소스코드

```c
#include <sys/types.h>
#include <sys/stat.h>
#include <dirent.h>
#include <pwd.h>
#include <grp.h>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>
char type(mode_t);
char *perm(mode_t);
void printStat(char*, char*, struct stat*);
char *dir,*dir2=".",*dir3=".";
 /* 디렉터리 내용을 자세히 리스트한다. */
int main(int argc, char **argv)
{
        DIR *dp;
        //char *dir,*dir2;
        struct stat st;
        struct dirent *d;
        char path[BUFSIZ+1];
        if (argc == 1){
                dir = ".";
                dir2=".";
                dir3=".";
        }

        else{
                if(argv[1][0]=='-'){
                        if(argv[2]!=NULL)
                                dir=argv[2];
                        else
                                dir = ".";
                        dir2 = argv[1];
                }
                else{
                        dir = argv[1];
```

```c
                }
        }
        if (lstat(dir, &st) < 0)
                perror(path);
        if(type(st.st_mode) != 'd') {
                if(strcmp(dir2,"-l")==0 || strcmp(dir2,"-al")==0 || strcmp(dir2,"-la")==0){
                        printf("%c%s ", type(st.st_mode), perm(st.st_mode));
                        printf("%3ld ", st.st_nlink);
                        printf("%s        %s        ",        getpwuid(st.st_uid)->pw_name,
getgrgid(st.st_gid)->gr_name);
                        printf("%9ld ", st.st_size);
                        printf("%.12s ", ctime(&st.st_mtime)+4);
                        printf("%s\n", dir);
                }
                else{
                        printf("%s\n",dir);
                }
        }

        else{
                if ((dp = opendir(dir)) == NULL) // 디렉터리 열기
                        perror(dir);
                while ((d = readdir(dp)) != NULL) { // 디렉터리의 각 파일에 대해
                        sprintf(path, "%s/%s", dir, d->d_name); // 파일경로명 만들기
                        if (lstat(path, &st) < 0) // 파일 상태 정보 가져오기
                                perror(path);
                        printStat(path, d->d_name, &st); // 상태 정보 출력
                        //putchar('\n');
                }
                closedir(dp);
        }
        exit(0);
}
/* 파일 상태 정보를 출력 */
void printStat(char *pathname, char *file, struct stat *st) {
        if((strcmp(dir2,"-a")!=0 && strcmp(dir2,"-la")!=0 && strcmp(dir2,"-al")!=0) &&
file[0] == '.') return;
        if(strcmp(dir2,"-a")==0 || strcmp(dir2,".")==0 || strcmp(dir2,"-l")==0 ||
strcmp(dir2,"-la")==0 || strcmp(dir2,"-al")==0){
                if(strcmp(dir2,"-l")==0 || strcmp(dir2,"-al")==0 ||strcmp(dir2,"-la")==0){
                        //printf("%5ld ", st->st_blocks);
```

```c
                             printf("%c%s ", type(st->st_mode), perm(st->st_mode));
                             printf("%3ld ", st->st_nlink);
                             printf("%s        %s        ",        getpwuid(st->st_uid)->pw_name,
getgrgid(st->st_gid)->gr_name);
                             printf("%9ld ", st->st_size);
                             printf("%.12s ", ctime(&st->st_mtime)+4);
                  }
                  printf("%s\n", file);
         }
}
/* 파일 타입을 리턴 */
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');
}
/* 파일 사용권한을 리턴 */
char* perm(mode_t mode) {
         int i;
         static char perms[10] = "---------";

         strcpy(perms, "---------");

         for (i=0; i < 3; i++) {
                  if (mode & (S_IREAD >> i*3))
                           perms[i*3] = 'r';
                  if (mode & (S_IWRITE >> i*3))
                           perms[i*3+1] = 'w';
                  if (mode & (S_IEXEC >> i*3))
```

```
                    perms[i*3+2] = 'x';
        }
        return(perms);
}
```

실행결과

```
yoosumi@yoosumi-virtual-machine:~/SystemPro/homework$ ls
a.out  makefile  myls  myls.c  myls.o
yoosumi@yoosumi-virtual-machine:~/SystemPro/homework$ ls -l
total 56
-rwxrwxr-x 1 yoosumi yoosumi 17360 10월 26 00:41 a.out
-rw-rw-r-- 1 yoosumi yoosumi    63 11월 14 19:39 makefile
-rwxrwxr-x 1 yoosumi yoosumi 17512 11월 14 19:52 myls
-rw-rw-r-- 1 yoosumi yoosumi  2968 11월 14 19:52 myls.c
-rw-rw-r-- 1 yoosumi yoosumi  7144 11월 14 19:52 myls.o
yoosumi@yoosumi-virtual-machine:~/SystemPro/homework$ ls -al /etc/passwd
-rw-r--r-- 1 root root 2744  9월 22 02:14 /etc/passwd
yoosumi@yoosumi-virtual-machine:~/SystemPro/homework$ ls a.out
a.out
yoosumi@yoosumi-virtual-machine:~/SystemPro/homework$ ls -a ..
.    a.out     list1    list2.c   mysql     student2.c  student.dat  test
..   homework  list1.c  makefile  student   student.c   student.o    week_9
yoosumi@yoosumi-virtual-machine:~/SystemPro/homework$
```