

파일 시스템 II

- 파일 처리 함수 1

컴퓨터소프트웨어학과

김병국 교수



- 파일의 상태정보를 추출할 수 있다.
- 파일 접근 시 위치(오프셋)을 설정할 수 있다.



- 파일 상태정보 추출
- 접근 위치 이동
- 접근 위치 이동 실습



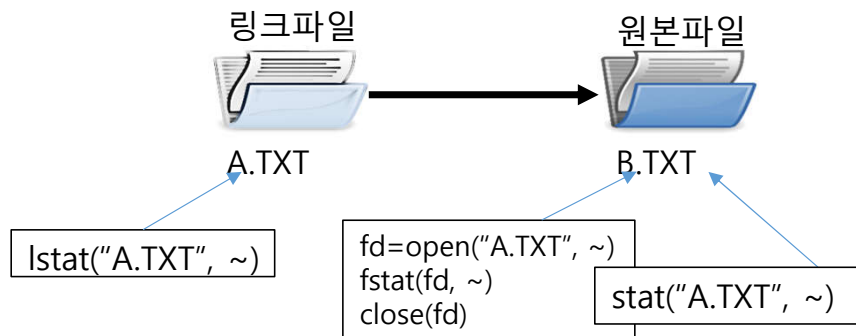
1. 파일 상태정보 추출 (1/5)

□ 파일 상태 추출

- `fstat()`, `stat()`, `lstat()`의 세 종류의 함수가 있음
- 지정한 파일에 대한 상태를 `statbuf`에 기록
- `fstat()` : 파일기술자를 통해 상태정보를 추출
- `stat()` & `lstat()`는 주어진 파일명(문자열)을 통해 상태정보를 추출
 - `stat()`의 경우 링크 파일일 때 원본에 접근
 - `lstat()`는 `stat()`와는 달리 주어진 파일 자체 정보를 추출

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>

int stat(const char *pathname, struct stat *statbuf);
int fstat(int fd, struct stat *statbuf);
int lstat(const char *pathname, struct stat *statbuf);
```



```
struct stat {
    dev_t    st_dev;        /* ID of device containing file */
    ino_t    st_ino;        /* Inode number */
    mode_t    st_mode;      /* File type and mode */
    nlink_t   st_nlink;     /* Number of hard links */
    uid_t    st_uid;       /* User ID of owner */
    gid_t     st_gid;       /* Group ID of owner */
    dev_t     st_rdev;      /* Device ID (if special file) */
    off_t     st_size;      /* Total size, in bytes */
    blksize_t st_blksize;   /* Block size for filesystem I/O */
    ...
};
```

[**stat** 구조체 내용 中]



1. 파일 상태정보 추출 (2/5)

실습

■ 공통 파일 (1/2)

```
1 #include <sys/types.h>
2 #include <sys/stat.h>
3 #include <time.h>
4 #include <stdio.h>
5 #include "print_fileinfo.h"
6
7 int
8 print_FileInfo(struct stat stBuf)
9 {
10     printf("File type:      ");
11     switch (stBuf.st_mode & S_IFMT) {
12         case S_IFBLK:  printf("block device\n");      break;
13         case S_IFCHR:  printf("character device\n");    break;
14         case S_IFDIR:  printf("directory\n");            break;
15         case S_IFIFO:  printf("FIFO/pipe\n");            break;
16         case S_IFLNK:  printf("symlink\n");              break;
17         case S_IFREG:  printf("regular file\n");         break;
18         case S_IFSOCK: printf("socket\n");               break;
19         default:       printf("unknown?\n");             break;
20     }
21 }
```

[print_fileinfo.c]

```
1 #include <sys/stat.h>
2
3 int print_FileInfo(struct stat stBuf);
4
```

[print_fileinfo.h]


print_fileinfo.c


print_fileinfo.h



1. 파일 상태정보 추출 (3/5)

□ 실습

▪ 공통 파일 (2/2)

