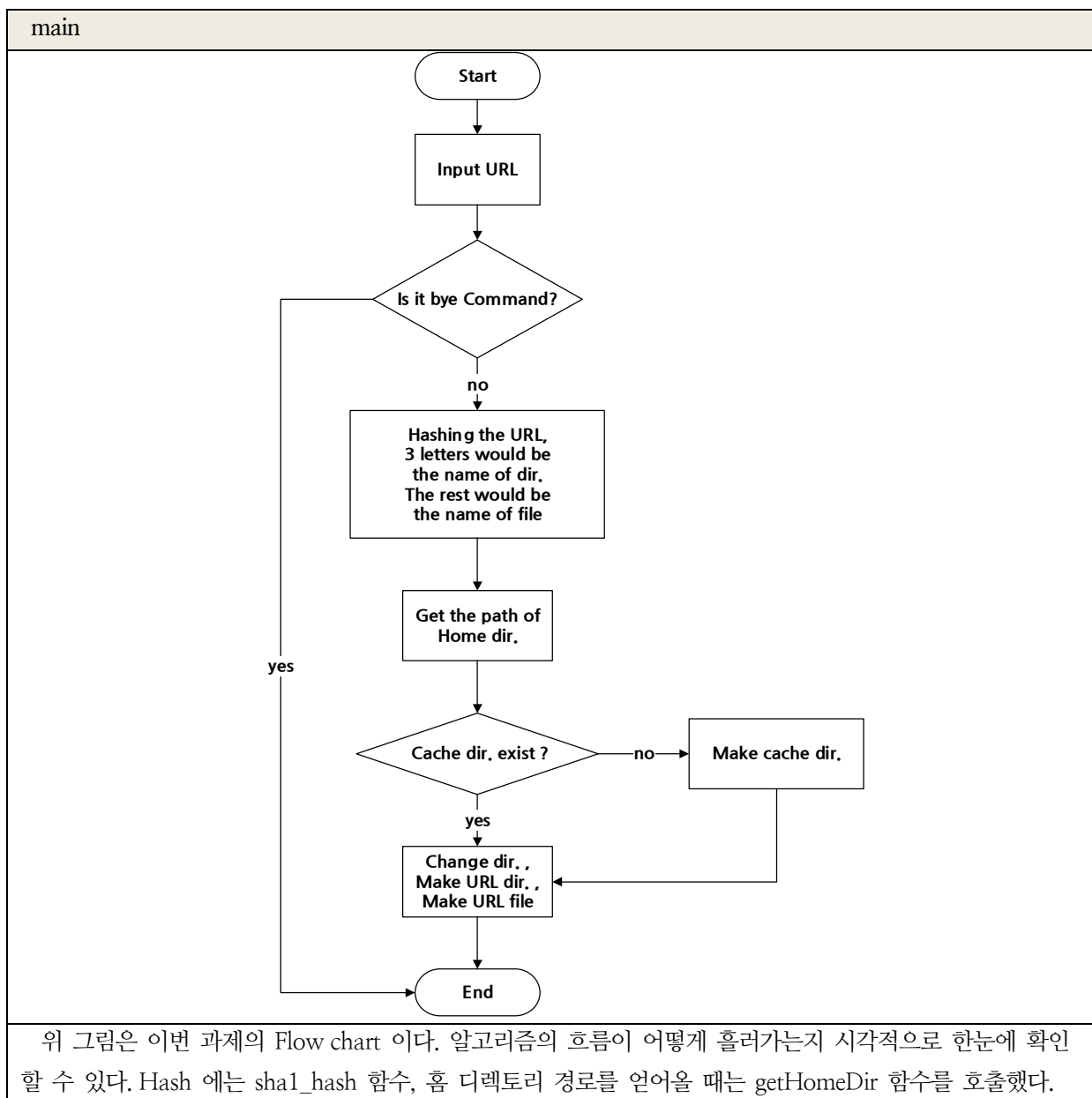


Professor	목 3 4 황호영 교수님
Department	컴퓨터공학과
Student ID	2012722028
Name	장 한 별
Date	2018. 03. 30

Introduction.

