

시스템프로그래밍



과 제 명 : Proxy 1-1

교 수 님 : 최상호 교수님

강의시간 : 월요일

학 번 : 2019202100

이 름 : 하주영

1. umask

mkdir() 이용하여 디렉토리 생성 시 발생하는 권한 문제를 umask(0)을 사용하여 해결하고 발생한 이유를 분석한다.

```
#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(0);
    mkdir(argv[1], S_IRWXU | S_IRWXG | S_IRWXO );
}
```

이유: 기본 umask 값이 '002'로 설정되어 있기 때문이다. 기본 umask 값이 '0'이 아닌 숫자인 이유는 만약 파일이나 디렉토리를 생성했을 때 모든 권한을 준다면 제 3자가 이 파일을 마음 대로 수정하거나 없애 버리는 것이 가능하므로 그 행위를 막기 위함이다. 기본 umask 값은 사용하고 있는 계정에 따라 다르며 명령어 umask 로 확인이 가능하다.

```
kw2019202100@ubuntu:~/1-1$ umask
0002
```

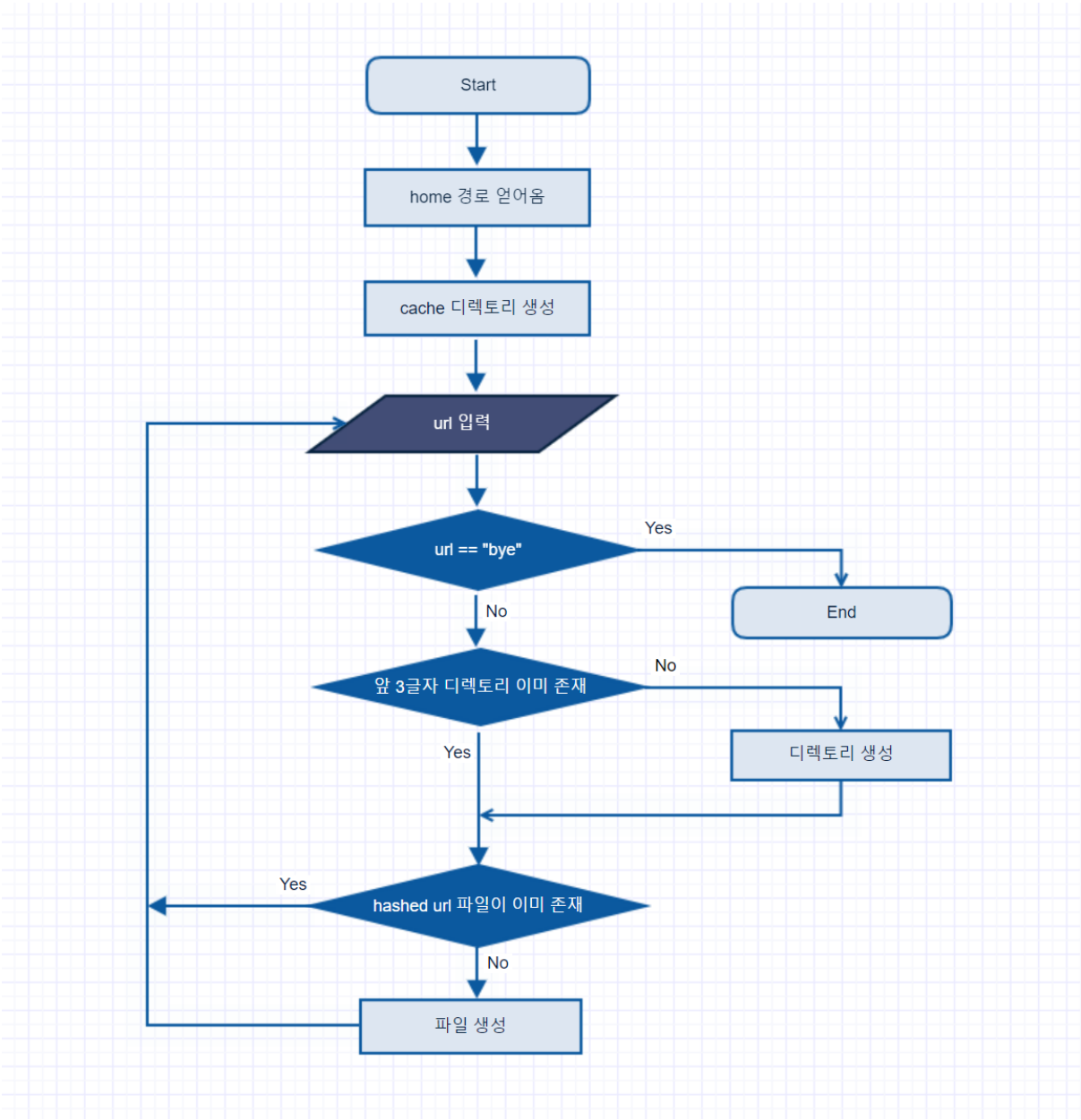
<- umask 명령어를 사용하여 확인

2. proxy #1-1

Introduction

url을 입력하면 제공된 sha1_hash 함수를 사용하여 입력 받은 hashed url로 변환한다. 제공된 getHomedir 함수를 사용하여 home 경로를 얻어 home 밑에 cache 디렉토리를 생성하고 그 밑에 hashed url의 3글자를 따 디렉토리를 만든다. 그리고 hashed url 디렉토리 밑에 파일 명이 hashed url인 파일을 생성한다. 최종적으로 proxy server의 cache를 만드는 기초 과정을 이해하는 것이 이번 과제의 목적이다..

