

LAB04

시스템 프로그래밍_류은경Prof

20211115744_권구태

1. command 테스트 및 간략 설명

- 1) mkdir -p ./a/b/c
- 2) find .
- 3) dirname ./a/b/c
- 4) ls -R .

- 1) mkdir은 새로운 디렉토리를 만들어라는 말이고, -p는 부모 디렉터리가 없으면 자동으로 생성하고, 이미 존재하는 디렉터리에 대해 에러를 발생시키지 않는다. 해당 폴더에서 최종적으로는 c디렉토리를 만들려고 했는데, 현재폴더 밑에 a,b도 없었음. 그러나 -p 옵션으로 자동으로 생성함.

```
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/1번문제$ mkdir -p ./a/b/c
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/1번문제$ ls
a
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/1번문제$ mkdir ./a/b/c
mkdir: cannot create directory './a/b/c': File exists
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/1번문제$ mkdir -p ./a/b/c
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/1번문제$
```

- 2) find .
현재 폴더에 있는 디렉토리 파일들을, parent-child 관계를 포함하여 모두 출력

```
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/1번문제$ find .
.
./a
./a/b
./a/b/c
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/1번문제$
```

- 3) dirname ./a/b/c
가장 마지막에 쓴 c를 제외한 디렉토리 부분만 출력함.

```
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/1번문제$ dirname ./a/b/c
./a/b
```

- 4) ls -R .
현재 디렉토리(.)와 그 하위 디렉토리의 모든 파일 및 폴더를 재귀적으로(-R 옵션) 나열함.

```
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/1번문제$ ls -R .
.:
a

./a:
b

./a/b:
c

./a/b/c:
```

2. (mv01.c) mv 구현

[코드 분석]

```
if (stat(av[1], &buf) != 0) {
    // 바꾸고자 하는 파일이 없는 상태
    printf("%s이 존재하지 않습니다.\n", av[1]);
} else {
    // 바꾸고자 하는 파일이 존재하는 상태
    struct stat st;

    if (stat(av[2], &st) == 0) {
        // 바꾸고자 하는 이름의 파일이 있는 상태
        if (S_ISDIR(st.st_mode)) {
            // ./mv01 file1.txt ./a(존재)

            char result[1024]; // 충분한 크기의 배열 선언 (버퍼
//오버플로우 주의)
            result[0] = '\0'; // 문자열을 빈 문자열로 초기화

            strcat(result, av[2]);
            strcat(result, "/");
            strcat(result, av[1]);

            rename(av[1], result);
        } else {
            // ./mv01 file1.txt file2.txt (file2가 정의되어있는 상태)
            rename(av[1], av[2]);
        }
    } else
        // 바꾸고자 하는 이름의 파일이 없는 상태
        // ./mv01 file1.txt ./a/file2.txt 또는
        // ./mv01 file1.txt file2.txt (file2가 정의되어있지 않은 상태)

        if (count_char_occurrences(av[2], "/") <= 1) {
            printf("디렉토리가 존재하지 않습니다.\n");
        } else {
            if (rename(av[1], av[2]) == -1) {
                printf("디렉토리가 존재하지 않습니다.\n");
            }
        }
    }
}
```

```
}
```

가장 첫번째 if문은 `./mv01 file1.txt file2.txt` 에서 바꿈의 대상이 되는 `file1.txt` 파일 자체가 존재하는지를 판단.

| 존재한다면 `av[2]`에 온 값이 있는지 없는지 판단함.

|| 존재한다면 해당 값이 디렉토리인지 아닌지 판단. -> 디렉토리라면 `rename` 함수에 넣어야 하는데, `./a` 형태 그대로 넣으면 `a`를 파일로 인식하고 파일 이름으로 바꿈. 따라서 `./a/file1.txt` 형태로 붙여줘야 함. 새롭게 배열을 만들어준 후, 두 문자열을 합침. 이후 `rename`함.

|| 만약 디렉토리가 아니라면 그냥 `rename`을 사용하여 바꿔주면 된다.

| 만약 `av[2]`의 값이 없다면, 해당 문자열이 디렉토리인지 아닌지 판단해야 한다.

