

시스템 프로그래밍

2차 프로그래밍 과제



SINCE 1947

서경대학교

SEOKYEONG UNIVERSITY

SEOKYEONG UNIVERSITY

서경대학교

# 시스템 프로그래밍 2차

-2019301005 김경민-

# 목차

1.코드 & 설명

2.사진



# 1. 코드 & 설명

```
#include <sys/types.h>
#include <sys/stat.h>
#include <dirent.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <time.h>
#include <pwd.h>
#include <grp.h>

char type(mode_t);
int sortsize(struct dirent **d1, struct dirent **d2);
int reverse(struct dirent **d1, struct dirent **d2);
void reDirent(char *dir);
void sumPath(char *path, char* p1, char* p2);
void print(DIR *dp, struct dirent *d,int flag);
void printl(DIR *dp, struct dirent *d);
void prints(char *dir,int size,int flag);
void printr(char *dir);
```

기본적인 헤더 파일에 문자열, 시간 그리고 UID 와 PID 를 출력하기 위한 헤더파일을 추가

type 함수는 mode\_t 에 따라서 디렉토리인지 일반 파일인지 구분

sortsize 함수는 prints 함수에 쓰일 scandir 의 크기 비교 함수

reverse 함수는 printr 함수에 쓰일 scandir 의 역순 함수

reDirent 함수는 ls -R 에 쓰일 재귀함수

sumPath 함수는 재귀함수에 쓰일 path 를 생성하는 함수

print 함수는 일반적인 출력과 -F 시 출력에 쓰이는 함수

printl 함수는 long 형태의 출력에 쓰이는 함수

prints 함수는 -s 시 출력에 쓰이는 함수

printr 함수는 -r 시 출력에 쓰이는 함수

```

int main(int argc, char*argv[]){
    DIR *dp;
    struct dirent *d;
    struct stat buf;
    char path[BUFSIZ];
    char *dir;
    char *home;
    int flag=0;
    home=".";
    dp=opendir(".");
    if(argc ==1){
        print(dp,d,0);
    }
    else if(argc ==2){
        if(strcmp(argv[1], "-l")==0){
            printl(dp,d);
        }
        else if(strcmp(argv[1], "-F")==0){
            print(dp,d,1);
        }
        else if(strcmp(argv[1], "-R")==0){
            reDirent(home);
        }
        else if(strcmp(argv[1], "-r")==0){
            printr(home);
        }
    }
    else if(argc==3){
        if(strcmp(argv[1], "-s")==0){
            prints(home,atoi(argv[2]),0);
        }
        else if(strcmp(argv[1], "-ls")==0){
            prints(home,atoi(argv[2]),1);
        }
        else if(strcmp(argv[1], "-d")==0){
            if(strcmp(argv[2], "-l")==0)
                print(dp,d,1);
        }
    }
    else{
        if(strcmp(argv[1], "-l")==0){
            if(strcmp(argv[2], "-s")==0)
                prints(home,atoi(argv[3]),1);
        }
    }

    closedir(dp);

    return 0;
}

```

### main 함수

인자의 개수가 1개이면 기본적인 출력

2개인 경우에는 argv[1]의 형태에 따라 출력

3개인 경우에는 argv[1]의 형태와 argv[2]의 형태에 따라 출력

그 외 경우는 인자가 4개인 한 경우 출력

```

char type(mode_t mode) {
    if(S_ISREG(mode))
        return('-');
    if(S_ISDIR(mode))
        return('d');
    if(S_ISCHR(mode))
        return('c');
    if(S_ISBLK(mode))
        return('b');
    if(S_ISLNK(mode))
        return('l');
    if(S_ISFIFO(mode))
        return('p');
    if(S_ISSOCK(mode))
        return('s');
}

void print(DIR *dp, struct dirent *d,int flag){
    struct stat buf;
    int cnt=0;
    while((d = readdir(dp))){
        if(strncmp(d->d_name, ".",1)==0){
            continue;
        }
        stat(d->d_name,&buf);

        if(flag==0){
            printf("%-8s ",d->d_name);
            cnt++;
            if(cnt==8){
                putchar('\n');
                cnt=0;
            }
        }
        else if(flag==1){
            if(type(buf.st_mode)=='d')
                printf("%s/ ",d->d_name);
            else
                printf("%s ",d->d_name);
            cnt++;
            if(cnt==8){
                putchar('\n');
                cnt=0;
            }
        }
    }
    putchar('\n');
}

