

2023년 1학기 시스템프로그래밍 & 시스템 프로그래밍 실습

# Assignment 2-2

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

## Assignment 2-2

- HTML\_Is의 결과를 다른 장치의 웹 브라우저에서 확인할 수 있도록 지원하는 서버 프로그램 작성
  - Steps
    - (1) (**web\_server**) Socket 생성
      - 각자 할당 받은 포트 번호 사용
    - (2) (**사용자**) 웹 브라우저로 사용자가 접속
      - ex) [http://223.194.46.163:port\\_number](http://223.194.46.163:port_number)
    - (3) (**web\_server**) 웹 브라우저가 보낸 HTTP request message를 처리
    - (4) (**web\_server**) HTML\_Is의 결과를 HTTP response message에 담아 웹 브라우저에 전송
    - (5) (**사용자**) 웹 브라우저에서 실행 결과를 확인

## Assignment 2-2 (cont'd)

- HTML\_Is의 결과를 다른 장치의 웹 브라우저에서 확인할 수 있도록 지원하는 서버 프로그램 작성
  - Code Requirements
    - -al 옵션 및 -l 옵션을 적용한 결과 출력
      - **Hint : getopt 함수에서 optind는 전역 변수이므로 재사용 시 리셋 필요**
      - ./web\_server의 실행위치를 root path(default directory)로 하고, root path인 경우 -l 옵션 적용
      - root path 밑의 **하위 디렉토리**인 경우 -al 옵션 적용
        - ex. 사용자 포트: 40000, ./web\_server 실행 위치가 /home/user/sp인 경우
          - [http:// 223.194.46.163: 40000](http://223.194.46.163:40000)로 접속 시, /home/user/sp의 ls -l결과를 출력
          - [http:// 223.194.46.163: 40000/tmp](http://223.194.46.163:40000/tmp)로 접속 시, /home/user/sp/tmp의 ls -al결과를 출력
      - 존재하지 않는 디렉토리의 URL 입력 시 **404 Error**
      - **포트번호는 40000 으로 고정 필수**
    - 추가적인 자세한 내용은 실행 예제를 참고
      - 실행 예제에 대한 것만 채점에 반영
      - 콘솔로 출력되는 내용은 채점에 반영하지 않음

## Assignment 2-2 (cont'd)

- HTML\_Is의 결과를 다른 장치의 웹브라우저에서 확인할 수 있도록 지원하는 서버 프로그램 작성
  - Parsing HTTP Request

### GET / HTTP/1.1

- Initial state of request
- List information of files **in default directory**

### GET /a.out HTTP/1.1

- (하이퍼링크가 클릭되었을 때) 해당 request가 binary 파일( 실행 파일 및 이미지 )을 요청하는 경우에는 다운로드 혹은 오픈 (예제 참고)

### GET /temp HTTP/1.1

- (하이퍼링크가 클릭되었을 때) Request of directory information
- List information of files **in specified directory**

## Assignment 2-2 (cont'd)

- 1. 하이퍼링크는 상대경로로만 생성
- 2. Server application을 실행한 디렉토리의 상위 디렉토리로 이동 불가
  - ex) Server application을 실행한 디렉토리의 path가 /home/sslslab/splab 이라면,
    - 다음의 디렉토리는 **이동 불가 (구현 시 고려하지 않도록 함)**
      - /home/sslslab/bonobono (\* 상위 디렉토리 내용물을 알 수 없으므로 이동 불가)
      - /home/sslslab
      - /home
      - /
      - ...
    - 다음의 디렉토리는 **이동 가능**
      - /home/sslslab/splab
      - /home/sslslab/splab/quiz3
      - ...

