

2022년 1학기 시스템프로그래밍

# Proxy #1-1

**System Software Laboratory**  
College of Software and Convergence  
Kwangwoon Univ.

# Assignment 1-1 Contents

- 1. mkdir()
- 2. proxy #1-1
- Report Requirements

# 1. mkdir( )

- **mkdir() 이용하여 디렉터리 생성 시 발생하는 권한 문제**
  - 아래 코드를 사용 시 directory의 권한이 mkdir의 인자로 준 것과 일치 하지 않음
  - **발생한 이유와 문제를 해결한 소스코드 캡처하여 보고서에 작성**
  - Hint : umask

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
void main(int argc, char *argv[])
{
    if(argc < 2){
        printf("error\n");
        return;
    }
    mkdir(argv[1], S_IRWXU | S_IRWXG | S_IRWXO);
}
```

## 2. proxy #1-1 Requirements

- Input url은 웹사이트 명으로 입력
  - e.x) www.kw.ac.kr
- Bye command를 입력으로 받았을 때만 프로그램 종료
- 제공된 sha1\_hash 함수를 사용하여 입력 받은 URL을 Hashed URL로 변환
- 제공된 getHomedir 함수를 사용하여 Home directory path를 얻고, Home directory에 cache 디렉토리를 생성
- 디렉토리 이름은 hashed\_url의 앞 세 글자로 함
  - (e.g. Hahsed\_url : e000f293fe62e97369e4b716bb3e78fababf8f90
  - 디렉토리 이름 : e00
  - 파일이름 : 0f293fe62e97369e4b716bb3e78fababf8f90
- 파일 생성시 다양한 함수 사용가능

## 2) proxy #1-1 -Create Cache Directory and File

### ■ To do list

- url을 SHA-1으로 hashing
  - SHA-1를 이용한 hashing 함수를 사용
  - SHA-1 library를 설치해 주어야 함.
    - **\$ sudo apt-get install libssl-dev**
  - SHA-1 library를 사용한 코드 컴파일 시, 옵션이 필요함.
    - **-lcrypto**
    - e.g. gcc proxy\_cache.c -lcrypto
- Hashed된 URL에 해당하는 directory와 file 생성
  - 루트 디렉토리는 "~/cache/" 로.
  - Hashed URL에 해당하는 위치에 file이 있는지 검사하고, 없을 경우에 file 생성

## 2) proxy #1-1 -Create Cache Directory and File

### ■ Hashing Function using SHA-1

- 아래의 빈 칸을 채워서 사용

```
#include <stdio.h>           // sprintf()
#include <string.h>          // strcpy()
#include <openssl/sha.h>     // SHA1()

char *sha1_hash(char *input_url, char *hashed_url) {
    unsigned char hashed_160bits[20];
    char hashed_hex[41];
    int i;

    SHA1(  );

    for(i=0; i<sizeof(hashed_160bits); i++)
        sprintf(hashed_hex + i*2, "%02x", hashed_160bits[i]);

    strcpy(  );

    return hashed_url;
}
```

- Function
  - unsigned char \***SHA1** (const unsigned char \***d**, unsigned long **n**, unsigned char \***md**);
    - **d** : Hashing 할 데이터
    - **n** : Hashing 할 데이터의 길이
    - **md** : 해시 된 데이터를 저장할 배열

## 2) proxy #1-1 -Create Cache Directory and File

### ■ Hashing Function using SHA-1

- 아래의 빈 칸을 채워서 사용

```
#include <stdio.h>           // sprintf()
#include <string.h>          // strcpy()
#include <openssl/sha.h>     // SHA1()

char *sha1_hash(char *input_url, char *hashed_url) {
    unsigned char hashed_160bits[20];
    char hashed_hex[41];
    int i;

    SHA1(  );

    for(i=0;i<sizeof(hashed_160bits);i++)
        sprintf(hashed_hex + i*2, "%02x", hashed_160bits[i]);

    strcpy(  );

    return hashed_url;
}
```

- Variables
  - **input\_url** : 원본 URL
  - **hashed\_url** : 16진수로 변환된 hashed URL
  - **hashed\_160bits** : 해시 된 160bits의 값. (160bits = 5 bits \* 32 blocks)
  - **hashed\_hex** : hashed\_160bits를 1byte씩 16진수로 표현하고, 이를 문자열로 변환한 것

## 2) proxy #1-1 -Get Path of Home Directory

- Home directory path가 필요한 경우, 아래의 함수를 정의하여 사용