```

**type 함수는 mode 를 인자로 받아서 각각 S\_IS\_\_ 매크로에 넣어서 참이면 true 를 반환 true 인 조건에서 각각 맞는 return 값을 가짐**

**print 함수는 일반적인 출력 함수 디렉토리 , dirent 구조체 , 임의로 지정한 flag 를 인자로 받음 stat 구조체 buf 생성, cnt 를 0 으로 초기화 후 readdir 로 디렉토리를 읽음**  
**strncmp 함수로 d\_name 에서 앞에서 1 개의 문자를 비교하여 숨겨진 파일들은 continue**  
**buf 에 d\_name 인 파일의 정보를 넣어준 후 flag==0 이면 기본 LS 명령 8 개파일씩 출력**  
**flag==1 이면 -F 의 출력 buf.st\_mode 로 mode 를 가져와 type 함수를 사용하여**  
**값이 'd' 인지 확인 후 같으면 디렉토리라고 판정하여 /를 붙여서 출력**

```

void sumPath(char *path, char* p1, char* p2){
    strcpy(path,p1);
    strcat(path,"/");
    strcat(path,p2);
}
void reDirent(char *dir){
    DIR* dp = opendir(dir);
    struct dirent *d;

    print(dp,d,0);
    rewinddir(dp);

    while((d=readdir(dp))!=NULL){
        struct stat buf;
        char path[BUFSIZ];
        if(strncmp(d->d_name,".",1)==0){
            continue;
        }

        sumPath(path, dir, d->d_name);

        lstat(path, &buf);

        if(S_ISDIR(buf.st_mode)){
            printf("%s:\n",path);
            reDirent(path);
        }
    }
    closedir(dp);
}

```

**reDirent 함수는 재귀를 통하여 하위 디렉토리의 내용을 출력함**  
**실행할 디렉토리 경로를 인자로 받아서 open 함**  
**초기에 먼저 디렉토리의 내용을 출력 후 rewinddir 로 오프셋 위치 초기화 readdir 로 읽음**  
**stat 구조체 buf 생성 경로를 담아줄 path 생성 숨긴 파일 continue**  
**sumPath 함수를 이용하여 현재 디렉토리 경로와 d\_name, "/"를 합침**  
**lstat 을 사용하여 buf 에 path 경로의 파일정보를 저장 디렉토리의 판별을 위해 lstat 사용**  
**S\_ISDIR 을 사용하여 디렉토리인지 판별 후 동일경우 path 경로 출력 후 재귀**

```

void printl(DIR *dp, struct dirent *d){
    struct stat buf;
    while((d = readdir(dp))!=NULL){
        if(strncmp(d->d_name, ".",1)==0){
            continue;
        }
        stat(d->d_name, &buf);
        printf("%c", type(buf.st_mode));
        printf("%c", buf.st_mode & S_IRUSR ? 'r' : '-');
        printf("%c", buf.st_mode & S_IWUSR ? 'w' : '-');
        printf("%c", buf.st_mode & S_IXUSR ? 'x' : '-');
        printf("%c", buf.st_mode & S_IRGRP ? 'r' : '-');
        printf("%c", buf.st_mode & S_IWGRP ? 'w' : '-');
        printf("%c", buf.st_mode & S_IXGRP ? 'x' : '-');
        printf("%c", buf.st_mode & S_IROTH ? 'r' : '-');
        printf("%c", buf.st_mode & S_IWOTH ? 'w' : '-');
        printf("%c", buf.st_mode & S_IXOTH ? 'x' : '-');
        printf("%ld ", buf.st_nlink);
        printf("%s ", getpwuid(buf.st_uid)->pw_name);
        printf("%s ", getgrgid(buf.st_gid)->gr_name);
        printf("%-6ld ", buf.st_size);
        int i = 0;
        while (ctime(&buf.st_mtime)[i] != '\n') {
            printf("%c", ctime(&buf.st_mtime)[i]);
            i++;
        }
        printf("\t%s\n", d->d_name);
    }
}