# Assignment 2-2 (cont'd)

## ■ HTTP Response Header 중..

### ■ Content-Type

- 콘텐츠의 종류(html, image, ...)를 MIME type(media type) 형태로 기입
- 본 과제에서는, **아래의 MIME type 및 target 만을 고려**
- 이미지의 경우 jpg, png, jpeg 확장자만 고려(대소문자 둘 다 가능)

MIME type	Target
text/html	Web page file (html)
text/plain	Normal text file, source code
image/*	Image file

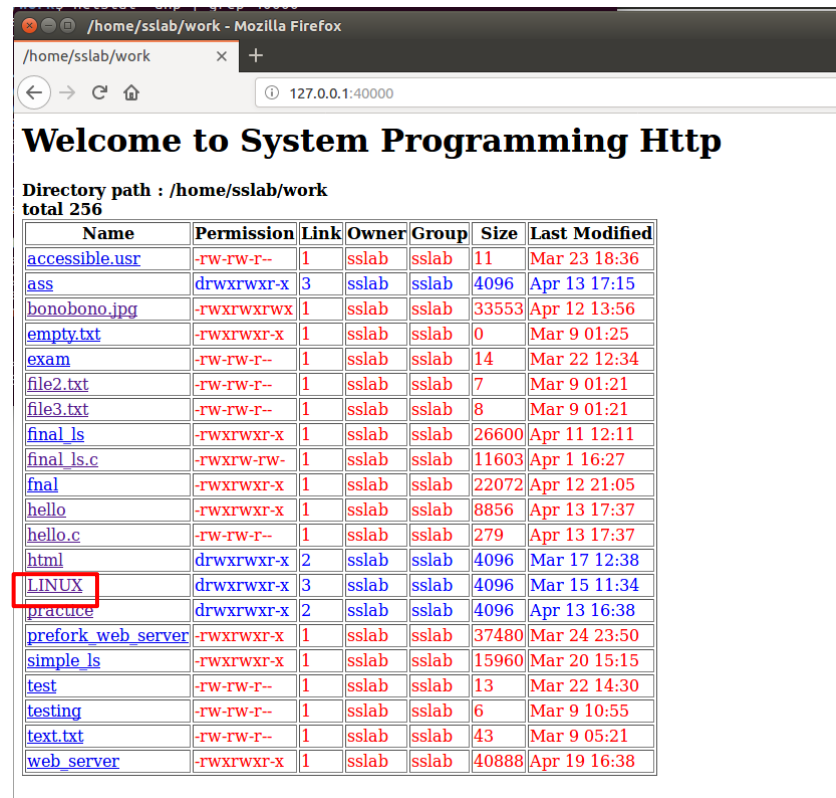
- Hint) \_GNU\_SOURCE를 #define 하면 fnmatch 사용 시 FNM\_CASEFOLD 옵션(대소문자무시) 추가 사용 가능
- 아래와 같은 코드 처리로 이미지 파일 처리

```
#define _GNU_SOURCE
if ( fnmatch("*.jpg", path, FNM_CASEFOLD) == 0 ||
     fnmatch("*.png", path, FNM_CASEFOLD) == 0 ||
     fnmatch("*.jpeg", path, FNM_CASEFOLD) == 0 )
```

# Assignment 2-2 (cont'd)

## ■ Root Path

- ./web\_server가 실행되는 root path에서는 -l 옵션의 결과 출력 및 상단에 “Welcome to System Programming Http” 출력
- Root Path의 부모 디렉토리로 이동할 수 없다고 가정
- Root Path의 하위 디렉토리 간에는 자유로운 이동이 가능해야 함



