

System Programming & OS 실습

A4. myCP, myLs

최민국, 정지현, 안석현, 김선재

Dankook University

{mgchoi, wlgjsjames7224, seokhyun, rlatjswo0824}@dankook.ac.kr

Index

❖ mycp

❖ mycp_adv

❖ myls

mycp

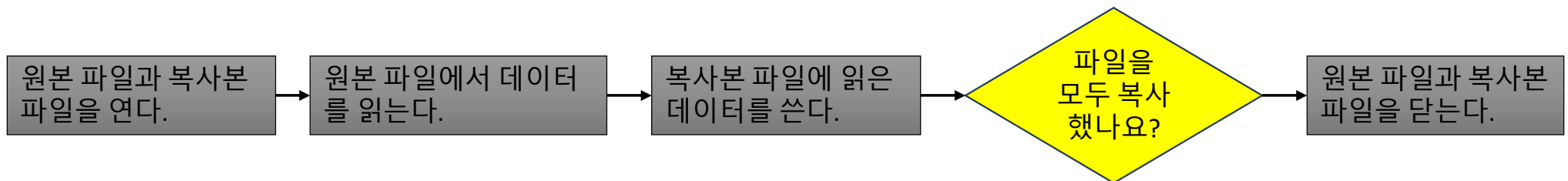
❖ Input

- Usage: ./mycp [*origin file*] [*destination file*]

❖ Output

- 원본 파일 origin file 의 user data가 적혀 있는 복사본 파일 destination file 생성

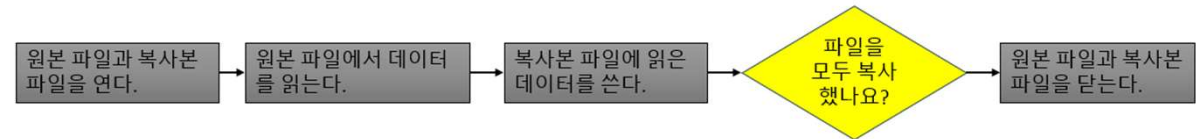
❖ mycp algorithm



mycp

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <fcntl.h>
5 #include <errno.h>
6 #define MAX_BUF 64
7
8 int main(int argc, char *argv[]){
9     int fd_origin, fd_dest, read_size, write_size = 0;
10    char buf[MAX_BUF];
11
12    if (argc!=3){
13        printf("USAGE: %s origin dest\n",argv[0]);
14        exit(-1);
15    }
16
17    fd_origin = [1] fill out here using system call ;
18    fd_dest = [1-1] fill out here using system call ;
19    if (fd_origin < 0 || fd_dest < 0){
20        //open error handling
21        perror("fd open error\n");
22    }
23
24    //read from the origin file
25    while((read_size = [2] fill out here using system call ) != 0 ){
26        //write to the dest file
27        write_size = [3] fill out here using system call ;
28    }
29
30    [4] fill out here using system call
31
32 }
```

❖ mycp algorithm



Using System call

open
read
write
close

mycp

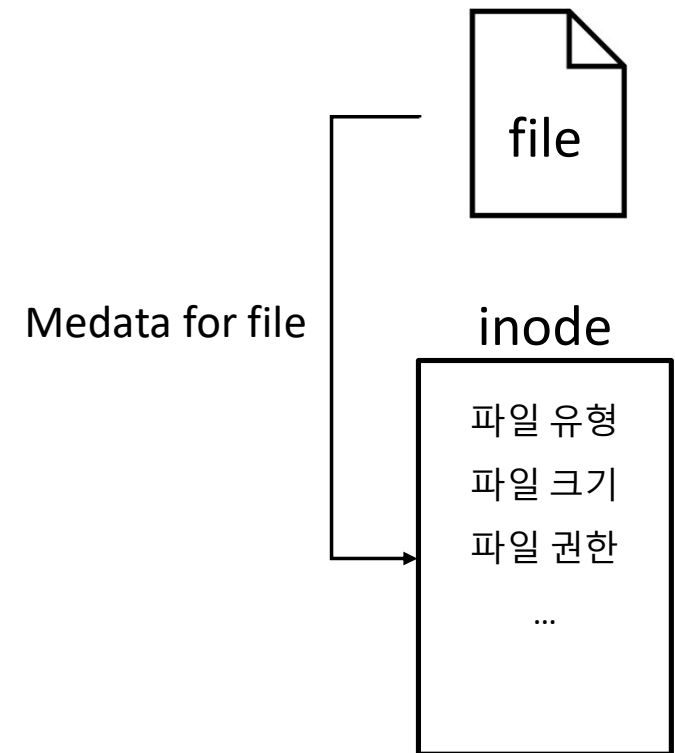
❖ mycp 실행화면

- cat [file]
- 파일의 데이터 비교