```

**printl** 함수는 디렉토리와 **dirent** 구조체를 인자로 받은 후 **readdir**, 숨김파일 **continue**  
**stat** 을 사용하여 **buf** 에 파일 정보를 넣은 후 각각 & 연산을 통해 조건에 따라 출력  
**uid** 와 **gid** 는 따로 **getpw**, **getgr** 함수를 사용해서 출력 포인터로 반환 ->로 출력  
**size** 는 6 칸을 가지며 왼쪽 정렬로 출력 **ld** 는 부호를 가지는 10 진수 정수  
**ctime** 은 시간 정보를 문자열로 변환 개행문자전까지 읽어서 출력  
 마지막에 **tab** 후 이름 출력

```

void prints(char *dir,int size,int flag){
    struct dirent **fileList = NULL;
    int num,cnt=0;
    struct stat buf;
    num = scandir(dir,&fileList,NULL,sortsize);
    for(int n=0;n<num;n++){
        if(!strcmp(fileList[n]->d_name,".",1)){
            continue;
        }
        stat(fileList[n]->d_name,&buf);
        if(flag==0){
            if(buf.st_size>size)
                printf("%-8s ",fileList[n]->d_name);
            cnt++;
            if(cnt==8){
                putchar('\n');
                cnt=0;
            }
        }
        else{
            if(buf.st_size>size){
                printf("%c", type(buf.st_mode));
                printf("%c", buf.st_mode & S_IRUSR ? 'r' : '-');
                printf("%c", buf.st_mode & S_IWUSR ? 'w' : '-');
                printf("%c", buf.st_mode & S_IXUSR ? 'x' : '-');
                printf("%c", buf.st_mode & S_IRGRP ? 'r' : '-');
                printf("%c", buf.st_mode & S_IWGRP ? 'w' : '-');
                printf("%c", buf.st_mode & S_IXGRP ? 'x' : '-');
                printf("%c", buf.st_mode & S_IROTH ? 'r' : '-');
                printf("%c", buf.st_mode & S_IWOTH ? 'w' : '-');
                printf("%c", buf.st_mode & S_IXOTH ? 'x' : '-');
                printf("%ld ", buf.st_nlink);
                printf("%s ", getpwuid(buf.st_uid)->pw_name);
                printf("%s ", getgrgid(buf.st_gid)->gr_name);
                printf("%-6ld ", buf.st_size);
                int i=0;
                while (ctime(&buf.st_mtime)[i] != '\n') {
                    printf("%c", ctime(&buf.st_mtime)[i]);
                    i++;
                }
                printf("\t%s\n", fileList[n]->d_name);
            }
        }
    }
}

int sortsize(struct dirent **d1, struct dirent **d2){
    struct stat buf1, buf2;
    stat((*d1)->d_name, &buf1);
    stat((*d2)->d_name, &buf2);
    if(buf1.st_size<=buf2.st_size){
        if(buf1.st_size==buf2.st_size){
            return 0;
        }
        else{return 1;}
    }
    else{return -1;}
}

```

prints 는 디렉토리, 원하는 크기, flag 를 인자로 받아 dirent 구조체 포인터의 포인터 fileList 생성  
 scandir 함수를 사용하여 dir 의 디렉토리에 있는 파일 및 디렉토리를 sortsize 함수를 사용하여  
 fileList 에 담겨진 후 fileList 를 반복문으로 fileList[]안에 있는 내용을 하나씩 참조  
 숨겨진 파일들은 continue 한 후에 flag 가 0 인경우 즉 -s 500 과 같이 -s 만 있는경우 출력  
 그 외의 경우는 -l 혹은 -ls 인 경우이기 때문에 세부사항까지 전부 출력  
 sortsize 함수는 비교할 두 dirent 구조체 포인터의 포인터를 인자로 받아서  
 각 stat 을 사용하여 buf1,2 에 담은 후 buf.st\_size 를 비교하여 return 합니다  
 Scndir 은 내부적으로 quick sort 를 사용하기 때문에 sortsize 에서 내림차순을 위한 return 값