```
#include <sys/types.h>
#include <unistd.h>
#include <pwd.h>
#include <string.h>

char *getHomeDir(char *home) {
    struct passwd *usr_info = getpwuid(getuid());
    strcpy(home, usr_info->pw_dir);

    return home;
}
```

- mkdir(), opendir() 함수 등에서 인자로 “~/”와 같이 ~를 입력하여 home directory에 접근하는 것이 불가능 함.
- 이러한 경우, 절대 경로(e.g. /home/sslabs/)를 사용하지 말고,  
위의 함수를 이용하여 home directory path를 얻을 것.
- 본 함수 구현에 대한 자세한 내용은 Proxy 1-3 강의자료에서 다룰 예정임



## 2) proxy #1-1 Requirements - Input

- Read URL from the STDIN (**as a shell**)
- Use '**bye**' command to exit a program
- e.g.

```
sslab@ubuntu:~/sslab$ ls
Makefile proxy_cache.c
sslab@ubuntu:~/sslab$ make
gcc -o proxy_cache proxy_cache.c -lcrypto
sslab@ubuntu:~/sslab$ ls
Makefile proxy_cache proxy_cache.c
sslab@ubuntu:~/sslab$ ./proxy_cache
input url> www.kw.ac.kr
input url> www.google.com
input url> bye
sslab@ubuntu:~/sslab$
```

## 2) proxy #1-1 Requirements - Output

- Create cache directory

- HIT인 경우. (Assignment#1-2에서 구현)
  - 아무 것도 하지 않음.

- MISS인 경우

- 입력 받은 URL을 SHA-1 모듈을 사용하여 Hashed URL로 변환
- Hashed URL에 따라 directory와 cached file을 구성함.

- Cache directories

- Using hash function results
- The root directory is "~/cache/"
- 생성한 모든 Directory는 모든 권한을 갖도록 구현함.
  - ex) 8진수로 777 할당

\* tree 명령어는 추가 package 설치 후 사용 가능  
\$ sudo apt-get install tree

```
sslslab@ubuntu:~/sslslab$ ./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
sslslab@ubuntu:~/sslslab$ tree ~/cache/
/home/sslslab/cache/
├── 3ef
│   └── 9fd210fb8e00c8114ff978d282258ed8a48ea
├── 88b
│   └── 99f68b208b5453b391cb0c6c3d6a9824f3c3a
├── e90
│   └── 0f293fe62e97369e4b716bb3e78fababf8f90
└── fec
    └── 818da7395e30442b1dcf45c9b6669d1c0ff6b

4 directories, 4 files
```

## 2) proxy #1-1 Requirements

- Create Cache Directory and File

- Hashed된 URL에 해당하는 directory와 file 생성
  - e.g.

```
sslab@ubuntu:~/sslab$ ./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
```