시스템 프로그래밍 강의 시간에 배운 proxy server 를 구현하는 것을 목표로 한다. 이번 과제는 그 중에서 가장 처음 부분인 URL 을 입력 받고, 입력 받은 URL을 SHA1 알고리즘을 이용해 Hashing 한다. 이때, Hashed URL 의 앞 3글자는 directory 이름, 나머지는 생성할 파일 이름이 된다. 올바르게 생성되었는지 확인하기 위해 tree 형태로 출력하는 것이 이번 과제의 내용이다.

Flow chart.



Pseudo code.

main

```
Int main(int argc, char* argc[])
{
    Input URL;
    If(input == "bye") break;
    Hashing;
    String copy(directory name, hashed URL, 3);
    String copy(file name, heshed URL +3);

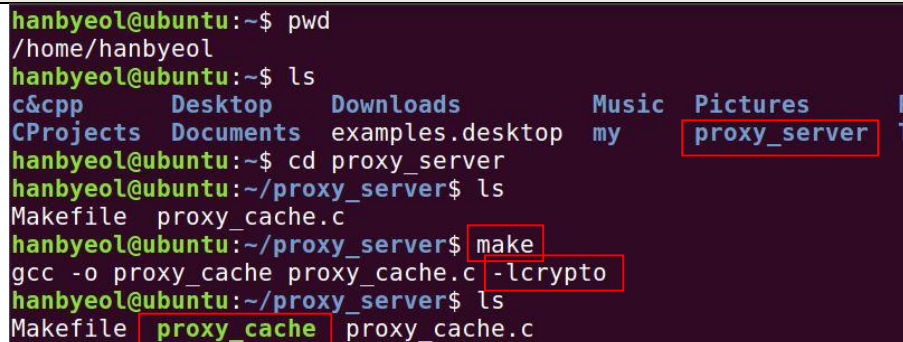
    If(cache == NULL)
        Make directory (cache);

    If(directory name==NULL)
    {
        Make directory(Hashed URL);
        Change directory(Hashed URL);
        Make file(Hashed URL);
    }
}
```

위 그림은 이번 과제 main 함수의 pseudo code 이다. 실제 c 언어로 어떻게 작성되었는지 간략하게 설명하여 코드에 대한 이해도를 높일 수 있다.

Result.

1-1.(1)



```
hanbyeol@ubuntu:~$ pwd
/home/hanbyeol
hanbyeol@ubuntu:~$ ls
c&cpp      Desktop    Downloads  Music      Pictures    P
CProjects  Documents  examples.desktop  my         proxy_server  T
hanbyeol@ubuntu:~$ cd proxy_server
hanbyeol@ubuntu:~/proxy_server$ ls
Makefile  proxy_cache.c
hanbyeol@ubuntu:~/proxy_server$ make
gcc -o proxy_cache proxy_cache.c -lcrypto
hanbyeol@ubuntu:~/proxy_server$ ls
Makefile  proxy_cache  proxy_cache.c
```

위 그림은 proxy_server 라는 directory 에 Make file 과 proxy_cache.c 파일을 생성 및 이번 과제를 모두 구현한 후 make 했을 때(SHA-1 library를 사용한 코드이기 때문에 -lcrypto 옵션을 추가한 것을 확인할 수 있다.)의 그림이다. 올바르게 실행파일이 만들어진 것을 확인할 수 있다.

1-1.(2)

```
hanbyeol@ubuntu:~/proxy_server$ ./proxy_cache
input URL> www.naver.com
input URL> www.google.com
input URL> www.daum.net
input URL> bye
hanbyeol@ubuntu:~/proxy_server$
```

위 그림은 그림 1-1.(1) 에서 만든 실행파일을 실행 시켰을 때의 그림이다. './proxy_cache' 는 '현재 directory 안에 있는 proxy_cache 라는 실행파일을 실행시켜라' 라는 뜻이다. Input URL 로 3개의 웹사이트(www.naver.com, www.google.com, www.daum.net) 을 입력한 후, bye 를 입력했다. Bye를 입력 시 곧바로 종료된 것을 확인할 수 있다.