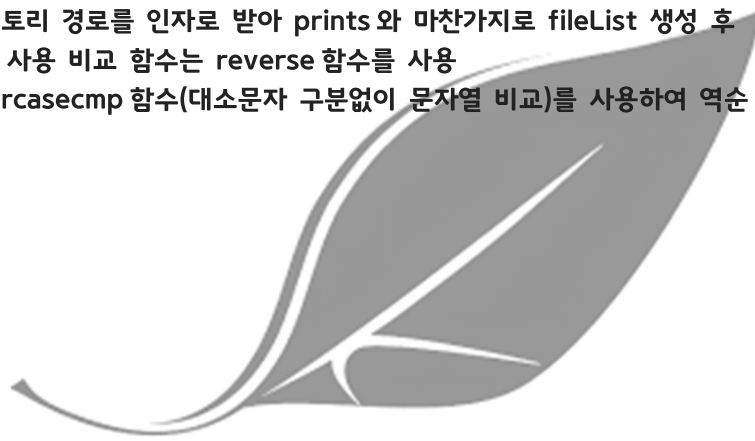


```

void printr(char *dir){
    struct dirent **fileList = NULL;
    int num,cnt=0;
    num = scandir(dir,&fileList,NULL,reverse);
    for(int i=0;i<num;i++){
        if(!strcmp(fileList[i]->d_name,".",1)){
            continue;
        }
        printf("%-8s ",fileList[i]->d_name);
        cnt++;
        if(cnt==8){
            putchar('\n');
            cnt=0;
        }
    }
    putchar('\n');
}
int reverse(struct dirent **d1, struct dirent **d2){
    return (strcasecmp((*d2)->d_name , (*d1)->d_name));
}

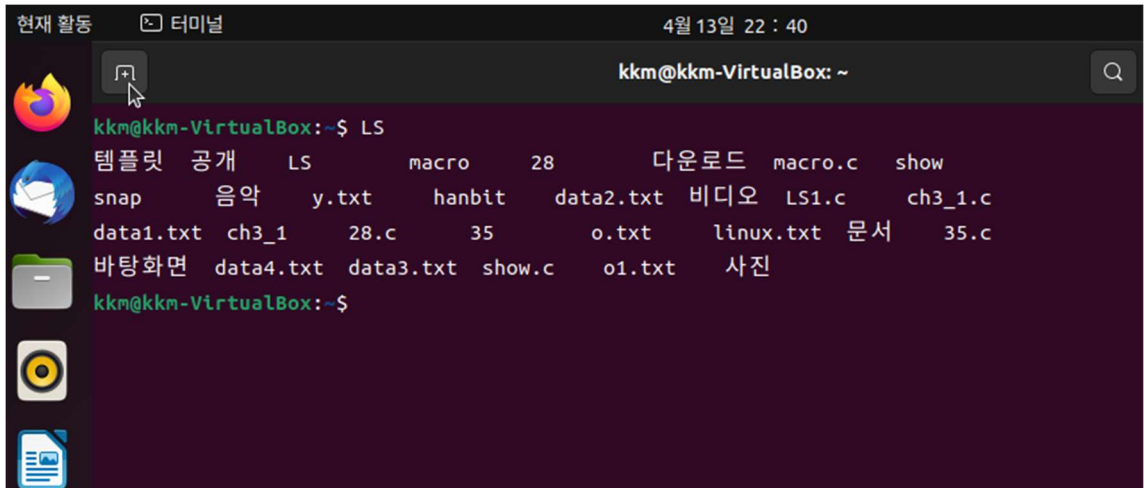
```

**printr 함수는 디렉토리 경로를 인자로 받아 prints와 마찬가지로 fileList 생성 후 scandir 를 함수를 사용 비교 함수는 reverse 함수를 사용 reverse 함수는 strcasecmp 함수(대소문자 구분없이 문자열 비교)를 사용하여 역순 정렬**



