

System Programming & OS 실습

9. mycreat, mycat, mycp, myls

이성현, 최민국
Dankook University
{leesh812, mgchoi}@dankook.ac.kr

- Linux File Programming 실습
 - open() 을 활용, mycreate 프로그램을 만들 수 있다.
 - read() 를 활용, mycat 프로그램을 만들 수 있다.
 - readdir() 를 활용, myls 프로그램을 만들 수 있다.
 - open(),read(),write(),fstat()을 활용, mycp 프로그램을 만들 수 있다.

- 실습1: 복습 mycreat, mycat
- 실습2: mycp - open(),read(),write()
- /* Break */
- 데이터와 메타데이터, 파일과 디렉토리
- 실습3: mycp 심화 - stat(), fstat()
- 실습4: myls - readdir()

```
$sudo yum install git
```

```
$git clone https://github.com/DKU-EmbeddedSystem-Lab/TABA\_OS\_2023.git
```

Before we start,

4

- Git

```
$sudo yum install git  
$git clone https://github.com/DKU-EmbeddedSystem-Lab/TABA\_OS\_2023.git  
$cd TABA_OS_2023
```

```
inhoinno@inhoinno:~/TABA_OS_2023$ ls  
LICENSE  mycat  mycp  mycp-adv  mycreat  myls
```

- Basic file operation in Linux

- **int** open(**const char** *pathname, **int** flags, [**mode_t** mode])

- Input params : file path
 - Return : (Success) ? file descriptor (int) : -1 (error)

- **ssize_t** read(**int** fd, **void** *buf, **size_t** count)

- Input params : file descriptor
 - Return : (Success) ? the number of bytes read : -1 (error)

- **ssize_t** write(**int** fd, **void** *buf, **size_t** count)

- Input params : file descriptor
 - Return : (Success) ? the number of bytes written : -1 (error)

- **int** close(**int** fd)

- Input params : file descriptor
 - Return : (Success) ? 0 : -1

실습1: mycreat, mycat

6

- mycreat

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#define MAX_BUF 64
char fname[] = "newfile";
char dummy[] = "TABA OS 2023 [your name here]\n";
int main(int argc, char *argv[]){
    //변수 선언
    int fd, read_size, write_size = 0;
    char buf[MAX_BUF];

    fd = /* [1] fill out here using system call */;
    write_size=write( /* [2] fill out here : write argument */ );
    //error handling
    printf("write to file %s write size %d\n", fname, write_size);
    /* [3] fill out here : */
}
```

[1] \$man open

NAME [top](#)

open, openat, creat - open and possibly create a file

SYNOPSIS [top](#)

#include <fcntl.h>

int open(const char *pathname, int flags);

int open(const char *pathname, int flags, mode_t mode);

[2] \$man -s 2 write

NAME [top](#)

write - write to a file descriptor

SYNOPSIS [top](#)

#include <unistd.h>

ssize_t write(int fd, const void *buf, size_t count);

실습1: mycreat, mycat

- mycreat 수행 결과 화면

```
$ make clean; make
```

```
inhoinno@inhoinno:~/TABA_OS_2023/mycreat$ make clean; make
rm -f *.o
rm -f *.gch
rm -f mycreat
gcc -g -c -o mycreat-answer.o mycreat-answer.c
gcc -o mycreat mycreat-answer.o
```

```
$ ./mycreat; cat newfile
```

```
inhoinno@inhoinno:~/TABA_OS_2023/mycreat$ ./mycreat
write to file newfile write size 23
inhoinno@inhoinno:~/TABA_OS_2023/mycreat$ cat
answer/          Makefile          mycreat          mycreat-answer.c  mycreat-answer.o  mycreat.c          newfile
inhoinno@inhoinno:~/TABA_OS_2023/mycreat$ cat newfile
TABA OS 2023 inhoinno
inhoinno@inhoinno:~/TABA_OS_2023/mycreat$ █
```

실습1: mycreat, mycat

8

• mycat

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#define MAX_BUF 64

int main(int argc, char *argv[]){
    //변수 선언
    int fd, read_size, write_size = 0;
    char buf[MAX_BUF];
    //예외 처리
    if (argc!=2){
        printf("USAGE: %s file_name\n",argv[0]);
        exit(-1);
    }
    fd = /* [1] fill out here using system call */;
    if (fd < 0){
        //open error handling
        perror("fd open error\n");
    }
    while((read_size = /* [2] fall through. fill out here using syscall. */) != 0){
        //printf("%s",buf);
        write_size = /* [3] fall through. fill out here using syscall. */;
    }
    /* [4] fall through. fd must be closed. */;
}
```