FlowChart

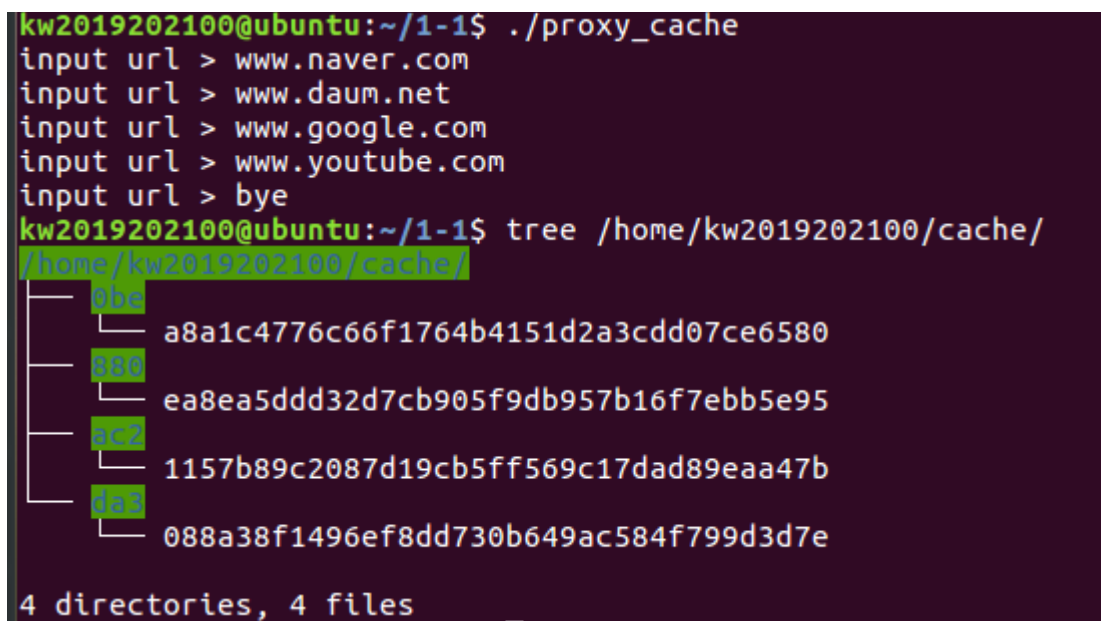


Pseudo code

```
int main() {  
    char url[256], hashed_url[256], char home[256], cache_path[256];  
    char d_name[4], d_path[256], char f_name[256], f_path[256];  
    getHomeDir(home);  
    home경로 밑에 cache디렉토리 경로를 추가하여 그 경로를 cache_path에 저장  
    mkdir(cache_path, 0777);  
    while( url != "bye" ) {  
        url 을 입력함;  
        sha1_hash(url, hashed_url);  
        hashed_url의 앞 세글자를 d_name 에 저장함;  
        cache경로 밑에 d_name디렉토리 경로를 추가하여 그 경로를 d_path에 저장  
        mkdir(d_path, 0777);  
        hashed_url의 3글자부터 끝까지 추출하여 f_name에 저장  
        d_path에 f_name파일 경로를 추가하여 그 경로를 f_path에 저장  
        open(f_path, O_WRONLY|O_TRUNC|O_CREAT, 0644);  
    }  
}
```

결과화면

처음 실행했을 때 4개의 디렉토리와 파일이 생성됨.



```
kw2019202100@ubuntu:~/1-1$ ./proxy_cache  
input url > www.naver.com  
input url > www.daum.net  
input url > www.google.com  
input url > www.youtube.com  
input url > bye  
kw2019202100@ubuntu:~/1-1$ tree /home/kw2019202100/cache/  
/home/kw2019202100/cache/  
├── 0be  
│   └── a8a1c4776c66f1764b4151d2a3cdd07ce6580  
├── 880  
│   └── ea8ea5ddd32d7cb905f9db957b16f7ebb5e95  
├── ac2  
│   └── 1157b89c2087d19cb5ff569c17dad89eaa47b  
└── da3  
    └── 088a38f1496ef8dd730b649ac584f799d3d7e  
  
4 directories, 4 files
```

같은 url을 입력했을 때 디렉토리와 파일이 생성되지 않음..