## 2. 사진

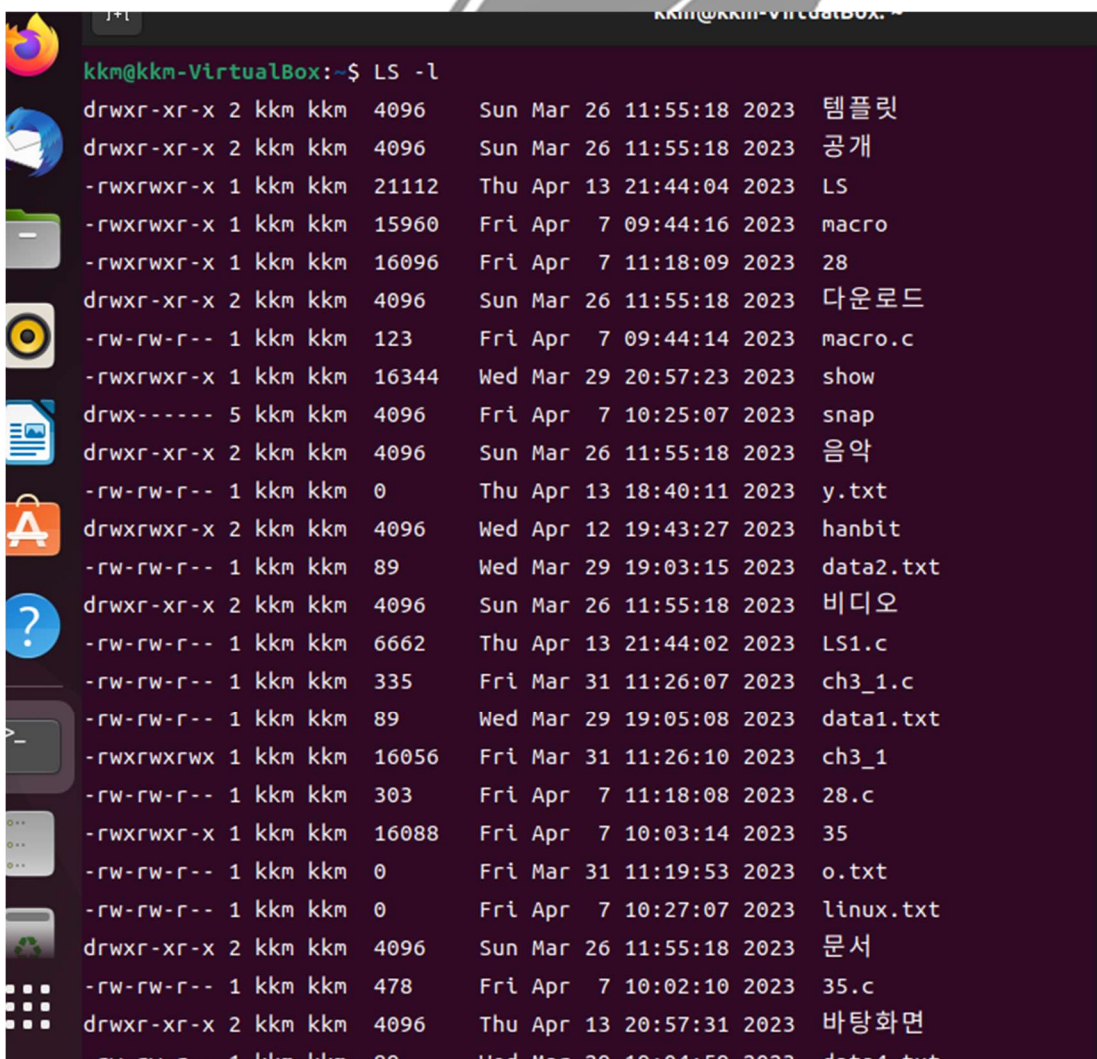
### 2-1. LS



A terminal window titled "현재 활동 터미널" with the date and time "4월 13일 22:40". The user is logged in as "kkm" on a machine named "kkm@kkm-VirtualBox". The prompt is "kkm@kkm-VirtualBox: ~". The user has entered the command "ls". The output lists files and directories in a grid-like format:

```
kkm@kkm-VirtualBox:~$ ls
템플릿  공개    LS      macro   28      다운로드 macro.c  show
snap    음악    y.txt   hanbit  data2.txt 비디오  LS1.c    ch3_1.c
data1.txt ch3_1    28.c    35      o.txt    linux.txt 문서     35.c
바탕화면 data4.txt data3.txt show.c   o1.txt   사진
kkm@kkm-VirtualBox:~$
```