hint

```
[ ] $man open
[ ] $man read
[ ] $man -s 2 write
[ ] $man close
```

standard file descriptors

```
/* Standard file descriptors. */
#define STDIN_FILENO 0 /* Standard input. */
#define STDOUT_FILENO 1 /* Standard output. */
#define STDERR_FILENO 2 /* Standard error output. */
```

<https://elixir.bootlin.com/uclibc-ng/latest/source/include/unistd.h#L214>

실습1: mycreat, mycat

- mycat 수행 결과 화면

```
$ cp ../mycreat/newfile .
```

```
$ make
```

```
$ ./mycat newfile
```

```
inhoinno@inhoinno:~/TABA_OS_2023/mycat$ make
```

```
gcc -g      -c -o mycat.o mycat.c
```

```
gcc -o mycat mycat.o
```

```
inhoinno@inhoinno:~/TABA_OS_2023/mycat$ ./mycat newfile
```

```
TABA OS 2023 inhoinno
```

```
inhoinno@inhoinno:~/TABA_OS_2023/mycat$ █
```

- 프로그램 요구사항 기술 - mycp

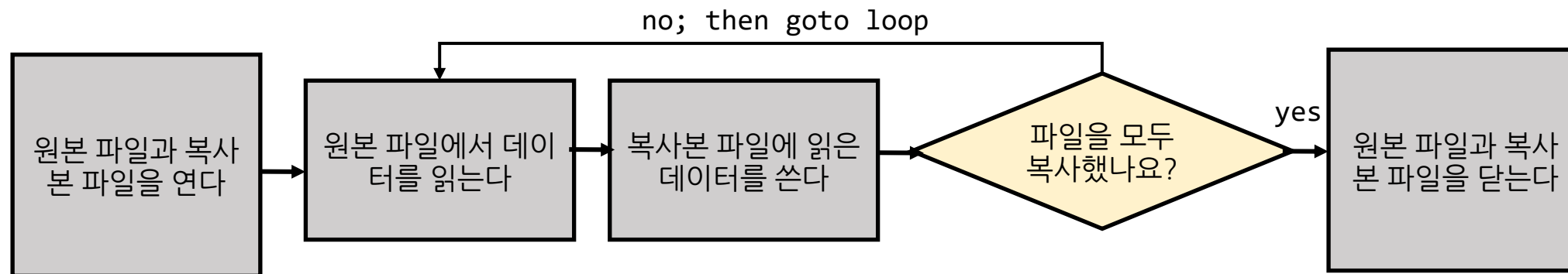
- Input

- USAGE : ./mycp origin_file_here dest_file_here
 - 내용이 적혀있는 원본파일origin file

- Output

- 원본 파일origin file 의 user data가 적혀있는 복사본 파일destination file

- mycp algorithm



- mycp

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#define MAX_BUF 64

int main(int argc, char *argv[]){
    //변수 선언
    int fd_origin, fd_dest, read_size, write_size =0;
    char buf[MAX_BUF];
    //예외 처리
    if (argc!=3){
        printf("USAGE: %s origin dest\n",argv[0]);
        exit(-1);
    }
    fd_origin = /* [1] fill out here using system call */;
    fd_dest = /* [1-1] fill out here using system call*/;
    if (fd_origin < 0 || fd_dest <0){
        //open error handling
        perror("fd open error\n");
    }
    //read from the origin file
    while((read_size = /* [2] fall through. fill out here using syscall. */) != 0){
        //write to the dest file
        write_size = /* [3] fall through. fill out here using syscall. */;
    }
    /* [4] fall through. fd must be closed. */;
}
```

hint

[]	\$man open
[]	\$man read
[]	\$man -s 2 write
[]	\$man close

- mycp 실행 화면

