

# 시스템프로그래밍 과제 보고서

## Proxy #1-1

수업 명: 시스템프로그래밍 월5수6

담당 교수: 김태석

학과: 컴퓨터정보공학부

학번: 2023202070

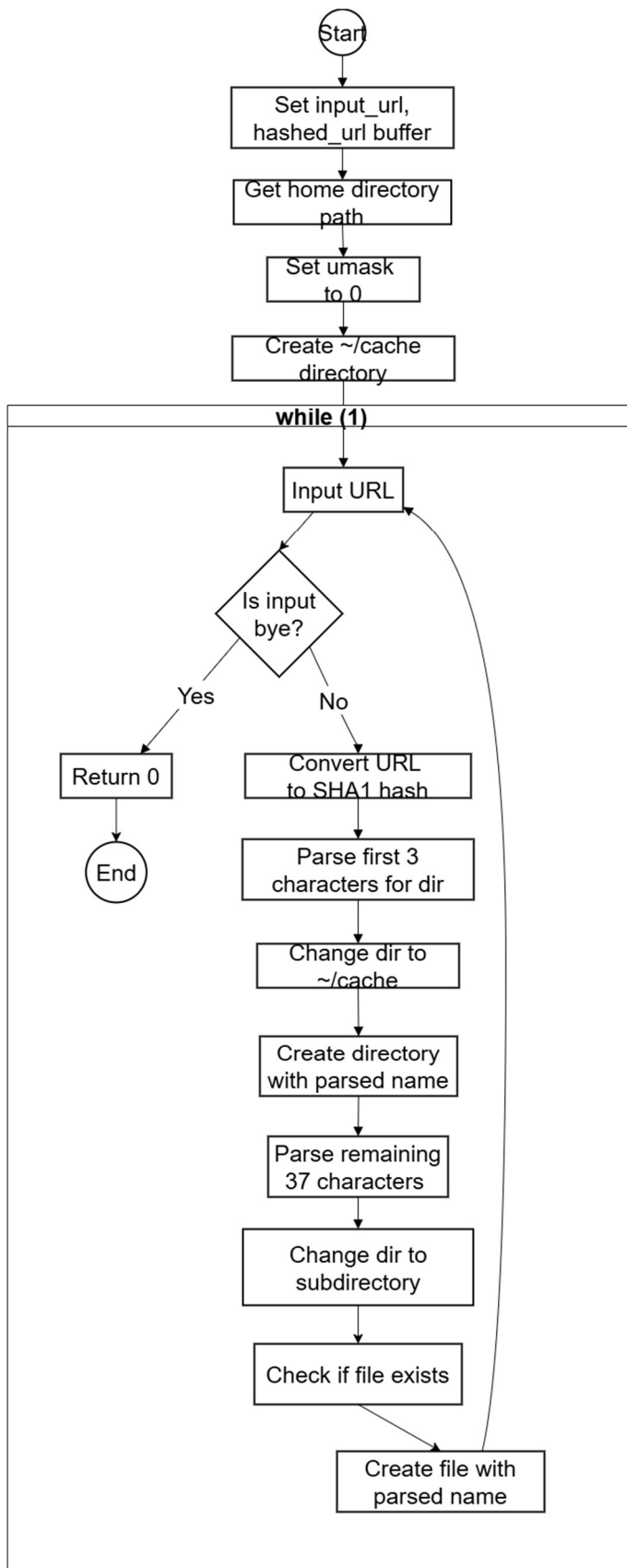
이름: 최현진

제출일: 2025.04.02

## Introduction

이번 Proxy #1-1 과제는 Ubuntu 환경에서의 시스템 프로그래밍 실습에서 기본적인 파일 생성과 관련된 함수들의 동작을 이해하고 이를 활용하여 proxy 캐시 기능을 구현하는 것을 포함한다. 과제는 크게 두 가지 주제로 나뉘는데, 먼저 mkdir()을 사용한 소스 코드에서 함수에 주어진 인자대로 권한이 설정되지 않는 문제를 점검한다. 해결하기 위해 mkdir()과 umask() 함수의 동작을 이해하는 과정이 포함된다. 최종적으로는 문제가 발생한 이유와 해결된 소스 코드를 첨부한다. 다음으로 입력 받은 URL들을 SHA1 해시값으로 변환하고, 이를 기반으로 캐시 디렉토리 및 파일을 생성하는 프로그램을 구현한다. 이를 위해 정의한 sha1\_hash() 함수는 URL을 SHA1 해시값으로 변환하기 위해 사용한다. getHomedir() 함수는 사용자 홈 디렉토리의 경로를 반환한다. mkdir() 함수 등에서 인자로 ~를 입력하여 홈에 접근하는 것이 불가능하기에 사용한다.