```
/home/sslab/cache/
├── 3ef
│   └── 9fd210fb8e00c8114ff978d282258ed8a48ea
├── d8b
│   └── 99f68b208b5453b391cb0c6c3d6a9824f3c3a
├── e00
│   └── 0f293fe62e97369e4b716bb3e78fababf8f90
└── fed
    └── 818da7395e30442b1dcf45c9b6669d1c0ff6b
```

URL		Hashed URL
www.kw.ac.kr		e00/0f293fe62e97369e4b716bb3e78fababf8f90
www.google.com	➔	d8b/99f68b208b5453b391cb0c6c3d6a9824f3c3a
www.naver.com		fed/818da7395e30442b1dcf45c9b6669d1c0ff6b
klas.kw.ac.kr		3ef/9fd210fb8e00c8114ff978d282258ed8a48ea

# Report Requirements

- **Ubuntu 16.04.5 Desktop 64bits 환경에서 채점**
- **보고서 구성**
  - **보고서 표지**
    - 수업 명, 과제 이름, 담당 교수님, 학번, 이름, 강의 시간 필히 명시
      - 과제 이름 → Proxy 1-1
      - 강의 시간 → 월요일, 수요일 인 경우 : 월요일로 표기  
→ 화요일, 목요일 인 경우 : 화요일로 표기
  - 아래의 내용은 보고서에 필히 포함
    - Introduction
      - 과제 소개 – 4줄 이상(background 제외) 작성
    - Flow Chart
      - 코드 작성 순서도
      - sha1\_hash 함수, getHomedir 함수 필수 작성 아님
      - 강의자료 appendix 내용 참고
    - Pseudo code
      - 알고리즘
      - Sha1\_hash 함수, getHomedir 함수 필수 작성 아님
      - 강의자료 appendix 내용 참고
    - 결과화면
      - 수행한 내용을 캡처 및 설명
    - 고찰
      - 과제를 수행하면서 느낀점 작성
    - Reference
      - 과제를 수행하면서 참고한 내용을 구체적으로 기록
      - 강의자료만 이용한 경우 생략 가능

# Report Requirements

- **소프트카피만 작성**
- **제출 파일**
  - 보고서 이름:
  - 보고서 pdf로 변환하여 제출
    - 과제명(1-1)\_학번\_수업시간.pdf
    - 수업시간 월요일, 수요일 인 경우 → 1-1\_2022722000\_mon.pdf
    - 수업시간 화요일, 수요일 인 경우 → 1-1\_2022722000\_tue.pdf
  - C 파일 명:
    - proxy\_cache.c
    - Comment 작성(강의자료 Appendix 내용 참고)
  - Makefile:
    - 실행파일명: proxy\_cache
  - C 파일명, 실행파일명 지정한 이름 외 다른 명으로 작성 시 감점
- **위 파일들을 압축해서 제출 (파일명:과제명(1-1)\_학번\_수업시간.tar.gz)**
  - 월요일, 수요일 → 1-1\_2022722000\_mon..tar.gz
  - 화요일, 목요일 → 1-1\_2022722000\_tue.tar.gz
- **tar 압축 방법:**
  - 압축 시 : tar -zcvf [압축 파일명].tar.gz [폴더 명]
  - 해제 시 : tar -zxvf 파일명.tar.gz
- 컴파일은 무조건 Makefile(makefile)을 이용한 make로 함.
  - Makefile(makefile) 없거나 실행 불가시 0점
  - 파일 압축 오류 시, 0점 처리

# Report Requirements

- 과제 제출
    - On-line 제출
      - **KLAS - 강의 과제 제출**
      - **2022년 3월 30일 23:59:59 까지**
        - **딜레이 받지 않음**
          - 제출 마감 시간 내 미제출시 해당과제 **0점 처리**
        - 교내 서버 문제 발생시, 메일로 과제 제출 허용
  - 수업시간 외 과제 질문
    - 수업시간 외 과제 질문은 “강의 묻고 답하기” 게시판을 통해 진행
    - **과제 제출 마감날 전날까지 업로드 된 질문에만 답변**
  - 실습 수업을 수강하는 학생이 이론 과목에 과제 제출 시
    - 간단한 txt 파일로 제출
- Ex.) 실습수업 때 과제 제출했습니다.
- **이론 과목에 간단한 txt 파일 미 제출 시 감점**

2022년 1학기 시스템프로그래밍 실습

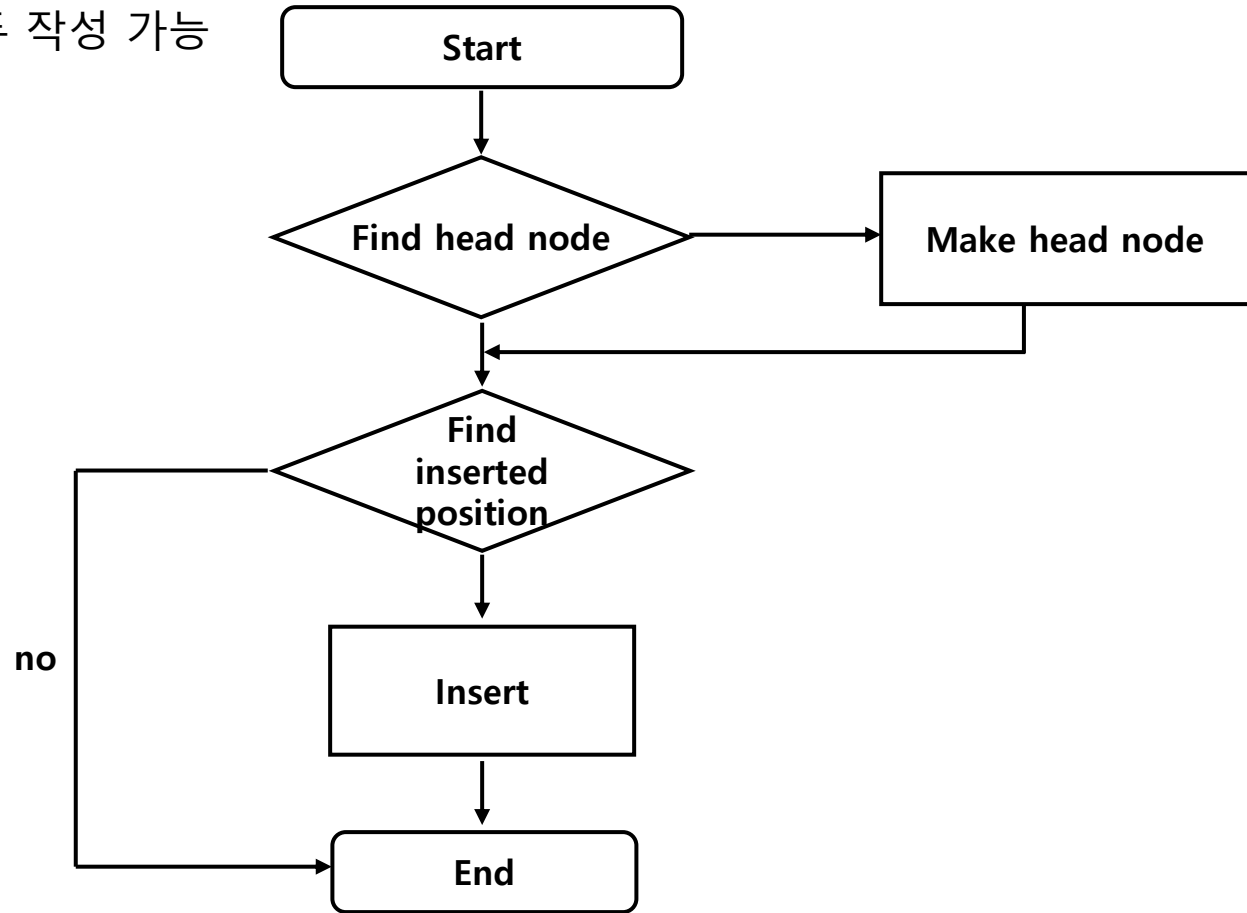
# Appendix

**System Software Laboratory**  
College of Software and Convergence  
Kwangwoon Univ.

# 보고서 작성 요령 (1/2)

## ■ Algorithm – Flow Chart (Each function)

- E.g.
- 국문, 영문 모두 작성 가능





# 보고서 작성 요령 (2/2)

## ■ Algorithm – Pseudo Code

- E.g.
- 국문, 영문 모두 작성 가능

