

# 시스템프로그래밍실습

## assignment 2-1

학 과: 컴퓨터정보공학부

담당교수: 이기훈 교수님

학 번: 2019202050

성 명: 이강현

# 1. Introduction

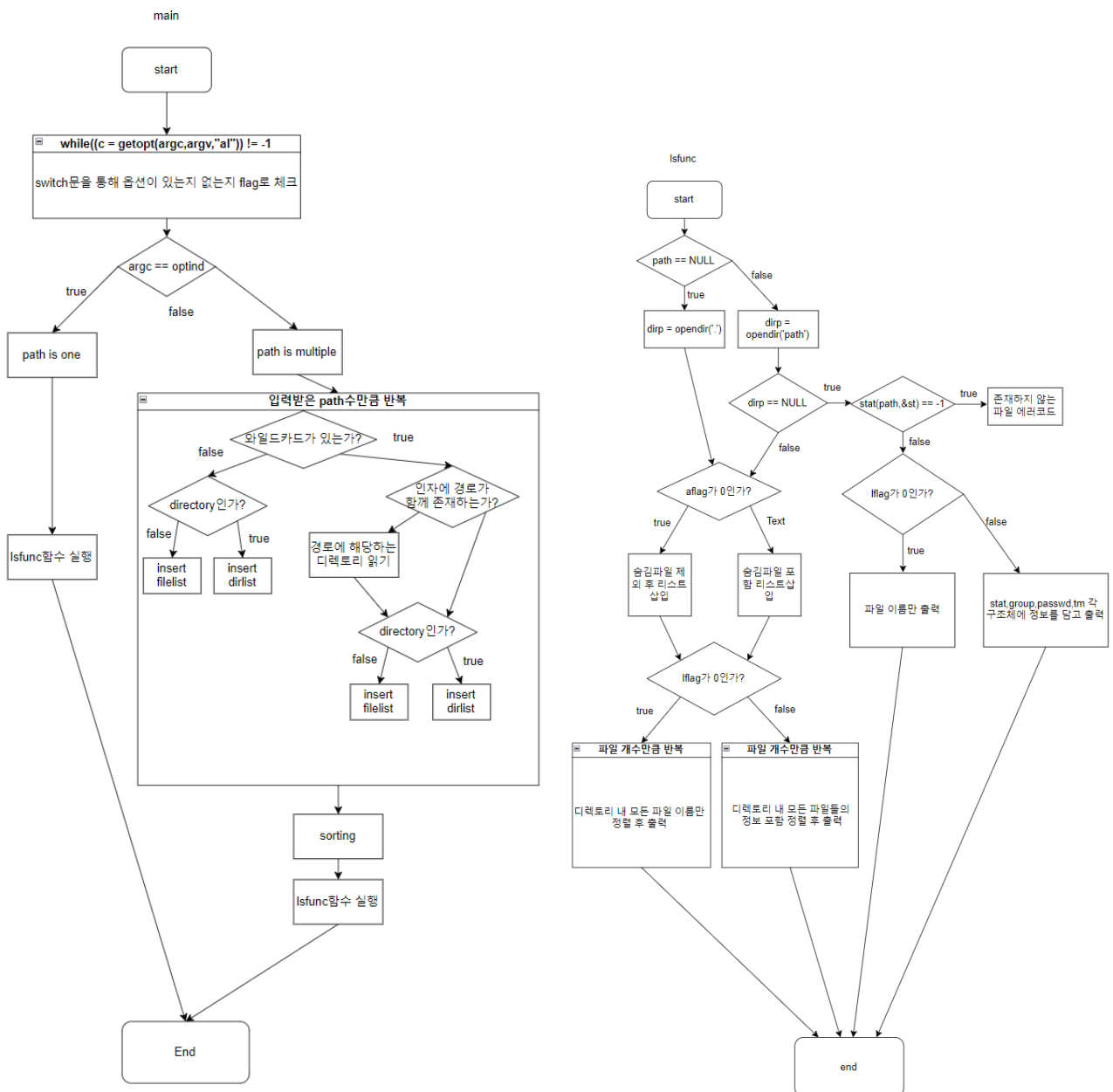
이번 실습은 html이 무엇인지 알고 html의 기본 문법을 익힌다.