Directory path : /home/sslslab/work  
total 256


Name	Permission	Link	Owner	Group	Size	Last Modified
<a href="#">accessible.usr</a>	-rw-rw-r--	1	sslslab	sslslab	11	Mar 23 18:36
<a href="#">ass</a>	drwxrwxr-x	3	sslslab	sslslab	4096	Apr 13 17:15
<a href="#">bonobono.jpg</a>	-rwxrwxrwx	1	sslslab	sslslab	33553	Apr 12 13:56
<a href="#">empty.txt</a>	-rwxrwxr-x	1	sslslab	sslslab	0	Mar 9 01:25
<a href="#">exam</a>	-rw-rw-r--	1	sslslab	sslslab	14	Mar 22 12:34
<a href="#">file2.txt</a>	-rw-rw-r--	1	sslslab	sslslab	7	Mar 9 01:21
<a href="#">file3.txt</a>	-rw-rw-r--	1	sslslab	sslslab	8	Mar 9 01:21
<a href="#">final_ls</a>	-rwxrwxr-x	1	sslslab	sslslab	26600	Apr 11 12:11
<a href="#">final_ls.c</a>	-rwxrw-rw-	1	sslslab	sslslab	11603	Apr 1 16:27
<a href="#">fnal</a>	-rwxrwxr-x	1	sslslab	sslslab	22072	Apr 12 21:05
<a href="#">hello</a>	-rwxrwxr-x	1	sslslab	sslslab	8856	Apr 13 17:37
<a href="#">hello.c</a>	-rw-rw-r--	1	sslslab	sslslab	279	Apr 13 17:37
<a href="#">html</a>	drwxrwxr-x	2	sslslab	sslslab	4096	Mar 17 12:38
<a href="#">LINUX</a>	drwxrwxr-x	3	sslslab	sslslab	4096	Mar 15 11:34
<a href="#">practice</a>	drwxrwxr-x	2	sslslab	sslslab	4096	Apr 13 16:38
<a href="#">prefork_web_server</a>	-rwxrwxr-x	1	sslslab	sslslab	37480	Mar 24 23:50
<a href="#">simple_ls</a>	-rwxrwxr-x	1	sslslab	sslslab	15960	Mar 20 15:15
<a href="#">test</a>	-rw-rw-r--	1	sslslab	sslslab	13	Mar 22 14:30
<a href="#">testing</a>	-rw-rw-r--	1	sslslab	sslslab	6	Mar 9 10:55
<a href="#">text.txt</a>	-rw-rw-r--	1	sslslab	sslslab	43	Mar 9 05:21
<a href="#">web_server</a>	-rwxrwxr-x	1	sslslab	sslslab	40888	Apr 19 16:38

클릭 시 해당  
디렉토리로 이동

## Assignment 2-2 (cont'd)

### 하위 디렉토리 폴더로 이동시

- al 옵션의 결과를 출력하고 상단에 “System Programming Http” 출력
- (단. 폴더명에는 공백이 없다고 가정한다.)
- ex Untitled Folder (이런 경우는 고려하지 않음)



해당 디렉토리로 이동된 URL

### System Programming Http

Directory path : /home/sslslab/work/LINUX  
total 32

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	3	sslslab	sslslab	4096	Mar 15 11:34
..	drwxrwxr-x	6	sslslab	sslslab	4096	Apr 19 16:56
<a href="#">file_a</a>	-rw-rw-r--	2	sslslab	sslslab	33	Mar 9 06:50
<a href="#">file_b</a>	-rw-rw-r--	2	sslslab	sslslab	33	Mar 9 06:50
<a href="#">file_c</a>	-rw-rw-r--	1	sslslab	sslslab	16	Mar 9 06:52
<a href="#">file_d</a>	-rw-rw-r--	1	sslslab	sslslab	33	Mar 9 06:52
<a href="#">test</a>	drwxrwxr-x	2	sslslab	sslslab	4096	Mar 21 00:08
<a href="#">world.txt</a>	-rw-rw-r--	1	sslslab	sslslab	4	Mar 9 06:44



# Assignment 2-2 (cont'd)

- Text 파일 및 소스 코드 클릭 시
  - 파일의 내용을 display
  - 링크파일 클릭 시 원본 파일내용 display

```
127.0.0.1:40000/practice/cli x +
127.0.0.1:40000/practice/cli.c

#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define BUFFSIZE 1024
#define PORTNO 40000

int main() {
    int socket_fd, len;
    struct sockaddr_in server_addr;
    char haddr[] = "127.0.0.1";
    char buf[BUFFSIZE];

    if ((socket_fd = socket(PF_INET, SOCK_STREAM, 0)) < 0) {
        printf("can't create socket.\n");
        return -1;
    }

    memset(buf, 0, sizeof(buf));
    memset(&server_addr, 0, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = inet_addr(haddr);
    server_addr.sin_port = htons(PORTNO);

    if (connect(socket_fd, (struct sockaddr*)&server_addr, sizeof(server_addr)) < 0) {
        printf("can't connect.\n");
        return -1;
    }
}
```

Mozilla Firefox

127.0.0.1:40000/practice/IPSL x +

127.0.0.1:40000/practice/IPSL

this is text file WOW!