```
$ make
$ ls -l

inhoinno@inhoinno:~/TABA_OS_2023/mycp$ ls -l
total 20
drwxrwxr-x 2 inhoinno inhoinno 4096  3월  8 01:04 answer
-rw-rw-r-- 1 inhoinno inhoinno  806  3월  8 01:07 bak_my
-rw-rw-r-- 1 inhoinno inhoinno  219  3월  8 01:04 Makefile
-rw-rw-r-- 1 inhoinno inhoinno 1074  3월  8 01:07 mycp.c
-rw-r--r-- 1 inhoinno inhoinno   29  3월  8 00:55 origin

$ ./mycp origin dest; cat dest
inhoinno@inhoinno:~/TABA_OS_2023/mycp$ cat dest
mycpprogram
I am origin file
inhoinno@inhoinno:~/TABA_OS_2023/mycp$ █
```

System Programming & OS 실습

mycreat, mycat, mycp

- 용어 정의 terminology

- 파일 file

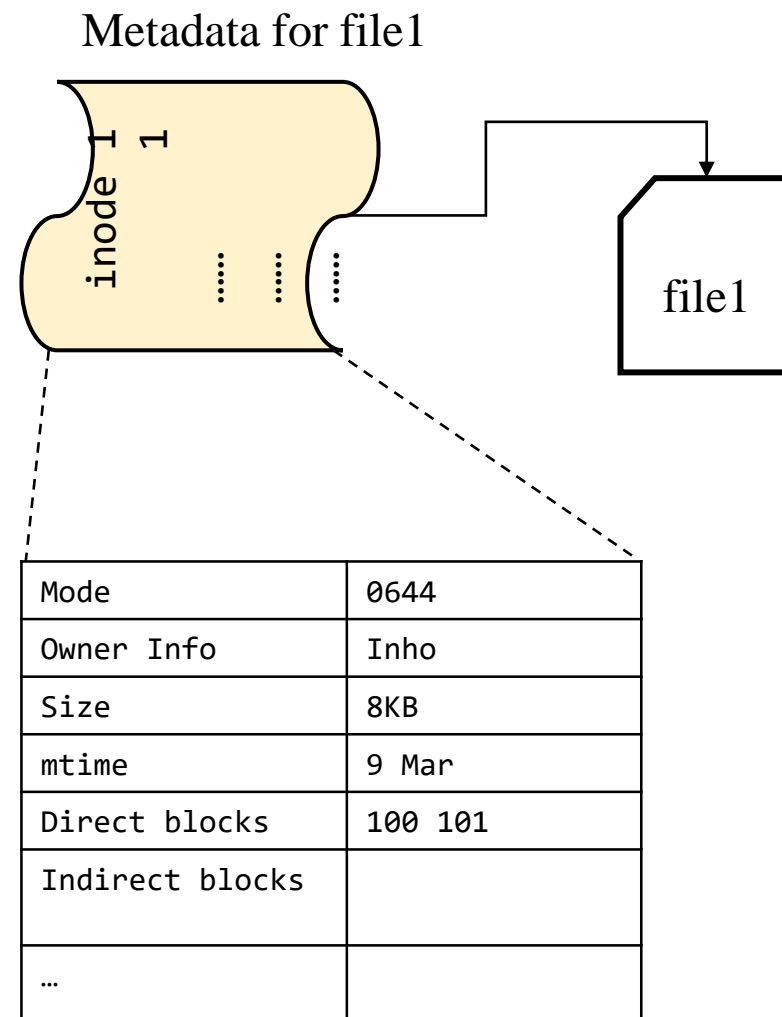
- 1. user data

- 실제 파일에 저장되는 데이터

- 2. Metadata

- File의 정보와 user data의 위치를 가리키는 데이터

→ 데이터와 메타데이터의 차이는?



- 프로그램 요구사항 기술 - mycp advanced
 - Input
 - USAGE : ./mycp origin_file_here dest_file_here
 - 내용이 적혀있는 원본파일origin file
 - Output
 - 원본 파일origin file 의 user data가 적혀있는 복사본 파일destination file
 - 이때 파일의 속성 정보를 포함하여 완전 복사 (Metadata copy)

- mycp

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#define MAX_BUF 64

int main(int argc, char *argv[]){
    int fd_origin, fd_dest, read_size, write_size =0;
    char buf[MAX_BUF];

    struct stat* stat_origin=(struct stat*)malloc(sizeof(struct stat));
    mode_t flag_origin;

    if (argc!=3){
        printf("USAGE: %s origin dest\n",argv[0]);
        exit(-1);    }
    fd_origin    = open(argv[1], O_RDONLY);

    /* [1] fall through. get file attribute structure from fd_origin */
    flag_origin = stat_origin-> /* [2] fall through. let's get member from struct stat "stat_origin->field_here" */;

    fd_dest      = open(argv[2], O_RDWR|O_CREAT|O_EXCL|O_SYNC, flag_origin);
    //open error handling
    //read from the origin file
    while((read_size = read(fd_origin, buf, MAX_BUF))!= 0){
        //write to the dest file
        write_size = write(fd_dest, buf, read_size);
    }
    close(fd_origin); close(fd_dest);
}
```