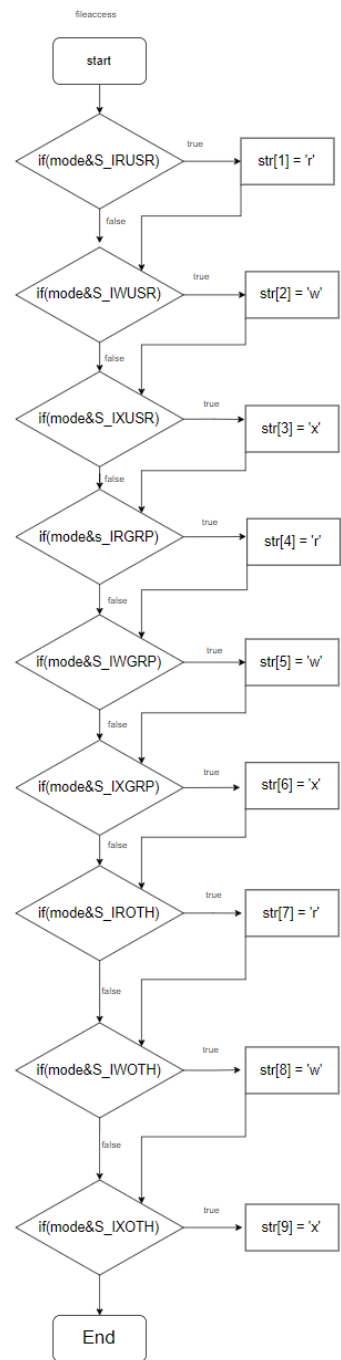
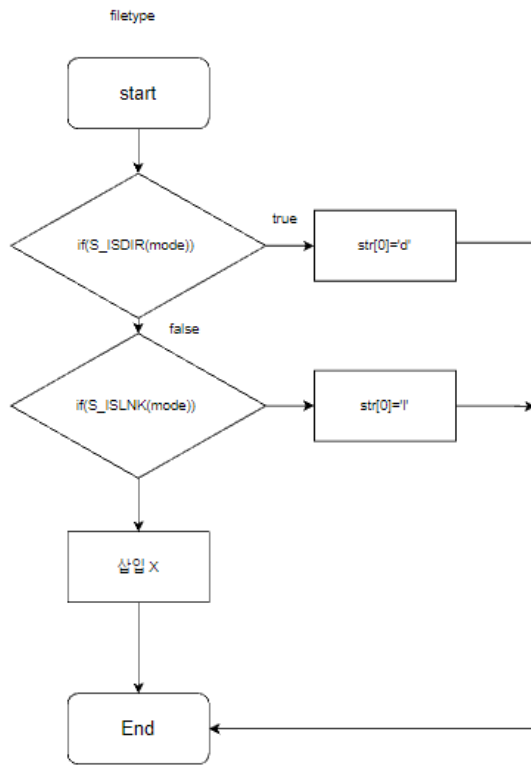
파일 스트림을 사용하는 법을 알고 이를 통해 html 파일을 생성하고 이전에 구현한 final\_ls의 결과를 볼 수 있게 한다.