Mozilla Firefox

127.0.0.1:40000/html/example x +

127.0.0.1:40000/html/example\_link

```
<html>
<head>
<title> Our first document </title>
</head>
<body>
Hello World
</body>
</html>
```

/home/sslslab/work - Mozilla Firefox

/home/sslslab/work x +

127.0.0.1:40000/practice

## System Programming Http

Directory path : /home/sslslab/work/practice  
total 48

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	2	sslslab	sslslab	4096	Apr 19 17:03
..	drwxrwxr-x	6	sslslab	sslslab	4096	Apr 19 16:56
cli	-rwxrwxr-x	1	sslslab	sslslab	9128	Apr 13 16:38
cli.c	-rw-rw-r--	1	sslslab	sslslab	1148	Apr 12 03:48
IPSL	-rw-rw-r--	1	sslslab	sslslab	23	Apr 19 17:03
srv	-rwxrwxr-x	1	sslslab	sslslab	9280	Apr 13 16:38
srv.c	-rw-rw-r--	1	sslslab	sslslab	1602	Apr 12 03:42
srv_request.c	-rw-rw-r--	1	sslslab	sslslab	2716	Apr 12 03:43

/home/sslslab/work - Mozilla Firefox

/home/sslslab/work x +

127.0.0.1:40000/html

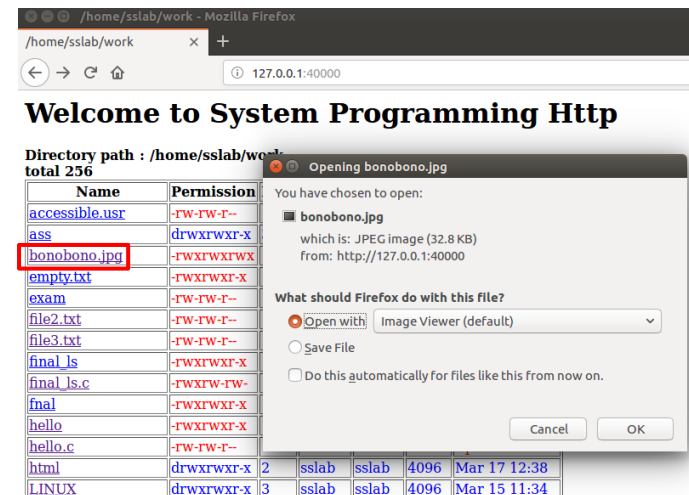
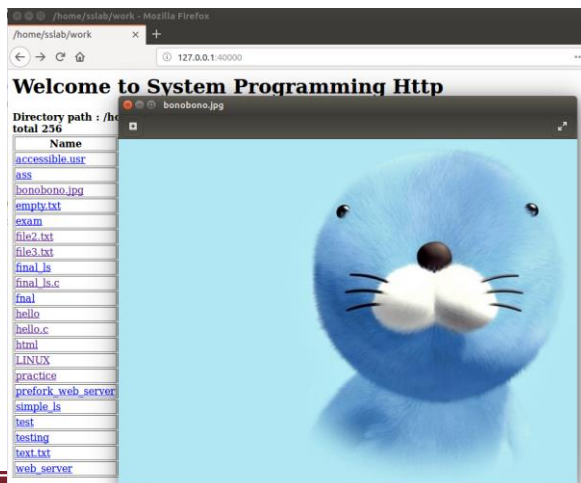
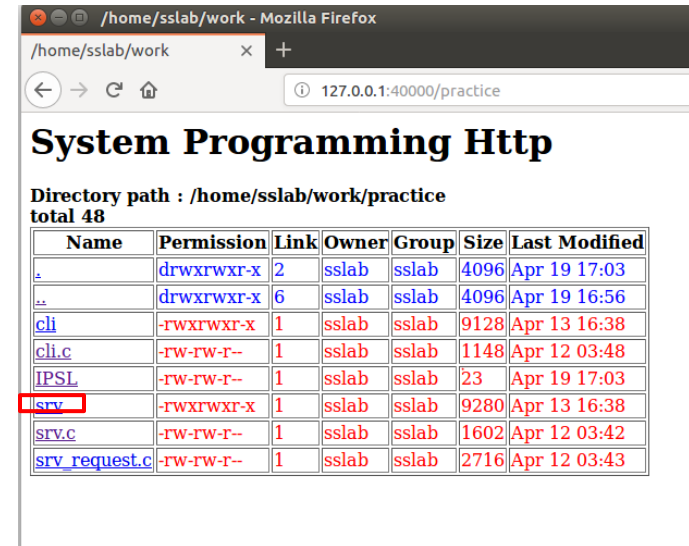
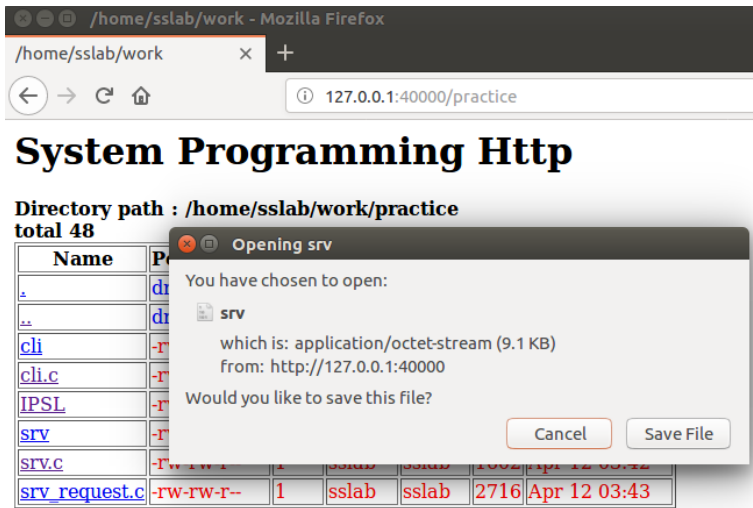
## System Programming Http