hint

[1] \$man fstat

- mycp 결과 화면

before \$ ls -l

```
inhoinno@inhoinno:~/TABA_OS_2023/mycp-adv$ ls -l
total 52
-rw-rw-r-- 1 inhoinno inhoinno  226  3월  7 23:54 Makefile
-rwxrwxr-x 1 inhoinno inhoinno 20872  3월  7 23:55 mycp
-rw-rw-r-- 1 inhoinno inhoinno  1162  3월  7 23:54 mycp-answer.c
-rw-rw-r-- 1 inhoinno inhoinno  9320  3월  7 23:55 mycp-answer.o
-rw-rw-r-- 1 inhoinno inhoinno   894  3월  7 23:16 mycp.c
-rw-r--r-- 1 inhoinno inhoinno    30  3월  7 23:22 origin
```

\$ cat origin

```
inhoinno@inhoinno:~/TABA_OS_2023/mycp-adv$ cat origin
mycp program
I am origin file
```

after \$ make

\$ ls -l

```
inhoinno@inhoinno:~/TABA_OS_2023/mycp-adv$ ./mycp origin dest
flag origin 100644
```

inhoinno@inhoinno:~/TABA_OS_2023/mycp-adv\$ ls -l

```
total 56
drwxrwxr-x 2 inhoinno inhoinno  4096  3월  8 00:04 answer
-rw-r--r-- 1 inhoinno inhoinno    30  3월  8 00:05 dest
-rw-rw-r-- 1 inhoinno inhoinno   219  3월  8 00:00 Makefile
-rwxrwxr-x 1 inhoinno inhoinno 20872  3월  8 00:04 mycp
-rw-rw-r-- 1 inhoinno inhoinno  1072  3월  8 00:04 mycp.c
-rw-rw-r-- 1 inhoinno inhoinno  9232  3월  8 00:04 mycp.o
-rw-r--r-- 1 inhoinno inhoinno    30  3월  7 23:22 origin
```

\$ cat dest

```
inhoinno@inhoinno:~/TABA_OS_2023/mycp-adv$ cat dest
mycp program
I am origin file
```