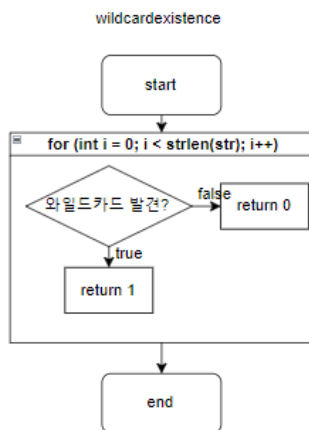
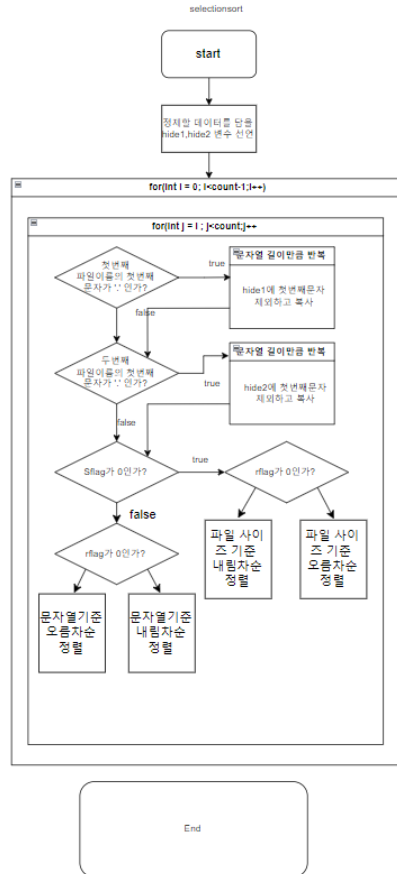
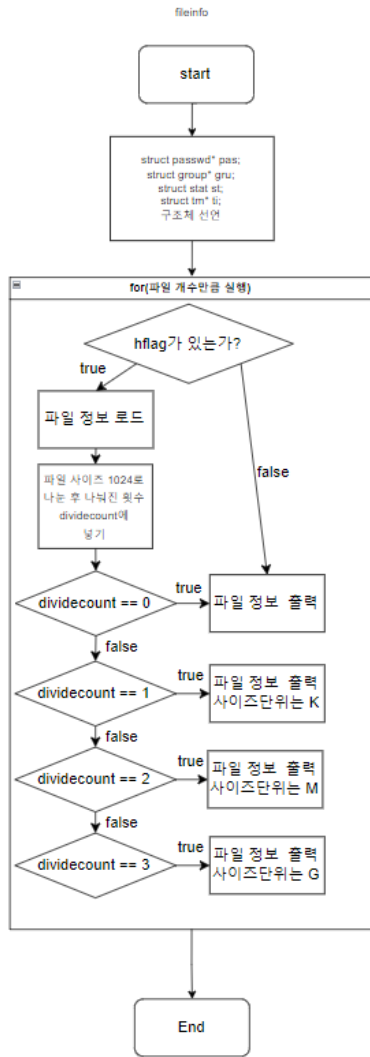
final\_ls의 파일 정보를 색깔로 구분하여 보면서 가독성을 높여본다.

하이퍼링크를 사용할 줄 알고 이를 통해 파일 혹은 디렉토리에 링크를 사용하여 본다.

# 2. Flow chart







모든 함수들의 흐름도이다. main 함수는 command 를 파싱하여 어떤 옵션이 들어왔는지 경로가 몇 개 들어왔는지를 체크하여 lsfunc 에 넘겨줄 인자와 lsfunc 이 실행되어야 하는 횟수를 결정한다.

이번 과제에서는 추가적으로 전역변수로 FILE 스트림을 선언하고 main에서 이를 열어 fprintf함수를 통해 html의 기본적인 틀을 형성한다.

lsfunc 함수는 main 함수에서 넘겨받은 인자에 따라 파일인지 폴더인지를 체크한 후 분기를 나누고 각각 옵션에 따라 요구받은 행동을 취한다.

lsfunc 함수내에서 실행되는 fileaccess함수와 filetype함수는 l옵션에서 제일 먼저 출력되는 파일 타입과 접근권한을 숫자에서 문자데이터로 변환해주는 함수이다. 이번 과제에서는 추가적으로 심볼릭링크에 대한 파일을 지원한다.

fileinfo 함수는 파일리스트와 파일 개수를 인자로 받아 파일 개수만큼 구조체에 파일정보를 저장하고 출력하는 것을 담당한다.

추가적으로 심볼릭 링크는 stat을 이용하면 심볼릭링크 자체의 정보를 가져올 수 없기 때문에 lstat을 사용하였다.