```
4 directories, 4 files
kw2019202100@ubuntu:~/1-1$ ./proxy_cache
input url > www.naver.com
input url > bye
kw2019202100@ubuntu:~/1-1$ tree /home/kw2019202100/cache/
/home/kw2019202100/cache/
├── 0be
│   └── a8a1c4776c66f1764b4151d2a3cdd07ce6580
├── 880
│   └── ea8ea5ddd32d7cb905f9db957b16f7ebb5e95
├── ac2
│   └── 1157b89c2087d19cb5ff569c17dad89eaa47b
└── da3
    └── 088a38f1496ef8dd730b649ac584f799d3d7e

4 directories, 4 files
```

고찰

처음에 파일을 생성했을 때 파일명 끝에 'w177'이 붙어 출력되는 이슈가 있었다.

어쩔 때는 제대로 파일명이 저장되고 어쩔 때는 'w177'이 저장되어 같은 디렉토리에 2개의 파일이 저장되는 문제점을 발생했다.

```
kw2019202100@ubuntu:~/1-1$ tree /home/kw2019202100/cache/
/home/kw2019202100/cache/
├── 11b
│   └── 2ec2184e2dfb823201bd46ddf5809b02547ba\177
├── 138
│   └── ab6b1a2a789c0a1e198809e79c9c0cdda4724\177
├── ac2
│   └── 1157b89c2087d19cb5ff569c17dad89eaa47b\177
├── da3
│   └── 088a38f1496ef8dd730b649ac584f799d3d7e\177
└── f45
    └── cca1dbc6aeaa2af430b76c8080b601adf5113\177

5 directories, 5 files
```

파일명 뒤에 이상한 문자가 붙었기 때문에 파일명 저장하는 부분을 유심히 살펴보았다. 문제가 발생한 부분은 hashed_url의 3번째부터 끝까지 추출하여 파일명을 저장하는 부분이었다.

처음에 작성한 코드는

```
for(int l = 0; l < strlen(hashed_url); i++) {
```

수정된 코드는

```
f_name[i-3] = hashed_url[i];    }
```

Reference

[\[리눅스\]기본 허가권,퍼미션 지정\(제어, 설정\)하기\(umask와 작동 원리\) 및 관련문제 \(tistory.com\)](http://tistory.com)