### 2-2. -l



A terminal window titled "현재 활동 터미널" with the date and time "4월 13일 22:40". The user is logged in as "kkm" on a machine named "kkm@kkm-VirtualBox". The prompt is "kkm@kkm-VirtualBox: ~". The user has entered the command "ls -l". The output shows detailed file information including permissions, size, owner, group, date, and filename:

```
kkm@kkm-VirtualBox:~$ ls -l
drwxr-xr-x 2 kkm kkm 4096 Sun Mar 26 11:55:18 2023 템플릿
drwxr-xr-x 2 kkm kkm 4096 Sun Mar 26 11:55:18 2023 공개
-rwxrwxr-x 1 kkm kkm 21112 Thu Apr 13 21:44:04 2023 LS
-rwxrwxr-x 1 kkm kkm 15960 Fri Apr 7 09:44:16 2023 macro
-rwxrwxr-x 1 kkm kkm 16096 Fri Apr 7 11:18:09 2023 28
drwxr-xr-x 2 kkm kkm 4096 Sun Mar 26 11:55:18 2023 다운로드
-rw-rw-r-- 1 kkm kkm 123 Fri Apr 7 09:44:14 2023 macro.c
-rwxrwxr-x 1 kkm kkm 16344 Wed Mar 29 20:57:23 2023 show
drwx----- 5 kkm kkm 4096 Fri Apr 7 10:25:07 2023 snap
drwxr-xr-x 2 kkm kkm 4096 Sun Mar 26 11:55:18 2023 음악
-rw-rw-r-- 1 kkm kkm 0 Thu Apr 13 18:40:11 2023 y.txt
drwxrwxr-x 2 kkm kkm 4096 Wed Apr 12 19:43:27 2023 hanbit
-rw-rw-r-- 1 kkm kkm 89 Wed Mar 29 19:03:15 2023 data2.txt
drwxr-xr-x 2 kkm kkm 4096 Sun Mar 26 11:55:18 2023 비디오
-rw-rw-r-- 1 kkm kkm 6662 Thu Apr 13 21:44:02 2023 LS1.c
-rw-rw-r-- 1 kkm kkm 335 Fri Mar 31 11:26:07 2023 ch3_1.c
-rw-rw-r-- 1 kkm kkm 89 Wed Mar 29 19:05:08 2023 data1.txt
-rwxrwxrwx 1 kkm kkm 16056 Fri Mar 31 11:26:10 2023 ch3_1
-rw-rw-r-- 1 kkm kkm 303 Fri Apr 7 11:18:08 2023 28.c
-rwxrwxr-x 1 kkm kkm 16088 Fri Apr 7 10:03:14 2023 35
-rw-rw-r-- 1 kkm kkm 0 Fri Mar 31 11:19:53 2023 o.txt
-rw-rw-r-- 1 kkm kkm 0 Fri Apr 7 10:27:07 2023 linux.txt
drwxr-xr-x 2 kkm kkm 4096 Sun Mar 26 11:55:18 2023 문서
-rw-rw-r-- 1 kkm kkm 478 Fri Apr 7 10:02:10 2023 35.c
drwxr-xr-x 2 kkm kkm 4096 Thu Apr 13 20:57:31 2023 바탕화면
-rw-rw-r-- 1 kkm kkm 89 Wed Mar 29 10:04:50 2023 data4.txt
```

## 2-3. -s 500

```
kkm@kkm-VirtualBox: ~  
$ ls -s 500  
LS      show      28      35      ch3_1    macro    LS1.c    템플릿  
공개    다운로드  snap    음악    hanbit   비디오  문서     바탕화면  
사진    show.c
```