이번 과제에서는 html에서 table을 출력하는 부분을 담당한다.

selectionsort함수는 선택정렬을 구현한 함수로 숨김파일을 구분하지 않고 정렬하기 위해 임시변수를 활용하여 구현하였다.

전체적인 흐름은 final\_ls와 동일하나 웹에서 출력을 하는 부분에 추가적으로 fprintf를 사용하여 파일을 작성하는 작업을 한다.

### 3. Pseudo Code

```
selectionsort
파일 사이즈 변수선언
파일 경로 만들기 위한 변수선언
문자열 정제 후 저장할 변수 선언

만약 입력 경로가 NULL이 아니라면
    for(파일 개수만큼)
        경로 생성하여 파일 사이즈를 얻어내고 변수에 저장
for(파일개수 -1만큼)
    for(파일개수 -1만큼)
        만약 파일의 앞 문자가 '.'이라면
            앞문자 제외하여 변수에 저장
        아니라면
            그대로 저장

소문자 모두 대문자로 변경

if(!sflag)
    if(!rflag)
        문자열 기준 오름차순 정렬
    else
        문자열 기준 내림차순 정렬
else
    if(!rflag)
        파일사이즈 기준 내림차순 정렬
        사이즈가 같다면 문자열 기준 정렬
    else
        문자열 기준 오름차순 정렬
        사이즈가 같다면 문자열 기준 정렬
```

```
filetype
입력받은 파일의 mode가 디렉토리라면
    입력받은 문자열의 맨 앞 글자를 d로 변경
입력받은 파일의 mode가 심볼릭 링크라면
    입력받은 문자열의 맨 앞 글자를 l로 변경
|