`av[2]`의 값이 없을 때 들어올 수 있는 값은(현재 `ls`에는 `file1.txt`, `a(dir)`이 있다고 가정)
`./a/file100.txt`

이는 바로 `rename` 함수에 넣으면 알아서 `a`를 폴더로 인식하고 교체함.

[추후 개선사항]

`av[2]`의 값이 없을 때 들어올 수 있는 값은 `./a/file100.txt` 외에도 아주 많다. `./b/c/v` 등 이중 삼중으로 존재하지 않는 명령어가 들어올 수도 있다. 개선 필요함. 그러나 현재 LAB에 있는 구현 사항은 모두 구현이 가능함.

1) 파일 이름 변경

```
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ls
file1.txt mv01 mv01.c 새틀게알게된.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ls
file1.txt mv01 mv01.c 새틀게알게된.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ls a
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ./mv01 ./a/file1.txt
잘못된 명령어입니다.
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ./mv01 file1.txt ./a/file2.txt
file2.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ls
mv01 mv01.c 새틀게알게된.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$
```

2) 파일 이동

```
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ touch file1.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ls
file1.txt mv01 mv01.c 새틀게알게된.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ls a
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ./mv01 file1.txt ./a
file1.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ls
a mv01 mv01.c 새틀게알게된.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$
```

3) 파일 이동 및 이름 변경

```
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ls
file2.txt mv01 mv01.c 새틀게알게된.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ touch file1.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ls
file1.txt file2.txt mv01 mv01.c 새틀게알게된.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ./mv01 file1.txt file2.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ls
file2.txt mv01 mv01.c 새틀게알게된.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ./mv01 file2.txt file1.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$ ls
file1.txt mv01 mv01.c 새틀게알게된.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의/시스템프로그래밍_류은경/src/HW/4주차/2번문제$
```

3번

[코드 설명]

```
if ((dir_ptr = opendir(dirname)) == NULL) {
    perror("opendir");
    return;
}

printf("%s:\n", dirname); // 현재 디렉토리 출력
```

do_ls의 파라미터로 전달된 **dirname**의 디렉토리를 열고, 안열어지면 에러표시, 열어지면 이름 출력

```
char **subdirs = NULL;
int subdir_count = 0;

while ((dirent_ptr = readdir(dir_ptr)) != NULL) {
    if (strcmp(dirent_ptr->d_name, ".") == 0 ||
        strcmp(dirent_ptr->d_name, "..") == 0)
        continue; // 현재 디렉토리(.)와 부모 디렉토리(..) 제외

    // 파일 및 디렉토리 전체 경로 생성
    char full_path[1024];
    snprintf(full_path, sizeof(full_path), "%s/%s", dirname,
        dirent_ptr->d_name);

    struct stat st;
    if (stat(full_path, &st) == 0) {
        if (S_ISDIR(st.st_mode)) {
            // 디렉토리면 리스트에 저장 후 나중에 재귀 호출
            subdirs = realloc(subdirs, sizeof(char*) * (subdir_count
+ 1));

            subdirs[subdir_count] = strdup(full_path);
            subdir_count++;
        }
    }

    printf("%s\n", dirent_ptr->d_name);
}
```

1. 이중포인터의 정확한 개념 - `**subdir -> *subdir[1], *subdir[2], *subdir[3], *subdir[4], ...`
2. `snprintf`의 사용방법

3. 현재 디렉토리 아래에 데이터에는 부모 디렉터리와 자기자신을 가리키는 폴더가 존재함. 따라서 의도적으로 그런 것이 아니라면, 자기 자신 폴더와 부모 폴더를 가리키지 않도록 조심.
4. **realloc** 사용법 및 사용처 - 포인터나 포인터 배열 등, 값이 지속적으로 변화하는 곳에 사용할 수 있음.
5. **strdup** 함수 사용법 - 문자열 배열에 문자열을 추가하는 함수