

LAB03

시스템 프로그래밍_류은경Prof

2021115744_권구태

1. (who3.c) who2.c를 수정해서 코드 작성 (수정된 부분, 핵심 요약)

```
// utmplib.h - 함수 원형 선언
#ifndef UTMLIB_H
#define UTMLIB_H

#include <utmp.h>

// 함수 원형 (프로토타입) 선언
int utmp_open(char *filename);
struct utmp *utmp_next();
int utmp_reload();
void utmp_close();

#endif
```

#ifndef, #define, #endif를 통해 중복 컴파일 방지

```
#include "utmplib.h"
```

in utmplib.c 파일, who3.c 파일 내부에 선언

[주의할 점 및 알게된 점]

1. read, close 함수는 unistd.h 헤더 파일 내부에 있다.
2. C언어에서는 함수가 사용되기 전에 컴파일러에게 해당 함수의 존재를 알려줘야 함. 이를 위해 함수원형(프로토타입)을 선언하지 않으면, 컴파일러는 해당 함수가 어떤 반환형을 가지는지, 매개변수 타입이 무엇인지 알 수 없다.
3. #include "utmplib.c"와 같이 소스 파일을 직접 include 하는 것은 일반적이지 않음. 중복 컴파일 문제나 가시성 문제 발생할 수 있음 -> gcc who3.c utmplib.c -o who3로 두 파일을 링킹하는 실행파일을 만든다고 하면, who3.c에서는 사용된 파일이 선언이 되어있어야 하기 때문에 #include "utmplib.h" 해야하고 utmplib.c에서는 .h에서 선언한 프로토타입과 일치하는지를 보기 위해 선언이 필요하다. 링킹 단계에서는 .h 파일을 사용하여 함수 시그니처를 연결하고 이를 통해 최종 실행파일을 만든다.

[실행 결과 사진]

```
kwongutae@kwongutae-Inspiron-15-5510: ~/Documents/경북대학교/강의자료/시스템프로그래밍_류은경/2주차/ch02$ gcc who3.c utmplib.c -o who3 -Wall
kwongutae@kwongutae-Inspiron-15-5510: ~/Documents/경북대학교/강의자료/시스템프로그래밍_류은경/2주차/ch02$ ./who3
kwonguta seat0 Mar 18 09:09 (login screen)
kwonguta tty2 Mar 18 09:09 (tty2)
```

2. (ls3.c) ls2.c를 수정하여 **ls -l** 명령어 출력과 같이 디렉토리 안의 모든 파일의 블록 크기 **“total”** 사이즈 출력하라.

[수정한 부분의 코드 분석 및 설명]

```
void do_ls( char dirname[], int inverse )
```

기존의 void do_ls(char dirname[])함수에 두번째 인자를 추가하여, void do_ls(char dirname[], int inverse)형태로 만들어줌.

```
if (inverse) {
    int i = 0;
    if( ( dir_ptr = opendir( dirname ) ) == NULL )
        fprintf(stderr, "ls2: cannot open %s\n", dirname);
    else
    {
        while ( ( direntp = readdir( dir_ptr ) ) != NULL ) {
            dostatTotal(direntp->d_name);
        }
        printf("total: %ld\n", total);
        closedir(dir_ptr);
    }
}
```

inverse가 1 혹은 0이 아니라면 total: [블록번호]를 출력하는 부분을 출력하도록 하고, 0이라면 해당 부분을 건너뛰도록 if문 작성.

```
void dostatTotal( char *filename )
{
    struct stat info;

    if( stat( filename, &info ) == -1 ) /* cannot stat */
        perror( filename );          /* say why */
    else
        /* else show info */
        CountTotal( filename, &info );
}

void CountTotal( char *filename, struct stat *info_p)
{
    total += (long) info_p->st_blksize;
}
```

stat 구조를 가진 info 구조체 생성하고, 해당 구조체의 주소값을 넘겨 파일이나 디렉터리의 상태 정보를 가져오는 stat 시스템 함수를 통해 성공적으로 info 구조체에 값을 저장하면, info의 주소를 넘기는 dostatTotal 함수와 info 구조체의 st_blksize(블록 사이즈)를 전역변수 total에 저장하는 CountTotal 함수를 추가로 작성.

```

if( ac == 1 )
    do_ls( "." , 0);
else if (ac == 2 && (strcmp(av[1], "-l") == 0)) {
    do_ls( "." , 1);
}
else
    while( --ac ) {
        printf("%s:\n", *++av );
        do_ls( *av , 0);
    }

```

이는 main 함수 안의 코드.

따라서 만약 `ac == 1`, 즉 `./ls` 만 명령어로 입력했다면 `do_ls(".", 0)`이 실행되고(리눅스의 파일 정보 출력 형식 출력), `./ls -l`을 입력했다면 `do_ls(".", 1)`이 실행(`total: [전체 블록의 합] + 리눅스의 파일 정보 출력 형식 출력`), 그 외에는 모두 오류를 알리는 출력이 나오도록 코드 변경.

[주의할 점 및 알게된 점]

1. 포인터배열(문자열)의 비교에는 `strcmp` 함수를 사용해야 함.
2. `stat` 구조체와 `stat` 함수: 파일이나 디렉터리의 상태 정보를 저장할 수 있는 구조의 구조체와 해당 구조의 정보를 가져오는 함수
3. `stat` 함수로 가져온 정보 안에 `st_blksize`라고 하여 블록 크기 정보가 들어있음.
4. `readdir`함수는 `DIR`포인터 변수를 인자로 받고 해당 포인터를 포인터 배열의 다음 인덱스로 넘기며 `NULL`이 될 때까지 파일의 정보를 읽어온다. 즉 한번 읽어오면 다음 읽어들이는 파일을 리턴하기전에 포인터변수의 인덱스를 1 증가시킴.