## Flow Chart



## Pseudo code

main:

input\_url, hashed\_url 버퍼 설정

home 디렉토리 경로 가져오기

umask 0 설정

루트 캐시 디렉토리(~/.cache) 생성

무한 반복

URL 입력 받기

입력이 "bye"면 반복 종료

입력된 URL을 SHA1 해싱

해시 앞 3글자 파싱

현재 디렉토리를 ~/.cache로 이동

앞 3글자를 이름으로 디렉토리 생성

나머지 37글자 파싱

현재 디렉토리를 앞 3글자로 디렉토리로 이동

파일이 이미 있는지 확인

나머지 37글자를 이름으로 파일 생성

반복 종료

0 반환

## 결과 화면

### 1. mkdir()

발생한 이유:

```
int mkdir(const char *pathname, mode_t mode);
```

에서 생성되는 디렉토리의 실제 권한은  $\text{mode} \& \sim \text{umask} \& 0777$ 로 계산된다.

여기서 0777은 디렉토리가 가질 수 있는 최대 권한을 의미한다.

umask는 mkdir, open, creat, mknod와 같은 파일 생성 함수에 적용되며, 지정된 권한에서 특정 비트를 제거하는 역할을 한다.

시스템의 기본 umask 값은 환경 변수로 설정되며, 리눅스에서 일반적으로 022이기 때문에, 주어진 코드와 같이 umask를 따로 설정하지 않고 mkdir을 실행하면 그 기본 umask 값에 따라 디렉토리의 권한이 결정된다.

따라서 문제를 해결하기 위해 umask를 0으로 설정한 후에 mkdir를 호출하면 생성된 디렉토리는 인자로 제공된 권한을 정확히 따르게 된다.

문제 해결한 소스 코드:

A terminal window with a dark background and light green text. The title bar shows a window icon and the text 'kw2023202070@ubuntu: ~'. The code is as follows:

```
#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);
}
```

## 2. proxy #1-1

```
kw2023202070@ubuntu:~/assignment1-1$ ls
Makefile proxy_cache.c
kw2023202070@ubuntu:~/assignment1-1$ make
gcc proxy_cache.c -o proxy_cache -lcrypto
kw2023202070@ubuntu:~/assignment1-1$ ls
Makefile proxy_cache proxy_cache.c
kw2023202070@ubuntu:~/assignment1-1$ ./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
kw2023202070@ubuntu:~/assignment1-1$ tree ~/cache/
/home/kw2023202070/cache/
├── 3e7
│   └── 9fd210fb8e00c8114ff978d282258ed8a48ea
├── 48b
│   └── 99f68b208b5453b391cb0c6c3d6a9824f3c3a
├── e05
│   └── 0f293fe62e97369e4b716bb3e78fababf8f90
└── 7c4
    └── 818da7395e30442b1dcf45c9b6669d1c0ff6b

4 directories, 4 files
kw2023202070@ubuntu:~/assignment1-1$
```

\$ ls: Makefile과 소스 코드(proxy\_cache.c)를 작성하였다.

\$ make: make를 실행하여 컴파일을 수행하였고, gcc 컴파일 명령어가 정상적으로 실행되었다.

\$ ls: 컴파일 결과, 실행 파일 proxy\_cache가 생성된 것을 확인했다.

\$ ./proxy\_cache: 실행 파일 실행 결과, 표준 입력으로 여러 개의 url들을 입력하였고, "bye"를 입력하여 프로그램을 종료했다.

\$ tree ~/cache/: ~/cache 구조 확인 결과, SHA1 해시된 url의 앞 3글자를 이름으로 하여 디렉토리가 생성되었다. 그 디렉토리의 하위에는 나머지 37글자 이름으로 파일이 생성되었다.

## 고찰

이번 과제를 수행하며 먼저 mkdir 함수를 사용한 소스 코드에서 권한이 원하는 대로 설정이 되지 않는 문제를 살펴보았다. 시스템프로그래밍 이론 수업에서 umask를 배울 때, 특정 권한을 제거하는 함수인데 예제 코드에서는 umask(0)를 작성하여 아무 권한도 삭제하지 않겠다는 뜻을 명시하는 줄 알고 교수님께 질문한 적이 있었다. 과제를 통해 다시 한 번 공부해보니 umask는 기본 설정된 값이 있기 때문에 꼭 umask(0)를 작성해야 mkdir의 인자로 주어진 권한을 그대로 부여하여 디렉토리를 생성할 수 있다는 새롭고도 중요한 사실을 알게 되었다.

또한 이번에 우분투 코딩을 하면서 c언어는 너무 오래 전에 배웠었고, c++이나 js, java 등의 언어를 더욱 많이 배운 상황이었기 때문에 처음에는 string 관련 함수 등 c 문법에 대해 기억을 되찾느라 힘들었지만, 점차 익숙해지며 다양한 언어를 사용해볼 수 있게 되어서 뿌듯했다. 우분투에서 vi를 사용해서 코드를 작성하는 것과 Makefile을 이용해서 컴파일하는 부분도 지난 Basic 과제에서부터 연습해온 덕에 조금 더 익숙해진 듯했다. 코드를 구현하며 이론 수업에서 배운 파일 생성 함수들을 직접 사용하는 경험도 시스템프로그래밍을 이해하는 데에 많은 도움이 되었다.

## Reference

<https://www.it-note.kr/205>: mkdir 사용법 관련하여 참고하였다.