- 용어 정의 terminology

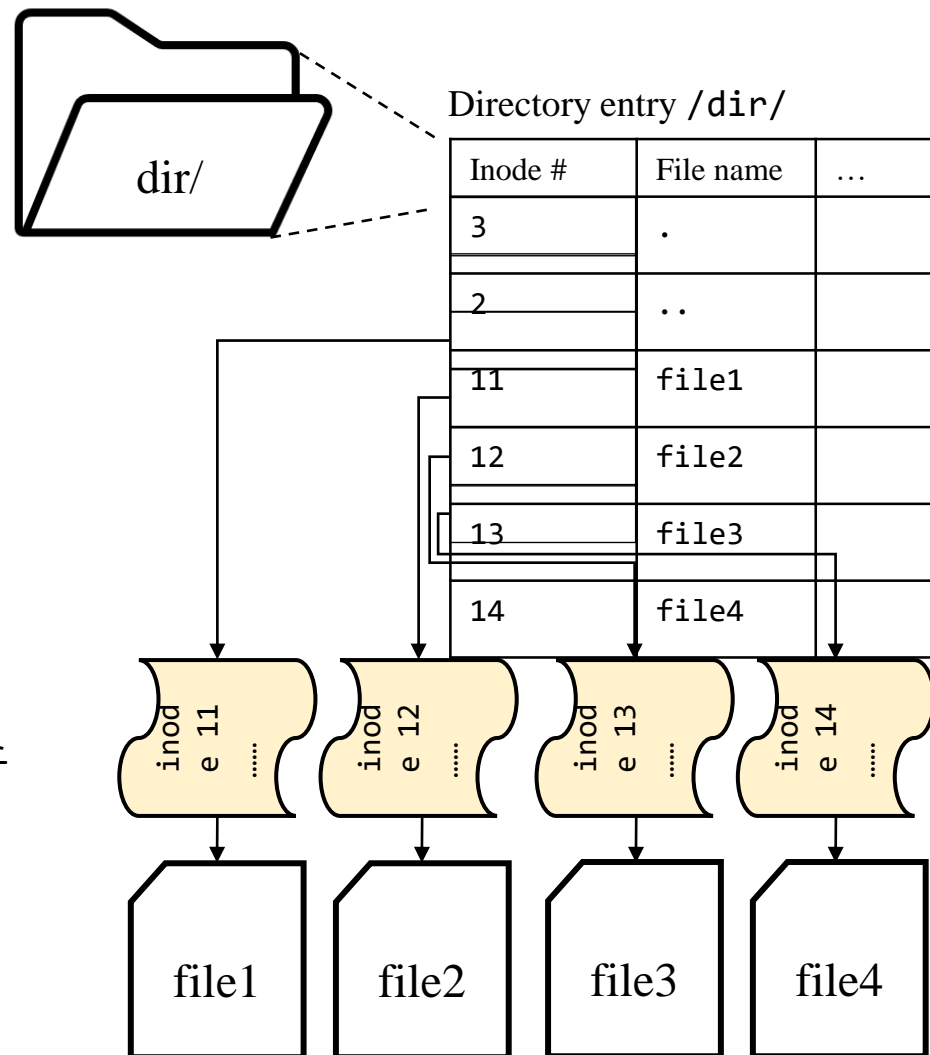
- 파일 file

1. User data
 - 실제 파일에 저장되는 데이터
2. Metadata
 - File의 정보와 user data의 위치를 가리키는 데이터

- 디렉토리 directory

1. User data
 - Directory entry
- :디렉토리 내 파일 및 하위 디렉토리를 가리키는 자료구조

Ⓢ Linux 에서 일반 파일과 비교했을 때 디렉토리가 가지는 특별한 차이점은?



- myls

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <dirent.h>
#include <sys/types.h>
int main(int argc, char *argv[]){
    //변수 선언
    DIR *dir=NULL;
    struct dirent* dentry=NULL;
    char *dir_name=".";
    //예외 처리
    if (argc == 1){ // args 없는 경우 현재 디렉토리 "." 내용을 보여줌.
        dir = opendir(dir_name);
    }
    else if (argc == 2){
        dir_name = argv[1]; //warning.
        /* [1] fall through. fill out here using directory syscall. */
    }else {printf("argc %d : We only accept 1 or 2 args for now\n", argc);
        exit(-1); }

    while((dentry =/* [2] using dir syscall*/)!=NULL){
        printf("%s \n", dentry->d_name);
    }
    /* [3] fall through. close directory here. */;
}
```

hint

```
[ ] $man opendir
[ ] $man readdir
[ ] $man closedir
```

struct dirent

```
struct linux_dirent64 {
    u64                d_ino;
    s64                d_off;
    unsigned short     d_reclen;
    unsigned char      d_type;
    char               d_name[];
};
```

<https://elixir.bootlin.com/linux/v5.15.98/source/include/linux/dirent.h>

- myls 결과 화면

```
$ make  
$ ./mysls
```

```
inhoinno@inhoinno:~/TABA_OS_2023/mysls$ ./mysls  
mysls.c  
mysls  
..  
mysls.o  
.  
Makefile
```

```
$ ./mysls ..
```

```
inhoinno@inhoinno:~/TABA_OS_2023/mysls$ ./mysls ..  
mycp  
mysls  
mycp-adv  
LICENSE  
..  
mycat  
mycreat  
.git  
.
```

- creat()
- mkdir(), readdir(), rmdir()
- pipe()
- mknod()
- link(), unlink()
- dup(), dup2()
- stat(), fstat()
- chmod(), fchmod()
- ioctl(), fcntl()
- Sync(), fsync()