fileaccess
입력받은 파일의 mode의 user access가 read가 가능하다면
    입력받은 문자열의 2번째를 r로 변경
입력받은 파일의 mode의 user access가 write가 가능하다면
    입력받은 문자열의 3번째를 w로 변경
입력받은 파일의 mode의 user access가 execution이 가능하다면
    입력받은 문자열의 4번째를 x로 변경

입력받은 파일의 mode의 group access가 read가 가능하다면
    입력받은 문자열의 5번째를 r로 변경
입력받은 파일의 mode의 group access가 write가 가능하다면
    입력받은 문자열의 6번째를 w로 변경
입력받은 파일의 mode의 group access가 execution이 가능하다면
    입력받은 문자열의 7번째를 x로 변경

입력받은 파일의 mode의 other access가 read가 가능하다면
    입력받은 문자열의 8번째를 r로 변경
입력받은 파일의 mode의 other access가 write가 가능하다면
    입력받은 문자열의 9번째를 w로 변경
입력받은 파일의 mode의 other access가 execution이 가능하다면
    입력받은 문자열의 10번째를 x로 변경
```

```
lsfunc
입력받은 경로로 디렉토리 열기
디렉토리가 null이라면
    if(!lflag)
        파일이름만 출력
    else
        파일 정보까지 출력(fileinfo함수)
if(!aflag)
    숨김파일 제외 파일 저장

else
    모두 읽기

if(!lflag)
    정렬 하고 파일들 이름만 출력
else