1-1.(3)

```
hanbyeol@ubuntu:~/proxy_server$ cd
hanbyeol@ubuntu:~$ ls
cache      Desktop    examples.desktop  Pictures
c&c++      Documents  Music             proxy_server
CProjects  Downloads  my               Public
hanbyeol@ubuntu:~$ cd cache
hanbyeol@ubuntu:~/cache$ ls
025  d8b  fed
```

Home directory를 확인해보면 cache directory 가 올바르게 생성된 것을 확인할 수 있다. 그 후 cache directory 로 cd 한 후 확인해보니 1-1.(2)에서 입력한 세 URL 이 Hashing 되어져 그 Hashed URL 의 앞 3글자를 딴 directory 가 만들어져 있는 것을 확인할 수 있다.

1-1.(4)

```
hanbyeol@ubuntu:~$ tree ~/cache/
/home/hanbyeol/cache/
├── 025
│   └── 15aba49d232baf9567495aa44eb8d20b2b764
├── d8b
│   └── 99f68b208b5453b391cb0c6c3d6a9824f3c3a
└── fed
    └── 818da7395e30442b1dcf45c9b6669d1c0ff6b

3 directories, 3 files
hanbyeol@ubuntu:~$
```

위 그림은 1-1.(3)의 과정 후, tree 형태로 출력한 그림이다. Hashed URL 중 directory 이름을 제외한 나머지가 file 이름으로 올바르게 생성되었음을 확인할 수 있다. Directory 개수와 file 개수도 확인할 수 있다. 이때, cache directory나 새롭게 생성된 directory 들이 초록색 바탕으로 칠해져있는 것을 확인할 수 있는데, 이것은 User, Group, Others 에게 R, W, X(read, write, execute)권한을 모두 부여했다는 뜻 이다.

1-1.(5)

```
hanbyeol@ubuntu:~$ tree ~/cache/
/home/hanbyeol/cache/
├── 025
│   └── 15aba49d232baf9567495aa44eb8d20b2b764
├── d8b
│   └── 99f68b208b5453b391cb0c6c3d6a9824f3c3a
└── fed
    └── 818da7395e30442b1dcf45c9b6669d1c0ff6b

3 directories, 3 files
hanbyeol@ubuntu:~$
```

만약 mkdir 에 0777 권한을 부여해 directory를 생성하게 되면, 위와 같이 모든 권한을 할당 받지 못한다. 그 이유는 현재 사용하고 있는 OS 의 기본 설정이 umask(002)로 되어있는데 0777 권한 에서 002 이 를 빼면 0775 즉, Others 에서 W 권한이 빠지게 된다. umask(000)을 코드에 추가해주면 이러한 문제점을

해결할 수 있고, 1-1.(4) 와 같이 올바른 출력을 확인할 수 있다.

Conclusion.

이번 과제는 URL 을 입력시켜 Hashing 후 Hashed URL 로 directory 와 file 을 생성하는 것이었다. 지난 과제에서 쓰인 명령어들을 직접 C언어로 구현하는 것이 참 흥미로웠다. 하지만 처음부터 흥미로웠던 것은 아니었다. 어떤 헤더파일을 추가시켜야 하는지, sha1_hash 함수나 getHomeDir 함수에 어떤 인자를 입력시키고 어떻게 반환되어 나오는지 이해가 가지않았다. 그리고 main 함수를 구현함에 있어 문자열(string) 관련 함수들을 오랜만에 쓰게 되어 문법들이 많이 헷갈려서 어떤 식으로 코딩 해야겠다는 생각은 잡아놨지만 실제로 구현하는 것이 쉽지만은 않았다.

Sha1_hash 함수는 main 함수에서 입력된 URL을 SHA1 알고리즘을 통해 Hash 한 후 Hashed URL을 반환한다는 사실과 getHomeDir 함수를 이용해 home directory 의 경로를 얻어온 다는 사실을 이해한 후 확실히 감이 오기 시작했고, OS Linux 보다 비교적 디버깅이 쉬운 OS Window 에서 Visual Studio 툴을 사용해 string 관련 함수들(strcpy, strncpy, strcat 등)을 연습해봄으로서 이번 과제에 실제로 쓰이는 함수들을 더욱 확실하게 익힐 수 있었다.

이번 과제는 MISS(입력한 URL 이 cache 에 없을 때) 까지만 구현하는 것이 목표이므로 HIT 상황일 때, 같은 Directory 안에 다른 파일을 생성해야 할 때, 사용자가 임의로 File 만을 삭제할 때 등 예외 처리는 1-2 부터 구현하도록 하겠다.