## 2-4. -l -s 500 or -ls 500

```
현재 활동  터미널  4월 13일 22:45  
kkm@kkm-VirtualBox: ~  
$ ls -l -s 500  
-rwxrwxr-x 1 kkm kkm 21112 Thu Apr 13 21:44:04 2023 LS  
-rwxrwxr-x 1 kkm kkm 16344 Wed Mar 29 20:57:23 2023 show  
-rwxrwxr-x 1 kkm kkm 16096 Fri Apr 7 11:18:09 2023 28  
-rwxrwxr-x 1 kkm kkm 16088 Fri Apr 7 10:03:14 2023 35  
-rwxrwxrwx 1 kkm kkm 16056 Fri Mar 31 11:26:10 2023 ch3_1  
-rwxrwxr-x 1 kkm kkm 15960 Fri Apr 7 09:44:16 2023 macro  
-rw-rw-r-- 1 kkm kkm 6662 Thu Apr 13 21:44:02 2023 LS1.c  
drwxr-xr-x 2 kkm kkm 4096 Sun Mar 26 11:55:18 2023 템플릿  
drwxr-xr-x 2 kkm kkm 4096 Sun Mar 26 11:55:18 2023 공개  
drwxr-xr-x 2 kkm kkm 4096 Sun Mar 26 11:55:18 2023 다운로드  
drwx----- 5 kkm kkm 4096 Fri Apr 7 10:25:07 2023 snap  
drwxr-xr-x 2 kkm kkm 4096 Sun Mar 26 11:55:18 2023 음악  
drwxrwxr-x 2 kkm kkm 4096 Wed Apr 12 19:43:27 2023 hanbit  
drwxr-xr-x 2 kkm kkm 4096 Sun Mar 26 11:55:18 2023 비디오  
drwxr-xr-x 2 kkm kkm 4096 Sun Mar 26 11:55:18 2023 문서  
drwxr-xr-x 2 kkm kkm 4096 Thu Apr 13 20:57:31 2023 바탕화면  
drwxr-xr-x 2 kkm kkm 4096 Sun Mar 26 11:55:18 2023 사진  
-rw-rw-r-- 1 kkm kkm 1475 Wed Mar 29 20:57:22 2023 show.c  
$ ls -ls 500  
-rwxrwxr-x 1 kkm kkm 21112 Thu Apr 13 21:44:04 2023 LS  
-rwxrwxr-x 1 kkm kkm 16344 Wed Mar 29 20:57:23 2023 show  
-rwxrwxr-x 1 kkm kkm 16096 Fri Apr 7 11:18:09 2023 28  
-rwxrwxr-x 1 kkm kkm 16088 Fri Apr 7 10:03:14 2023 35  
-rwxrwxrwx 1 kkm kkm 16056 Fri Mar 31 11:26:10 2023 ch3_1  
-rwxrwxr-x 1 kkm kkm 15960 Fri Apr 7 09:44:16 2023 macro  
-rw-rw-r-- 1 kkm kkm 6662 Thu Apr 13 21:44:02 2023 LS1.c
```

## 2-5. -F or -d -l

```

kkm@kkm-VirtualBox: ~
kkm@kkm-VirtualBox:~$ LS -F
템플릿/ 공개/ LS macro 28 다운로드/ macro.c show
snap/ 음악/ y.txt hanbit/ data2.txt 비디오/ LS1.c ch3_1.c
data1.txt ch3_1 28.c 35 o.txt linux.txt 문서/ 35.c
바탕화면/ data4.txt data3.txt show.c o1.txt 사진/
kkm@kkm-VirtualBox:~$ LS -d -l
템플릿/ 공개/ LS macro 28 다운로드/ macro.c show
snap/ 음악/ y.txt hanbit/ data2.txt 비디오/ LS1.c ch3_1.c
data1.txt ch3_1 28.c 35 o.txt linux.txt 문서/ 35.c
바탕화면/ data4.txt data3.txt show.c o1.txt 사진/
kkm@kkm-VirtualBox:~$
```

## 2-6. -R

```

kkm@kkm-VirtualBox:~$ LS -R
템플릿 공개 LS macro 28 다운로드 macro.c show
snap 음악 y.txt hanbit data2.txt 비디오 LS1.c ch3_1.c
data1.txt ch3_1 28.c 35 o.txt linux.txt 문서 35.c
바탕화면 data4.txt data3.txt show.c o1.txt 사진
./템플릿:

./공개:

./다운로드:

./snap:
snap-store snapd-desktop-integration firefox
./snap/snap-store:
common 638 current
./snap/snap-store/common:

./snap/snap-store/638:

./snap/snapd-desktop-integration:
49 57 common current
./snap/snapd-desktop-integration/49:

./snap/snapd-desktop-integration/57:

./snap/snapd-desktop-integration/common:
```

```

common    638      current
./snap/snap-store/common:

./snap/snap-store/638:

./snap/snapd-desktop-integration:
49         57         common    current
./snap/snapd-desktop-integration/49:

./snap/snapd-desktop-integration/57:

./snap/snapd-desktop-integration/common:

./snap/firefox:
common    2487     current
./snap/firefox/common:

./snap/firefox/2487:

./음악:

./hanbit:
1.txt
./비디오:

./문서:

```

## 2-7. -r

```

kkm@kkm-VirtualBox: ~
kkm@kkm-VirtualBox:~$ LS -r
템플릿 음악 사진 비디오 바탕화면 문서 다운로드 공개
y.txt snap show.c show o1.txt o.txt macro.c macro
LS1.c LS linux.txt hanbit data4.txt data3.txt data2.txt data1.txt
ch3_1.c ch3_1 35.c 35 28.c 28
kkm@kkm-VirtualBox:~$

```

감사합니다.