[실행 결과 사진]

```

kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의자료/시스템프로그래밍_류은경/3주차/ch03$ ./ls -l
total: 73728
-rw-rw-r-- 1 kwongutae kwongutae 338 Dec 28 13:49 filesize.c
drwxrwxr-x 3 kwongutae kwongutae 4096 Mar 18 10:08 ..
-rw-rw-r-- 1 kwongutae kwongutae 0 Mar 18 14:44 text.txt
-rw-r--r-- 1 kwongutae kwongutae 12288 Mar 18 11:11 ls.swp
-rw-rw-r-- 1 kwongutae kwongutae 1886 Mar 18 17:05 ls4.c
-rwxrwxr-x 1 kwongutae kwongutae 16472 Mar 18 17:05 ls4
-rwxrwxr-x 1 kwongutae kwongutae 16096 Mar 18 10:26 filesize
-rwxrwxr-x 1 kwongutae kwongutae 16136 Mar 18 10:34 fileinfo
-rw-rw-r-- 1 kwongutae kwongutae 1039 Dec 28 13:49 fileinfo.c
-rw-r--r-- 1 kwongutae kwongutae 12288 Mar 18 10:21 ls1.c.swp
-rwxrwxr-x 1 kwongutae kwongutae 16792 Mar 18 11:23 ls2
-rw-rw-r-- 1 kwongutae kwongutae 0 Mar 18 16:55 text2.txt
-rw-rw-r-- 1 kwongutae kwongutae 756 Mar 18 12:17 ls1.c
-rw-rw-r-- 1 kwongutae kwongutae 3374 Mar 18 11:26 ls2.c
-rwxrwxr-x 1 kwongutae kwongutae 16232 Mar 18 12:15 ls1
-rwxrwxr-x 1 kwongutae kwongutae 16936 Mar 19 10:43 ls
drwxrwxr-x 2 kwongutae kwongutae 4096 Mar 19 10:43 .
-rw-rw-r-- 1 kwongutae kwongutae 4229 Mar 19 10:41 ls3.c

```

3. (ls4.c)

1) (Sorting 후 출력) Modify ls1.c so it sorts the list of files. (qsort 함수 사용)

- qsort 사용 예제 참고

(<http://www.cplusplus.com/reference/cstdlib/qsort/>)

2) (역순으로 출력) Standard ls supports a -r option to print the list in reverse order. Add that option.

[수정한 부분의 코드 분석 및 설명]

```
void do_ls( char dirname[], int inverse )
```

기존의 void do_ls(char dirname[])함수에 두번째 인자를 추가하여, void do_ls(char dirname[], int inverse)형태로 만들어줌.

```
char *strArr[MAX];

while ((direntp = readdir(dir_ptr)) != NULL) {
    strArr[i] = strdup(direntp->d_name); // 파일명 복사
    i++;
}

closedir(dir_ptr);
quickSort(strArr, 0, i - 1); // i == 11
if (inverse) {
    for (int j = i - 1; j >= 0; j--) {
        printf("%s\n", strArr[j]);
    }
} else {
    for (int j = 0; j < i; j++) {
        printf("%s\n", strArr[j]);
    }
}
```

포인터 배열을 선언한 후, DIR 포인터 변수를 이용해 readdir 함수로 파일이 NULL을 사리킬 때까지, 파일을 불러와, 불러온 파일의 이름을 포인터 배열에 저장함. 이후 quickSort함수를 이용해 오름차순 정리, 이후 만약 inverse가 1 혹은 0이 아니라면 역순으로 출력, 0이라면 오름차순 출력하도록 if/else문 작성.

```
if( ac == 1)
    do_ls(".", 0);
```

```

else if (ac == 2 && strcmp(av[1], "-r") == 0) {
    do_ls(".", 1);
} else {
    printf("%s\n\n %d", av[1], ac);
    while( --ac ){
        printf("%s:\n", *++av );
        do_ls( *av , 0);
    }
}

```

메인 함수에서 ./ls 만 입력하면 오름차순 정렬을 출력하도록 do_ls 함수의 두번째 인자에 0을 넣고, ./ls -r 을 입력하면 역순 출력을 할 수 있도록, 그 외에는 오류라고 출력하도록 if/else 문을 작성함.

[주의할 점 및 알게된 점]

1. quickSort함수를 사용할 때, 넣어주는 인자의 수를 잘 보고 넣어야 함. 그렇지 않으면 segmentation error 발생.

[실행 결과 사진]

```

kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의자료/시스템프로그래밍_류은경/3주차/ch03$ ./ls1
..
.ls.swp
.ls1.c.swp
fileinfo.c
filesize
ls4
fileinfo
ls4.c
filesize.c
ls1.c
ls
ls2
ls1
ls2.c
ls3.c
text.txt
text2.txt
kwongutae@kwongutae-Inspiron-15-5510:~/Documents/경북대학교/강의자료/시스템프로그래밍_류은경/3주차/ch03$ ./ls1 -r
text2.txt
text.txt
ls3.c
ls2.c
ls1
ls2
ls
ls1.c
filesize.c
ls4.c
fileinfo
ls4
filesize
fileinfo.c
.ls1.c.swp
.ls.swp
..

```