```
FixHeap(Node *root, Key k)
{
    Node vacant, largerChild;
    vacant = root;
    while( vacant is not leaf ) {
        largerChild = the child of vacant with the larger key;
        if( k < largerChild's Key ) {
            copy largerChild's key to vacant;
            vacant = largerChild;
        }
        else exit loop;
    }
}
```

Set grade counter to one

While grade counter is less than or equal to ten

    Input the next grade

    Add the grade into the total

Set the class average to the total divided by ten

Print the class average

# Comment 작성 요령 (1/2)

## ■ File Head Comment

- E.g.
- 국문, 영문 모두 작성 가능

```
////////////////////////////////////  
// File Name      : Main.c                      //  
// Date           : 2022/03/01                  //  
// Os             : Ubuntu 16.04 LTS 64bits      //  
// Author         : Hong Gil Dong               //  
// Student ID     : 2022123456                  //  
// ----- //  
// Title : System Programming Assignment #1-1 (proxy server) //  
// Description : ...                            //  
////////////////////////////////////
```

# Comment 작성 요령 (2/2)

## ■ Function Head Comment

- E.g.
- 국문, 영문 모두 작성 가능

```
////////////////////////////////////  
// InsertNode                                                    //  
// =====                                                    //  
// Input: Node* -> Insert Node,                                  //  
//         Node* -> Column node before insert node            //  
//         Node* -> Row node before insert node                //  
//         (Input parameter Description)                        //  
// Output: int   - 1 success                                     //  
//           0 fail                                             //  
//         (Out parameter Description)                          //  
// Purpose: Inserting node                                       //  
////////////////////////////////////
```

# Comment 작성 요령 (3/3)

## ■ In-line Comment

- e.g.
- 국문, 영문 모두 작성 가능

```
if( pRowPos->pNextRow != pRowPos ) {  
    pTemp->pNextRow = pRowPos->pNextRow;           // pTemp set next row  
    if( !( pRowPos->pNextRow->bHead ) ){  
        pRowPos->pNextRow->NodeItem.pPrevRow = pTemp;  
    } // end of if  
} // end of if  
else {  
    pTemp->pNextRow = pRowPos;                       // pTemp set next row  
} // end of else  
pTemp->NodeItem.pPrevRow = pRowPos;                 // pTemp set previous row  
pRowPos->pNextRow = pTemp;
```