Directory path : /home/sslslab/work/html  
total 28

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	2	sslslab	sslslab	4096	Apr 19 17:42
..	drwxrwxr-x	6	sslslab	sslslab	4096	Apr 19 16:56
ex.html	-rw-rw-r--	1	sslslab	sslslab	92	Mar 17 12:38
example_link	lrwxrwxrwx	1	sslslab	sslslab	7	Apr 19 17:42
file1.txt	-rw-rw-r--	1	sslslab	sslslab	4	Mar 9 01:00
file2.txt	-rw-rw-r--	1	sslslab	sslslab	5	Mar 9 01:00
file3.txt	-rw-rw-r--	1	sslslab	sslslab	4	Mar 9 01:00
hello_copy.txt	-rw-rw-r--	1	sslslab	sslslab	5	Mar 9 01:00

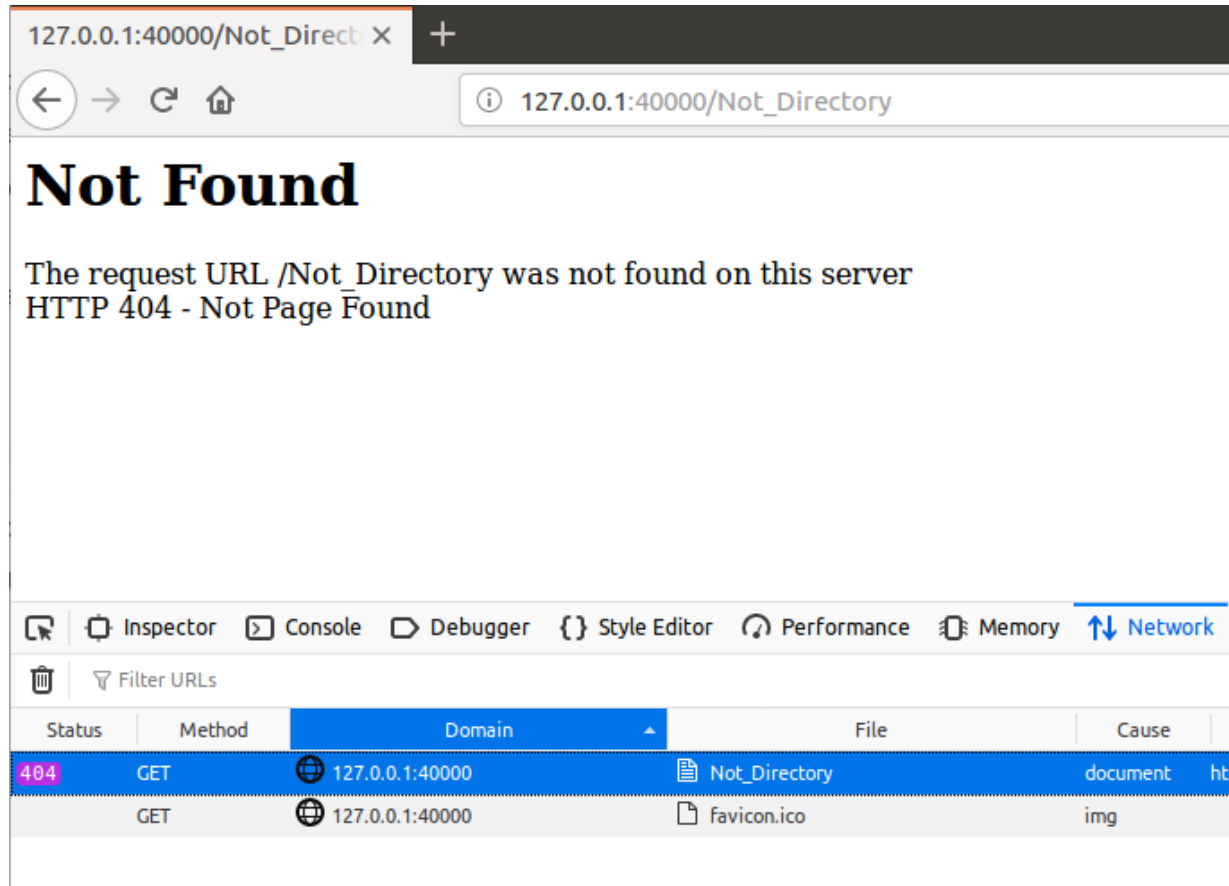
# Assignment 2-2 (cont'd)

