

# LAB 3

System Programming

ls 명령어

# 실습 문제

- 다음 명령어들의 실행 결과를 설명하고 어떻게 구현할 수 있을 지 생각해 보세요

\$ ls - i

\$ ls - l

\$ ls - F

\$ ls - R

```
[redbrain@localhost sub]$ ls -F
catn* convert= hello.txt lncatn@ pipefile| quiz2* temp/ test
[redbrain@localhost sub]$ ls -lF
합계 32
-rwxrwxr-x. 1 redbrain redbrain 8704 7월 9 2023 catn*
srwxrwxr-x. 1 redbrain redbrain 0 2월 18 15:18 convert=
-rw-rw-r--. 1 redbrain redbrain 33 7월 9 2023 hello.txt
lrwxrwxrwx. 1 redbrain redbrain 4 4월 10 16:47 lncatn -> catn*
prw-rw-r--. 1 redbrain redbrain 0 4월 10 16:45 pipefile|
-rwxrwxr-x. 1 redbrain redbrain 8664 4월 19 2023 quiz2*
drwxrwxr-x. 2 redbrain redbrain 32 4월 10 16:41 temp/
-rw-rw-r--. 1 redbrain redbrain 16 2월 18 14:16 test
```

```
[redbrain@localhost sub]$ ls -R
.:
catn convert hello.txt lncatn pipefile| quiz2 temp test

./temp:
file1 file2
```

## ls -i (list1.c)

```
#include <sys/types.h>
#include <sys/stat.h>
#include <dirent.h>
#include <stdio.h>
#include <stdlib.h>

/* 디렉터리 내의 파일 이름들을 리스트한다. */
int main(int argc, char **argv) {
    DIR *dp;
    char *dir;
    struct dirent *d;
    struct stat st;
    char path[BUFSIZ+1];

    if (argc == 1)
        dir = "."; // 현재 디렉터를 대상으로
    else dir = argv[1];

    if ((dp = opendir(dir)) == NULL) { // 디렉터리 열기
        perror(dir);
        exit(1);
    }
    while ((d = readdir(dp)) != NULL) // 디렉터리 내의 각 엔트리에 대해
        printf("%s %lu\n", d->d_name, d->d_ino); // 파일 이름, i-노드 번호 출력

    closedir(dp);
    exit(0);
}
```

# ls - l (list2.c)

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

char type(mode_t);
char *perm(mode_t);
void printStat(char*, char*, struct stat*);
```

```
/* 디렉토리 내용을 자세히 리스팅한다. */
int main(int argc, char **argv)
{
    DIR *dp;
    char *dir;
    struct stat st;
    struct dirent *d;
    char path[BUFSIZ+1];

    if (argc == 1)
        dir = ".";
    else dir = argv[1];

    if ((dp = opendir(dir)) == NULL) // 디렉토리 열기
        perror(dir);

    while ((d = readdir(dp)) != NULL) { // 디렉토리 내의 각 파일에 대해
        sprintf(path, "%s/%s", dir, d->d_name); // 파일 경로명 만들기
        if (lstat(path, &st) < 0) // 파일 상태 정보 가져오기
            perror(path);
        else
            printStat(path, d->d_name, &st); // 상태 정보 출력
    }

    closedir(dp);
    exit(0);
}
```

```
/* 파일 상태를 출력 */
```

```
void printStat(char *pathname, char *file, struct stat *st) {
```

```
    printf("%5d ", st->st_blocks);
```

```
    printf("%c%s ", type(st->st_mode), perm(st->st_mode));
```

```
    printf("%3d ", st->st_nlink);
```

```
    printf("%s %s ", getpwuid(st->st_uid)->pw_name, getgrgid(st->st_gid)->gr_name);
```

```
    printf("%9d ", 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');
```

```
}
```

## ls -l (list2.c)

```
/* 파일 허가를 리턴 */
```

```
char* perm(mode_t mode) {
```

```
    int i;
```

```
    static char perms[10];
```

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

```
    for (i=0; i < 3; i++) {
```

```
        if (mode & (S_IRUSR >> i*3))
```

```
            perms[i*3] = 'r';
```

```
        if (mode & (S_IWUSR >> i*3))
```

```
            perms[i*3+1] = 'w';
```

```
        if (mode & (S_IXUSR >> i*3))
```

```
            perms[i*3+2] = 'x';
```

```
    }
```

```
    return(perms);
```

```
}
```

# ls - F (list3.c) 구현하기

파일 형식을 알리는 문자를 각 파일 뒤에 추가한다. 일반적으로 실행파일은 "\*", 경로는 "/", 심블릭 링크는 "@", FIFO는 "|", 소켓은 "=", 일반적인 파일은 없다.

- 파일의 종류에 대한 정보를 얻을 수 있는 방법은?
- 실행파일인지 아닌지 구분하는 방법은?

```
[redbrain@localhost sub]$ ./list3
convert=
catn*
hello.txt
test
quiz2*
temp/
pipefile|
lncatn@
list*
list4*
list3*
[redbrain@localhost sub]$
[redbrain@localhost sub]$ ls -F
catn*  convert=  hello.txt  list*  list3*  list4*  lncatn@  pipefile|  quiz2*  temp/  test
[redbrain@localhost sub]$
```

# ls - F (list3.c) 구현하기

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

```
char type(mode_t);
int IsExecutable(mode_t mode);
void printClass(char *pathname, char *file, struct stat *st);
```

```
int main(int argc, char **argv)
{
    DIR *dp;
    char *dir;
    struct stat st;
    struct dirent *d;
    char path[BUFSIZ+1];

    if (argc == 1)
        dir = ".";
    else dir = argv[1];

    if ((dp = opendir(dir)) == NULL) // 디렉토리 열기
        perror(dir);

    while ((d = readdir(dp)) != NULL) { // 디렉토리 내의 각 파일에 대해
        sprintf(path, "%s/%s", dir, d->d_name); // 파일 경로명 만들기
        if (lstat(path, &st) < 0) // 파일 상태 정보 가져오기
            perror(path);
        else
            printClass(path, d->d_name, &st); //
    }

    closedir(dp);
    exit(0);
}
```

## ls - F (list3.c) 구현하기

```
int IsExecutable(mode_t mode){

}

void printClass(char *pathname, char *file, struct stat *st) {

}

}
```

```
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');
}
```

## 자가 테스트

1. 각 함수의 역할 이해
2. 각 함수의 논리적 흐름을 생각할 수 있고, 필요한 주요 구문을 생각해 낼 수 있다.
3. 강의 자료나 책을 참조하여 코드를 구성할 수 있다.
4. 대부분의 코드나 구문을 참조하지 않고도 함수를 실행 가능하도록 (거의) 정확하게 코드를 작성할 수 있다.



# ls -R (list4.c) 구현하기

하위 경로와 그 안에 있는 모든 파일들도 나열한다.

```
[redbrain@localhost sub]$ ls -R
.:
catn  convert  hello.txt  list  list3  list4  lncatn  pipefile  quiz2  temp  test

./temp:
file1  file2
[redbrain@localhost sub]$
```

디렉토리내의 파일의 종류가  
디렉토리면 다시 디렉토리를  
열고 내용 읽기를 반복.

```
[redbrain@localhost sub]$ ./list4
.:
convert
catn
hello.txt
test
quiz2
temp
pipefile
lncatn
list
list4

./temp:
file1
file2
[redbrain@localhost sub]$
```