```
22 2 printf("I-node number:          %ld\n", (long)stBuf.st_ino);
23   printf("Mode:                  %lo (octal)\n",
24         (unsigned long)stBuf.st_mode);
25   printf("Link count:             %ld\n", (long)stBuf.st_nlink);
26   printf("Ownership:              UID=%ld  GID=%ld\n",
27         (long)stBuf.st_uid, (long)stBuf.st_gid);
28   printf("Preferred I/O block size: %ld bytes\n",
29         (long)stBuf.st_blksize);
30   printf("File size:                  %lld bytes\n",
31         (long long)stBuf.st_size);
32   printf("Blocks allocated:           %lld\n",
33         (long long)stBuf.st_blocks);
34   printf("Last status change:         %s", ctime(&stBuf.st_ctime));
35   printf("Last file access:           %s", ctime(&stBuf.st_atime));
36   printf("Last file modification:     %s", ctime(&stBuf.st_mtime));
37
38   printf("\n");
39
40   return 0;
41 }
```

[print_fileinfo.c]

1. 파일 상태정보 추출 (4/5)

□ 실습

- 파일명: stat.c

```
1 #include <sys/types.h>
2 #include <sys/stat.h>
3 #include <time.h>
4 #include <stdio.h>
5 #include <stdlib.h>
6
7 #include "print_fileinfo.h"
8
9 int
10 main(int argc, char* argv[])
11 {
12     struct stat stBuf;
13
14     if (argc != 2) {
15         fprintf(stderr, "Usage: %s <pathname>\n", argv[0]);
16         return -1;
17     }
18
```

```
18
19     if (stat(argv[1], &stBuf) == -1) {
20         fprintf(stderr, "lstat");
21         return -1;
22     }
23
24     print_FileInfo(stBuf);
25
26     return 0;
27 }
28
```

stat() 함수 호출



1. 파일 상태정보 추출 (5/5)

□ 실습

- 파일명: lstat.c

```
1 #include <sys/types.h>
2 #include <sys/stat.h>
3 #include <time.h>
4 #include <stdio.h>
5 #include <stdlib.h>
6
7 #include "print_fileinfo.h"
8
9 int
10 main(int argc, char* argv[])
11 {
12     struct stat stBuf;
13
14     if (argc != 2) {
15         fprintf(stderr, "Usage: %s <pathname>\n", argv[0]);
16         return -1;
17     }
18
```

```
18
19     if (lstat(argv[1], &stBuf) == -1) {
20         fprintf(stderr, "lstat");
21         return -1;
22     }
23
24     print_FileInfo(stBuf);
25
26     return 0;
27 }
28
```

lstat() 함수 호출

2. 접근 위치 이동(L) (1/2)

□ 위치 이동(저수준)

- 함수 : `lseek()`
 - 파일의 접근 위치를 주어진 값으로 이동
 - 파일 기술자를 통한 접근 방식
- 인자 :
 - `fd` : 파일의 기술자
 - `offset` : 이동할 위치
 - `whence` : 기준(`SEEK_SET`, `SEEK_CUR`, `SEEK_END`)
- 결과 값:
 - 성공 : 변경된 오프셋
 - 실패 : `-1`

```
#include <sys/types.h>
#include <unistd.h>

off_t lseek(int fd, off_t offset, int whence);
```

[`lseek()` 함수의 프로토타입]

기준	설명
SEEK_SET	offset의 위치로 옮김
SEEK_CUR	현재까지 읽은 곳으로부터 offset만큼 이동
SEEK_END	마지막 위치에서부터 offset만큼 이동(좌: 음수, 우: 양수)



2. 접근 위치 이동(L) (2/2)

□ 실습

- 지정한 위치에서 내용 출력

```
1 #include <sys/types.h>
2 #include <sys/stat.h>
3 #include <fcntl.h>
4 #include <stdio.h>
5 #include <unistd.h>
6
7 int main(int argc, char *argv[])
8 {
9     int nFd = -1;
10    int nWhence, nCount, nOffset;
11    char pBuf[BUFSIZ];
12    int nLen = 0;
13
14    if (argc != 2)
15    {
16        printf("Usage: cmd [filename]\n");
17        return -1;
18    }
19
20    nFd = open(argv[1], O_RDONLY);
```