```
[ec2-user@ip-172-31-15-105 taba9]$ ls -l
total 28
-rwxr-xr-x. 1 ec2-user ec2-user 17744 Aug 27 17:49 mycp
-rw-r--r--. 1 ec2-user ec2-user  766 Aug 27 18:06 mycp.c
-rw-r--r--. 1 ec2-user ec2-user   17 Aug 27 18:24 origin
[ec2-user@ip-172-31-15-105 taba9]$ ./mycp origin dest
[ec2-user@ip-172-31-15-105 taba9]$ cat dest
I am origin file
[ec2-user@ip-172-31-15-105 taba9]$ cat origin
I am origin file
[ec2-user@ip-172-31-15-105 taba9]$
```

❖ 파일 file

- User data
 - 실제 파일에 저장되는 데이터
- Metadata
 - File의 정보와 user data의 위치를 가리키는 데이터
 - 파일 유형, 파일 크기, 파일 권한...



mycp_adv

❖ Input

- Usage: ./mycp_adv [*origin file*] [*destination file*]

❖ Output

- 원본 파일 origin file 의 user data가 적혀 있는 복사본 파일 destination file 생성
- 파일의 속성 정보 (Metadata)를 포함한 완전 복사

mycp_adv

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <fcntl.h>
5 #include <errno.h>
6 #include <sys/stat.h>
7 #define MAX_BUF 64
8
9 int main(int argc, char *argv[]){
10     int fd_origin, fd_dest, read_size, write_size = 0;
11     char buf[MAX_BUF];
12     struct stat* stat_origin = (struct stat*)malloc(sizeof(struct stat));
13     mode_t flag_origin;
14
15     if (argc != 3){
16         printf("USAGE: %s origin dest\n", argv[0]);
17         exit(-1);
18     }
19     fd_origin = open(argv[1], O_RDONLY);
20
21     [1] get file attribute structure from fd_origin
22
23     flag_origin = stat_origin-> [2] let's get member from struct stat "stat_origin->field_here"
24
25     fd_dest = open(argv[2], O_RDWR | O_CREAT | O_EXCL | O_SYNC, flag_origin);
26     while((read_size = read(fd_origin, buf, MAX_BUF)) != 0){
27         write_size = write(fd_dest, buf, read_size);
28     }
29     close(fd_origin);
30     close(fd_dest);
31 }
```

Using System call

fstat

mycp_adv

❖ mycp_adv 실행화면

- cat [file]
- 파일의 데이터 비교

```
[ec2-user@ip-172-31-15-105 taba9]$ ./mycp_adv origin dest
[ec2-user@ip-172-31-15-105 taba9]$ ls -l
total 56
-rw-r--r--. 1 ec2-user ec2-user    17 Aug 27 20:23 dest
-rwxr-xr-x. 1 ec2-user ec2-user 17744 Aug 27 17:49 mycp
-rwxr-xr-x. 1 ec2-user ec2-user 17792 Aug 27 20:23 mycp_adv
-rw-r--r--. 1 ec2-user ec2-user   805 Aug 27 20:18 mycp_adv.c
-rw-r--r--. 1 ec2-user ec2-user   766 Aug 27 18:06 mycp.c
-rw-r--r--. 1 ec2-user ec2-user    17 Aug 27 18:24 origin
[ec2-user@ip-172-31-15-105 taba9]$ cat origin
I am origin file
[ec2-user@ip-172-31-15-105 taba9]$ cat dest
I am origin file
```

mycp-adv

❖ stat [file]

- 파일의 메타데이터 비교

```
[ec2-user@ip-172-31-15-105 taba9]$ stat origin
  File: origin
  Size: 17          Blocks: 8          IO Block: 4096   regular file
Device: 10302h/66306d Inode: 8413986      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/ec2-user)   Gid: ( 1000/ec2-user)
Context: unconfined_u:object_r:user_home_t:s0
Access: 2024-08-27 18:24:23.238560580 +0000
Modify: 2024-08-27 18:24:12.954589960 +0000
Change: 2024-08-27 18:24:12.954589960 +0000
 Birth: 2024-08-27 18:24:12.954589960 +0000
[ec2-user@ip-172-31-15-105 taba9]$ stat dest
  File: dest
  Size: 17          Blocks: 8          IO Block: 4096   regular file
Device: 10302h/66306d Inode: 8413988      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/ec2-user)   Gid: ( 1000/ec2-user)
Context: unconfined_u:object_r:user_home_t:s0
Access: 2024-08-27 20:24:06.411915356 +0000
Modify: 2024-08-27 20:23:42.476984046 +0000
Change: 2024-08-27 20:23:42.476984046 +0000
 Birth: 2024-08-27 20:23:42.476984046 +0000
```

myls

❖ Input

- Usage: ./mysl *[argument]*
- [argument]는 디렉토리

*선택

❖ Output

- 디렉토리 내의 파일과 디렉토리 목록을 보여줌
- 1. ./mysl
 - 현재 디렉토리 내의 파일과 디렉토리 목록을 보여줌
- 2. ./mysl [argument]
 - argument에 해당하는 디렉토리 내의 파일과 디렉토리 목록을 보여줌

myls

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <fcntl.h>
5 #include <errno.h>
6 #include <dirent.h>
7 #include <sys/types.h>
8
9 int main(int argc, char *argv[]) {
10 // 변수 선언
11     DIR *dir = NULL;
12     struct dirent* dentry = NULL;
13     char *dir_name = ".";
14
15     // 예외 처리
16     if (argc == 1) { // args 없는 경우 현재 디렉토리 "." 내용을 보여줌.
17         dir = opendir(dir_name);
18     } else if (argc == 2) {
19         dir_name = argv[1]; // 디렉토리 이름을 argv에서 가져옴
20         dir = [1] fill out here using directory syscall
21     } else {
22         printf("argc %d : We only accept 1 or 2 args for now\n", argc);
23         exit(-1);
24     }
25     while ((dentry = [2] using dir syscall) != NULL) {
26         printf("%s \n", dentry->d_name); // 디렉토리 항목 이름을 출력
27     }
28     [3] using dir syscall
29 }
```

Using System call

opendir
readdir
closedir

myls

❖ myls 실행화면

- 인자(argv) 0개와 1개 비교

```
[ec2-user@ip-172-31-15-105 taba9]$ ./myls
```

```
.  
..  
mycp  
mycp.c  
origin  
mycp_adv.c  
mycp_adv  
dest  
myls.c  
myls
```

```
[ec2-user@ip-172-31-15-105 taba9]$ ./myls ..
```

```
.  
..  
.bash_logout  
.bash_profile  
.ssh  
.bash_history  
.vimrc  
.bashrc  
day2  
taba7  
taba9  
.viminfo
```