    정렬하고 파일들 정보까지 출력

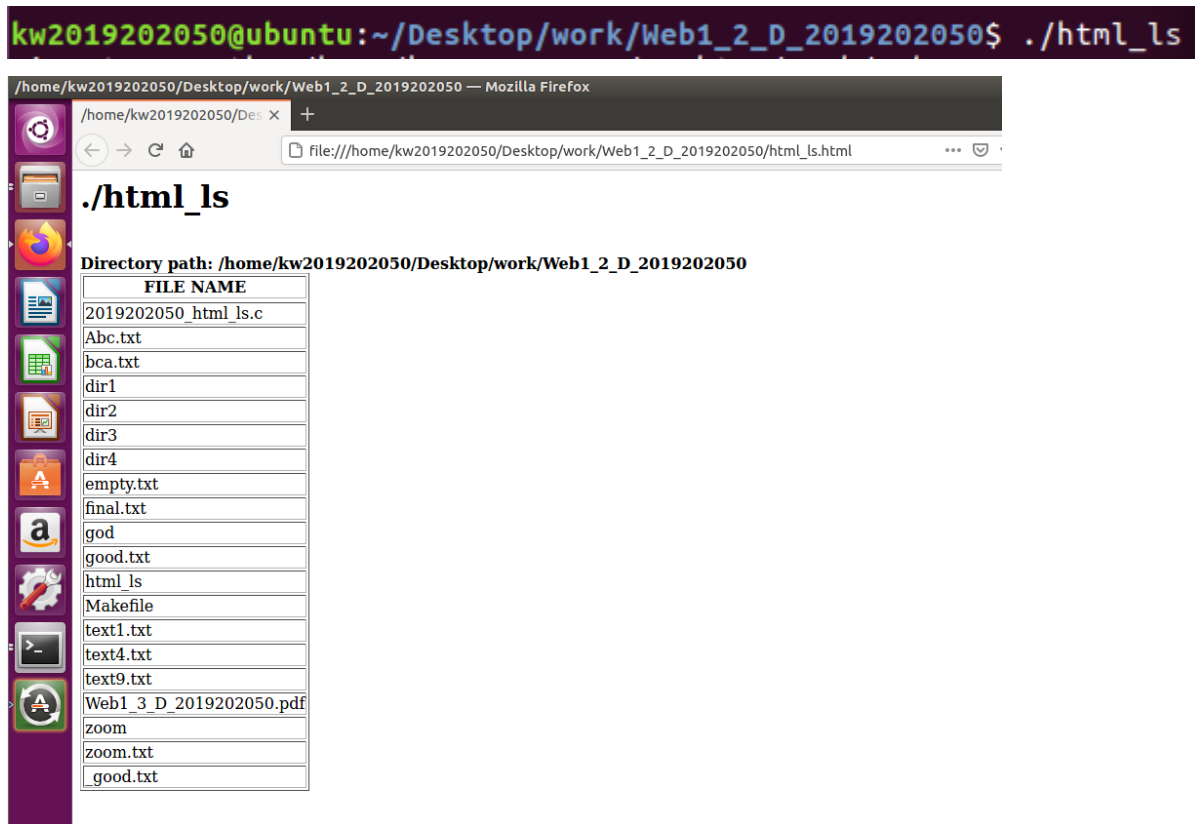
main
command parsing
if(입력받은 논옵션부분이 없다면)
    현재디렉토리를 경로로하여 lsfunc 호출
else
    for(논옵션 인자개수만큼 반복)
        if(wildcard가 있다면)
            if(wildcard와 경로가 함께 존재한다면)
                경로로 디렉토리를 열고 디렉토리내부 파일들중
                와일드카드패턴과 일치하는 파일들을
                디렉토리는 dirlist에 파일은 filelist에 저장
            else
                현재경로로 디렉토리를 열고 내부 파일들을
                와일드카드패턴과 일치하는 파일들을
                디렉토리는 dirlist에 파일은 filelist에 저장
        else
            파일은 filelist에 디렉토리는 dirlist에 저장

filelist와 dirlist 각각 정렬
출력
```

```
fileinfo
구조체 선언
입력받은 경로로 파일 정보 얻어와 구조체에 담기
파일 정보를 문자열로 변환
if(hflag)
    파일 정보중 파일 사이즈 부분을 나누어 단위를 붙여줌
파일 정보 출력

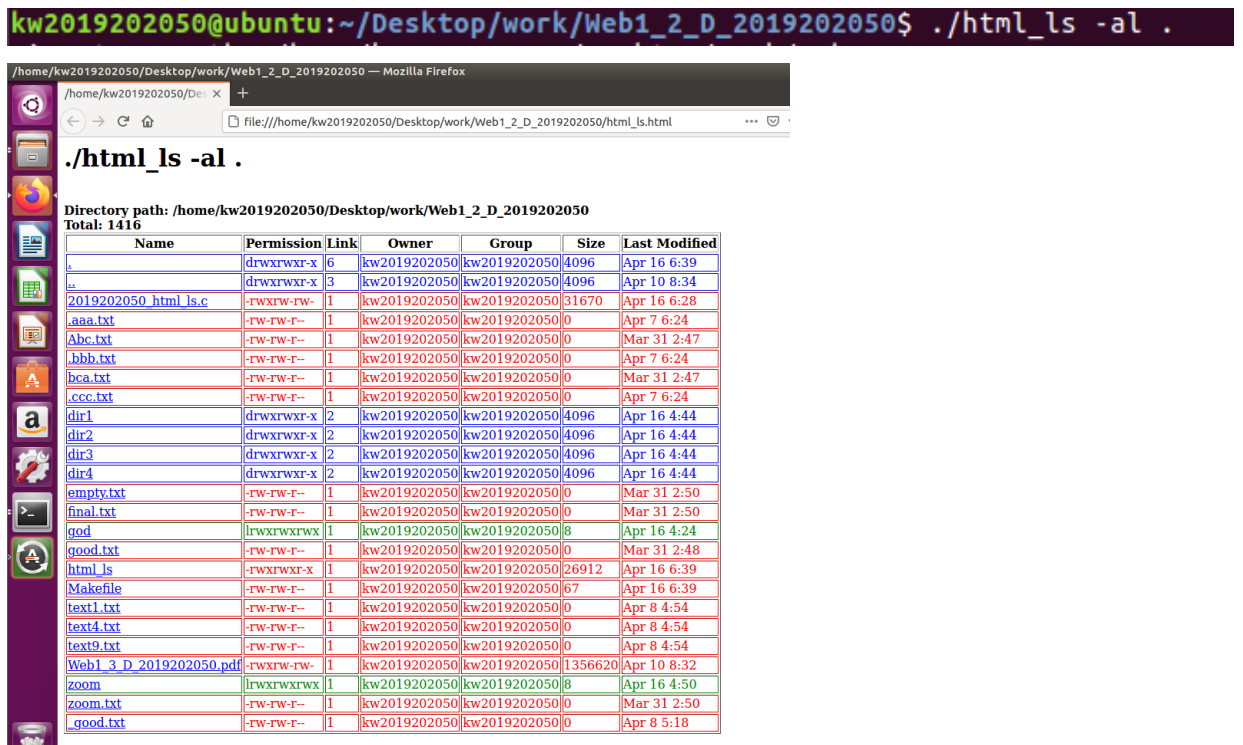
wildcardexistence
wildcard = ".*?[]"
if(와일드 카드를 입력받은 문자열에서 찾으면)
    return 1
못찾으면
    return 0
```