```
22 printf("Select whence: SEEK_SET(0), SEEK_CUR(1), SEEK_END(2)\n");
23 scanf("%d", &nWhence);
24 printf("Where will you start to read?\n");
25 scanf("%d", &nOffset);
26 printf("How many characters will you read?\n");
27 scanf("%d", &nCount);
28
29 switch (nWhence)
30 {
31     case 0:
32         lseek(nFd, nOffset, SEEK_SET);
33         break;
34     case 1:
35         lseek(nFd, nOffset, SEEK_CUR);
36         break;
37     default:
38         lseek(nFd, nOffset, SEEK_END);
39         break;
40 }
41
42 nLen = read(nFd, pBuf, nCount);
43 printf("Data at (%d):\n", nOffset);
44 printf("%s\n", pBuf);
45
46 close(nFd);
47
48 return 0;
49 }
```

[파일명: lseek.c]

3. 접근 위치 이동(H) (1/5)

□ 위치 이동(고수준) (1/3)

- 함수 : fseek()
 - 파일의 접근 위치를 주어진 값으로 이동
 - FILE 구조체를 통한 접근 방식(고수준)
- 인자 :
 - *stream : FILE 구조체 포인터
 - offset : 이동할 위치
 - whence : 기준(SEEK_SET, SEEK_CUR, SEEK_END)
- 결과 값:
 - 성공 : 변경된 위치값
 - 실패 : -1

```
#include <stdio.h>

int fseek(FILE *stream, long offset, int whence);
long ftell(FILE *stream);

void rewind(FILE *stream);

int fgetpos(FILE *restrict stream, fpos_t *restrict pos);
int fsetpos(FILE *stream, const fpos_t *pos);
```

【프로토타입】



3. 접근 위치 이동(H) (2/5)

□ 위치 이동(고수준) (2/3)

- 함수 : `ftell()`
 - 현재 위치를 반환
 - 인자 :
 - `*stream` : FILE 구조체 포인터
 - 결과 값:
 - 성공 : 현재 위치
 - 실패 : -1

- 함수 : `rewind()`
 - 현재 위치를 시작점(처음)으로 변경
 - 인자 :
 - `*stream` : FILE 구조체 포인터
 - 결과 값:
 - 성공 : 0
 - 실패 : -1

```
#include <stdio.h>

int fseek(FILE *stream, long offset, int whence);
long ftell(FILE *stream);

void rewind(FILE *stream);

int fgetpos(FILE *restrict stream, fpos_t *restrict pos);
int fsetpos(FILE *stream, const fpos_t *pos);
```

[프로토타입]



3. 접근 위치 이동(H) (3/5)

□ 위치 이동(고수준) (3/3)

- 함수 : fgetpos()
 - 현재 위치값을 추출
 - 인자 :
 - *stream : FILE 구조체 포인터
 - *pos : 현재 위치
 - 결과 값:
 - 성공 : 0
 - 실패 : -1

- 함수 : fsetpos()
 - 현재 위치값을 설정
 - 인자 :
 - *stream : FILE 구조체 포인터
 - *pos : 설정할 위치
 - 결과 값:
 - 성공 : 0
 - 실패 : -1

```
#include <stdio.h>

int fseek(FILE *stream, long offset, int whence);
long ftell(FILE *stream);

void rewind(FILE *stream);

int fgetpos(FILE *restrict stream, fpos_t *restrict pos);
int fsetpos(FILE *stream, const fpos_t *pos);
```

[프로토타입]



3. 접근 위치 이동(H) (4/5)

□ 실습

- 지정한 위치의 내용을 출력

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include <unistd.h>
5
6  int
7  main(int argc, char* argv[])
8  {
9      FILE* fp = NULL;
10     long nOffset = 0;
11     char strBuffer[BUFSIZ] = { 0, };
12
13     if (argc != 3) {
14         printf("Usage: %s <pathname> <offset>\n", argv[0]);
15         return -1;
16     }
```

```
18     nOffset = atoi(argv[2]);
19
20     fp = fopen(argv[1], "r");
21     if (fseek(fp, nOffset, SEEK_SET) < 0) {
22         printf("Setting Position has been failed.\n");
23         return -1;
24     }
25
26     memset(strBuffer, 0, BUFSIZ);
27
28     if (fgets(strBuffer, BUFSIZ, fp) != NULL) {
29         printf("Data : %s", strBuffer);
30     }
31
32     fclose(fp);
33
34     return 0;
35 }
```

[파일명: fseek.c]

수고하셨습니다.