- 실행 파일(binary file, image file) 클릭 시
  - 파일 Download가 정상적으로 되어야 함



## Assignment 2-2 (cont'd)

- 존재하지 않는 디렉토리 URL 입력 시



- 과제 진행 시 favicon.ico Request는 무시

# Report Requirements

- **Ubuntu 16.04.5 Desktop 64bits 환경에서 채점**
- **Copy 발견 시 0점 처리**
- **보고서 구성**
  - **보고서 표지**
    - 수업 명, 과제 이름, 담당 교수님, 학번, 이름 필히 명시
      - 과제 이름 → assignment2-2
  - **과제 내용**
    - Introduction
      - 과제 소개 - 4줄 이상(background 제외) 작성
    - Flow chart (appendix 참고)
    - Pseudo code (appendix 참고)
    - 결과화면
      - 수행한 내용을 캡처 및 설명
    - 고찰
      - 과제를 수행 결과 분석 및 해석 해결한 부분, 문제점 발생에 대한 요약
    - Reference
      - 과제를 수행하면서 참고한 내용을 구체적으로 기록
      - 강의자료만 이용한 경우 생략 가능

# Report Requirements

- Softcopy Upload

- 제출 파일

- 보고서 + 소스파일 하나의 압축 파일로 압축하여 제출(tar.gz)
    - 보고서(.pdf. 파일 변환)
    - 소스코드
      - 학번\_web\_server.c
      - Makefile
      - 실행 파일명 : web\_server
      - 실행 파일명 다르게 작성 시 감점

- 보고서 및 압축 파일 명 양식


- Web2\_2\_수강분류코드\_학번 으로 작성

수강요일	이론1 월5수6	이론2 월2수1	실습1 금12	실습2 금56	실습3 금78
수강분류 코드	A	B	C	D	E

- 예시-이론 월5 수6 수강하는 학생인경우
    - 보고서 Web2\_2\_A\_20231234567.pdf
    - 압축 파일 명: Web2\_2\_A\_20231234567.tar.gz

# Report Requirements

- 실습 수업을 수강하는 학생인 경우
  - 실습 과목에 과제를 제출(.tar.gz)
  - 이론 과목에 간단한 .txt 파일로 제출

 실습수업때제출했습니다.