#### 4. 결과화면

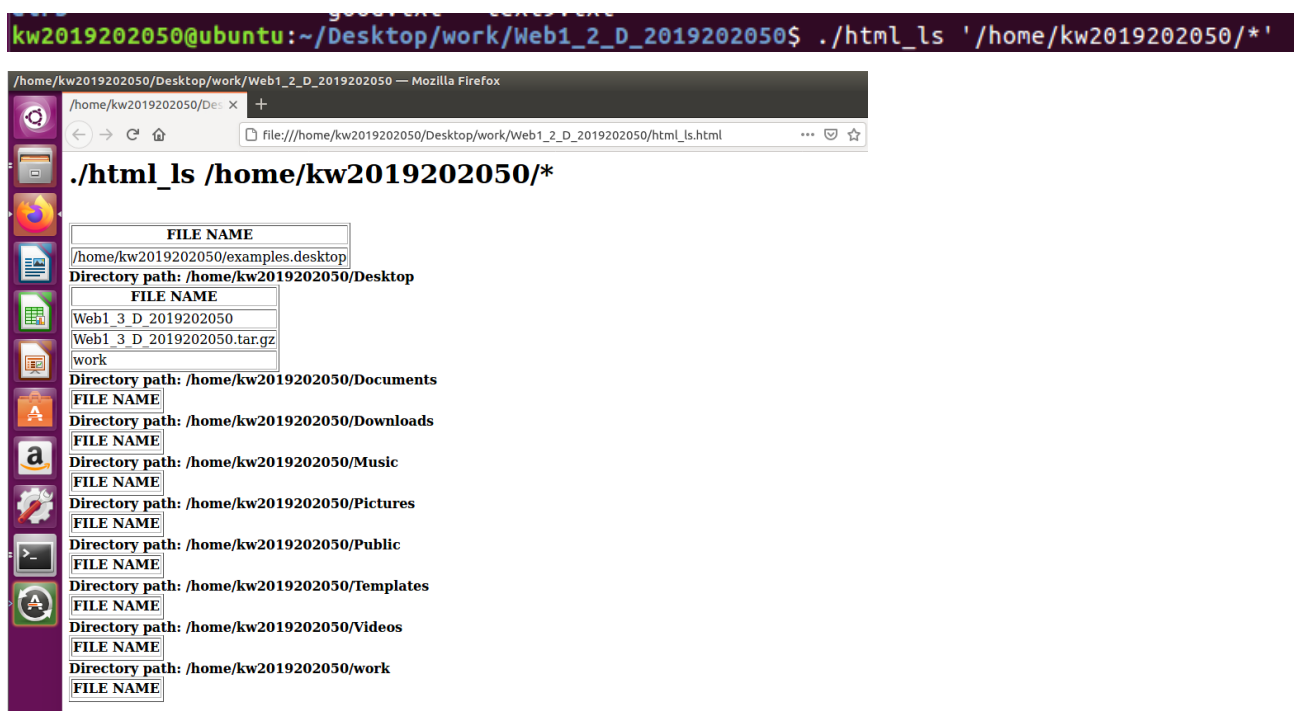


위의 명령어를 실행하고 생성된 `html_ls.html`을 열어본 결과이다. title은 현재 디렉토리로 출력되었다. 명령어가 위에 head부분으로 출력되었고 그 밑에는 아무 인자가 없었기에 현재 경로를 출력한다.

파일은 옵션이 없기 때문에 이름만 table형식으로 나열하여 출력하였다.



위는 -al 옵션을 주어 현재 디렉토리를 출력해본 것이다. 현재 디렉토리의 자세한 정보가 나타나 있고 파일들의 이름들은 하이퍼링크가 달려있다. -al 옵션에 맞게 정렬되어 있고 디렉토리일때는 파란색, 심볼릭링크일 경우에는 초록색, 그외의 파일들은 빨간색으로 구분되어있다.



다음은 와일드카드를 사용해보았다. 파일들을 먼저 출력하고 아래에 directory path와 함께 디렉토리내에 파일이 있다면 출력한다.



## 5. 고찰

이번 과제는 단순히 현재까지 구현했던 `final_ls`의 출력을 html화 하는 것이었다. 따라서 `final_ls`에서 `printf` 함수를 통해 쉘에 결과를 보여주었던 코드라인에 `fprintf` 함수를 통해 추가적으로 html에 결과내용을 써주기만 하면 되었다. 실습은 간단했지만 html의 기본 문법을 습득하는 것이 핵심 과제였던 것 같다. 이번 실습때는 symbolic link파일을 색깔로 구분하여 table에 나타내는 것이 있었는데 기존의 `stat` 함수는 파일에 연결된 링크들을 따라가 원본파일의 정보를 가져와 주었기 때문에 symbolic link의 경로를 통해 `stat` 함수를 호출하면 이 파일이 가리키고 있는 파일의 정보를 가져왔다. 따라서 파일의 mode를 출력해주는 부분이 잘 안됐었는데 `lstat`이라는 함수를 사용하여 이를 해결할 수 있었다.

## 6. Reference

시스템프로그래밍실습 강의자료 참조