인덕대학교

파일 시스템 II
- 파일 처리 함수 1

컴퓨터소프트웨어학과

김병국 교수

학습목표

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

목차

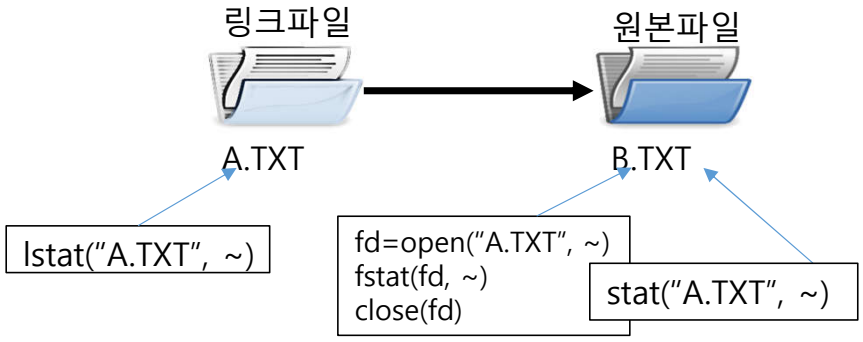
1. 파일 상태정보 추출 (1/5)

❑ 파일 상태 추출

- fstat(), stat(), lstat()의 세 종류의 함수가 있음
- 지정한 파일에 대한 상태를 statbuf에 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

```c
#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);
```




```c
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

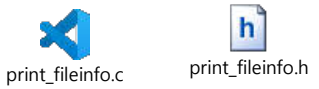
- 공통 파일 (1/2)

```c
#include <sys/stat.h>

int print_FileInfo(struct stat stBuf);

```

[print_fileinfo.h]




```c
#include <sys/types.h>
#include <sys/stat.h>
#include <time.h>
#include <stdio.h>
#include "print_fileinfo.h"

int
print_FileInfo(struct stat stBuf)
{
    printf("File type:                ");
    switch (stBuf.st_mode & S_IFMT) {
    case S_IFBLK:  printf("block device\n");            break;
    case S_IFCHR:  printf("character device\n");        break;
    case S_IFDIR:  printf("directory\n");               break;
    case S_IFIFO:  printf("FIFO/pipe\n");               break;
    case S_IFLNK:  printf("symlink\n");                 break;
    case S_IFREG:  printf("regular file\n");            break;
    case S_IFSOCK: printf("socket\n");                  break;
    default:       printf("unknown?\n");                break;
    }
```

[print_fileinfo.c]

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

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

#include "print_fileinfo.h"

int
main(int argc, char* argv[])
{
    struct stat stBuf;

    if (argc != 2) {
        fprintf(stderr, "Usage: %s <pathname>\n", argv[0]);
        return -1;
    }

    if (stat(argv[1], &stBuf) == -1) {
        fprintf(stderr, "lstat");
        return -1;
    }

    print_FileInfo(stBuf);

    return 0;
}
```

1

2

stat() 함수 호출

1. 파일 상태정보 추출 (5/5)

인덕대학교

❑ 실습

- 파일명: lstat.c

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

#include "print_fileinfo.h"

int
main(int argc, char* argv[])
{
    struct stat stBuf;

    if (argc != 2) {
        fprintf(stderr, "Usage: %s <pathname>\n", argv[0]);
        return -1;
    }

    if (lstat(argv[1], &stBuf) == -1) {
        fprintf(stderr, "lstat");
        return -1;
    }

    print_FileInfo(stBuf);

    return 0;
}
```

1

2

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]

2. 접근 위치 이동(L) (2/2)

❑ 실습

- 지정한 위치에서 20개 문자 출력

```c
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdlib.h> // for atoi()
#include <stdio.h>
#include <unistd.h>

int
main(int argc, char* argv[])
{
    int nFd = -1;
    char pBuf[BUFSIZ] = { 0, };
    int nLen = 0;

    if (argc != 4)
    {
        printf("Usage: cmd [filename] [Whence] [Offset]₩n");
        printf("₩t-Whence: SEEK_SET(0), SEEK_CUR(1), SEEK_END(2)₩n");
        return -1;
    }

    nFd = open(argv[1], O_RDONLY);

    switch (atoi(argv[2]))
    {
        case 0:
            lseek(nFd, atoi(argv[3]), SEEK_SET);
            break;
        case 1:
            lseek(nFd, atoi(argv[3]), SEEK_CUR);
            break;
        default:
            lseek(nFd, atoi(argv[3]), SEEK_END);
            break;
    }

    nLen = read(nFd, pBuf, 20);
    printf("Data: %s₩n", pBuf);

    close(nFd);

    return 0;
}
```

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 *stream, fpos_t *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 *stream, fpos_t *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

```c
#include <stdio.h>

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

long ftell(FILE *stream);

void rewind(FILE *stream);

int fgetpos(FILE *stream, fpos_t *pos);

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

[프로토타입]

3. 접근 위치 이동(H) (4/5)

❑실습

```c
#include <stdio.h>
#include <string.h>
#include <unistd.h>

int
main(int argc, char* argv[])
{
    FILE* fp = NULL;
    long nOffset = 0;
    char strBuffer[BUFSIZ] = { 0, };

    if (argc != 2) {
        fprintf(stderr, "Usage: %s <pathname>\n", argv[0]);
        return -1;
    }

    fp = fopen(argv[1], "r");

    memset(strBuffer, 0, BUFSIZ);

    for(int i=0; ; i++) {
        nOffset = ftell(fp);
        if (fgets(strBuffer, BUFSIZ, fp) == NULL) {
            break;
        }

        printf("%03d : %03d : %s", i , nOffset, strBuffer);

        memset(strBuffer, 0, BUFSIZ);
    }

    fclose(fp);

    return 0;
}
```

3. 접근 위치 이동(H) (5/5)

❑실습

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>

int
main(int argc, char* argv[])
{
    FILE* fp = NULL;
    long nOffset = 0;
    char strBuffer[BUFSIZ] = { 0, };

    if (argc != 3) {
        fprintf(stderr, "Usage: %s <pathname> <offset>\n", argv[0]);
        return -1;
    }

    nOffset = atoi(argv[2]);

    fp = fopen(argv[1], "r");
    if (fseek(fp, nOffset, SEEK_SET) < 0) {
        fprintf(stderr, "Setting Position has been failed.\n");
        return -1;
    }
    memset(strBuffer, 0, BUFSIZ);
    if (fgets(strBuffer, BUFSIZ, fp) != NULL) {
        printf("Data : %s", strBuffer);
    }

    fclose(fp);

    return 0;
}
```

수고하셨습니다.