2022-08-29 오후 3:58    텍스트 문서

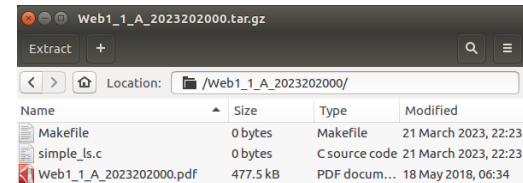
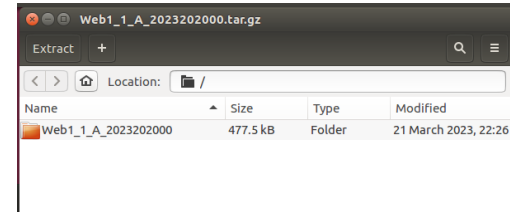
OKB

- 이론 과목에 .txt 파일 미 제출 시 감점
- 과제 제출
  - KLAS – 강의 과제 제출
  - 2023년 5월 3일 수요일 23:59까지 제출
    - 딜레이 받지 않음
      - 제출 마감 시간 내 미제출시 해당 과제 0점 처리
      - 교내 서버 문제 발생 시, 메일로 과제 제출 허용
- Tar 압축 및 해제 방법
  - 압축 시 → `tar -zcvf [압축 파일명].tar.gz[폴더 명]`
  - 해제 시 → `tar -zxvf 파일명.tar.gz`

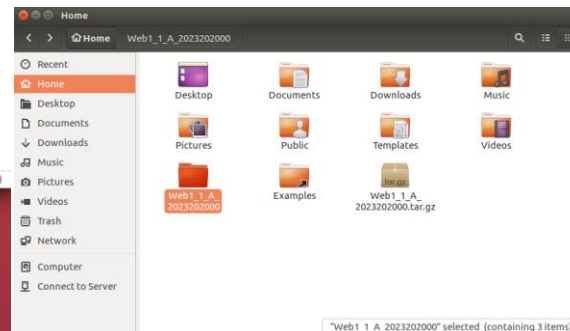
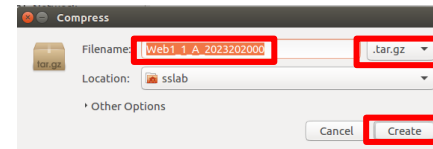
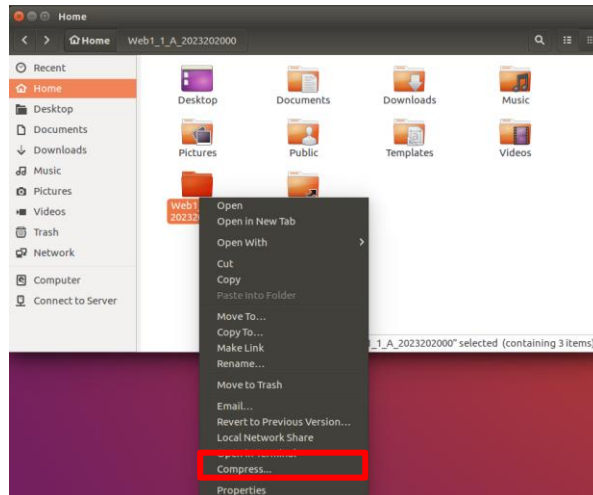
# 과제 제출 양식

- 소스파일, 보고서, Makefile만 하나의 tar.gz 로 압축, 아래 두 가지 방법 중 택 1
- 압축 명령어 사용

```
sslslab@ubuntu: ~/Web1_1_A_2023202000
sslslab@ubuntu:~$ ls
Desktop  Downloads      Music  Public  Videos
Documents examples.desktop Pictures Templates Web1_1_A_2023202000
sslslab@ubuntu:~$ cd Web1_1_A_2023202000/
sslslab@ubuntu:~/Web1_1_A_2023202000$ ls
Makefile simple_ls.c Web1_1_A_2023202000.pdf
sslslab@ubuntu:~/Web1_1_A_2023202000$
sslslab@ubuntu:~$ tar -zcvf Web1_1_A_2023202000.tar.gz Web1_1_A_2023202000/
Web1_1_A_2023202000/
Web1_1_A_2023202000/Makefile
Web1_1_A_2023202000/simple_ls.c
Web1_1_A_2023202000/Web1_1_A_2023202000.pdf
sslslab@ubuntu:~$ ls
Desktop  examples.desktop  Public  Web1_1_A_2023202000
Documents Music           Templates Web1_1_A_2023202000.tar.gz
Downloads Pictures         Videos
```



- 디렉토리에서 직접 압축



2023년 1학기 시스템프로그래밍실습

# Appendix

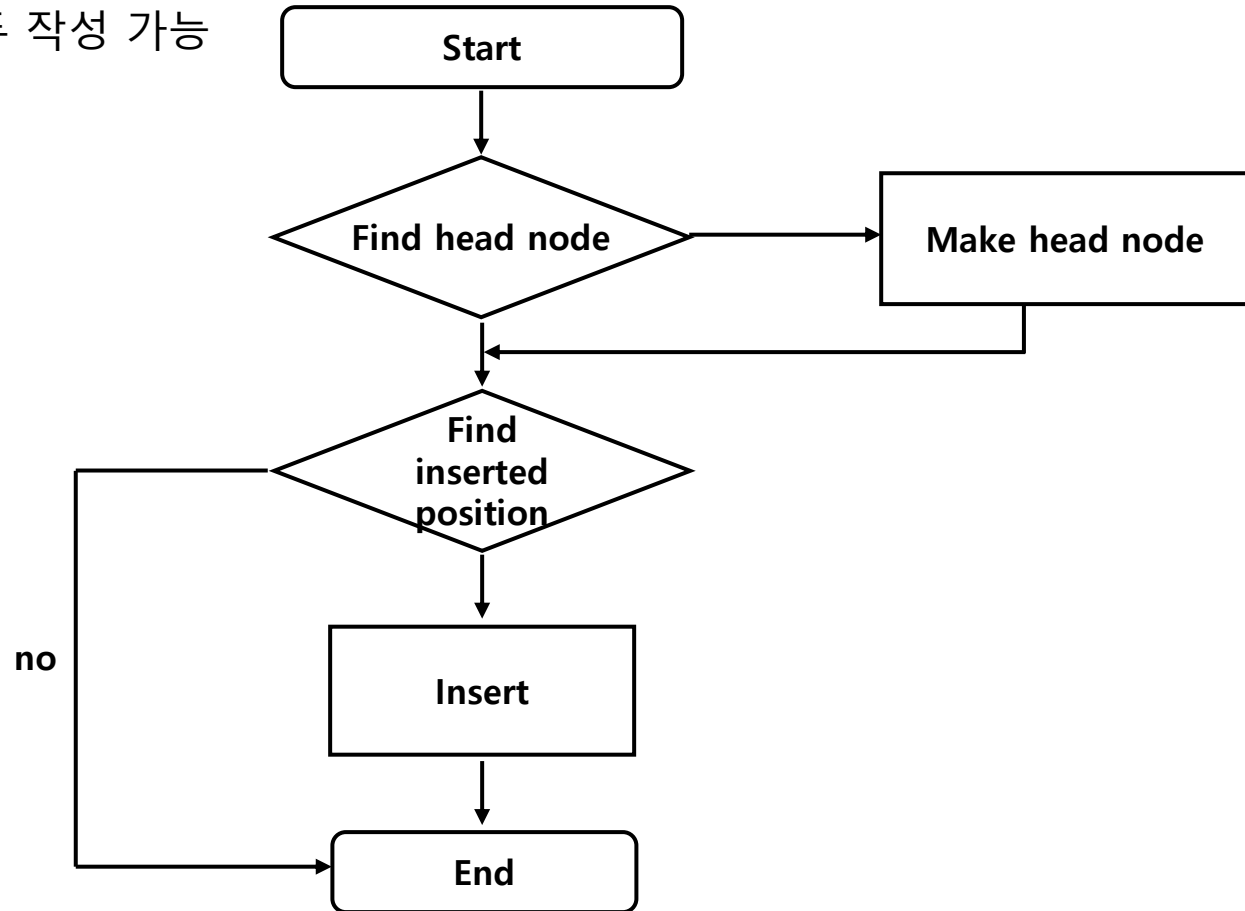
**System Software Laboratory**  
College of Software and Convergence  
Kwangwoon Univ.



# 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 (1/3)

## ■ File Head Comment

- E.g.
- 국문, 영문 모두 작성 가능

```
////////////////////////////////////  
// File Name      : Main.c                               //  
// Date           : 2023/03/01                           //  
// Os             : Ubuntu 16.04 LTS 64bits              //  
// Author         : Hong Gil Dong                       //  
// Student ID     : 2023123456                           //  
// ----- //   
// Title : System Programming Assignment #1-1 (proxy server) //  
// Description : ...                                     //  
////////////////////////////////////
```

# Comment 작성 요령 (2/3)

## ■ 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;
```