

Report 3



과 목 명	시스템프로그래밍
학 번	32202546
성 명	안지성
담당교수	최종무 교수님
제 출 일	2021.10.15

1. 기본 복사 mycp.c와 접근권한까지 복사 mycp_bonus.c 생성

```
sys32202546@embedded:~/HW3$ vi mycp.c
sys32202546@embedded:~/HW3$ vi mycp_bonus.c
sys32202546@embedded:~/HW3$ ls -l
total 12
-rw-rw-r-- 1 sys32202546 sys32202546 15 10월 15 16:05 dal.txt
-rw-rw-r-- 1 sys32202546 sys32202546 1140 10월 15 17:10 mycp_bonus.c
-rw-rw-r-- 1 sys32202546 sys32202546 924 10월 15 17:09 mycp.c
```

2. 실행파일 mycp 사용

```
sys32202546@embedded:~/HW3$ gcc -o mycp mycp.c
sys32202546@embedded:~/HW3$ gcc -o mycp_bonus mycp_bonus.c
sys32202546@embedded:~/HW3$ ./mycp
USAGE: ./mycp source_file destination_file
sys32202546@embedded:~/HW3$ ./mycp unicorn.txt po.txt
Can't open unicorn.txt
sys32202546@embedded:~/HW3$ ./mycp dal.txt hyang.txt
sys32202546@embedded:~/HW3$ ls -l
total 32
-rw-rw-r-- 1 sys32202546 sys32202546 15 10월 15 16:05 dal.txt
-r--r--r-- 1 sys32202546 sys32202546 15 10월 15 17:11 hyang.txt
-rwxrwxr-x 1 sys32202546 sys32202546 5584 10월 15 17:10 mycp
-rwxrwxr-x 1 sys32202546 sys32202546 5880 10월 15 17:11 mycp_bonus
-rw-rw-r-- 1 sys32202546 sys32202546 1140 10월 15 17:10 mycp_bonus.c
-rw-rw-r-- 1 sys32202546 sys32202546 924 10월 15 17:09 mycp.c
sys32202546@embedded:~/HW3$ cat dal.txt
unicorn heart!
sys32202546@embedded:~/HW3$ cat hyang.txt
unicorn heart!
```

- ./mycp -> mycp 뒤에 파일 2개 쓰도록 메시지 출력
 - ./mycp unicorn.txt po.txt -> 복사하는 파일인 unicorn.txt가 없어 열 수 없음을 출력
 - ./mycp dal.txt hyang.txt -> dal.txt 내용을 hyang.txt로 복사
- > cat dal.txt와 cat hyang.txt를 하면 두 파일의 내용이 같음을 확인!
- > 현재는 복사한 파일과 대상 파일이 접근권한이 같지 않음을 알 수 있다!

3. 실행파일 mycp_bonus 사용

```
sys32202546@embedded:~/HW3$ rm hyang.txt
rm: remove write-protected regular file 'hyang.txt'? y
sys32202546@embedded:~/HW3$ ls
dal.txt  mycp  mycp_bonus  mycp_bonus.c  mycp.c
sys32202546@embedded:~/HW3$ ./mycp_bonus dal.txt hyang.txt
sys32202546@embedded:~/HW3$ cat hyang.txt
unicorn heart!
sys32202546@embedded:~/HW3$ ls -l
total 32
-rw-rw-r-- 1 sys32202546 sys32202546  15 10월  15 16:05 dal.txt
-rw-rw-r-- 1 sys32202546 sys32202546  15 10월  15 17:13 hyang.txt
-rwxrwxr-x 1 sys32202546 sys32202546 5584 10월  15 17:10 mycp
-rwxrwxr-x 1 sys32202546 sys32202546 5880 10월  15 17:11 mycp_bonus
-rw-rw-r-- 1 sys32202546 sys32202546 1140 10월  15 17:10 mycp_bonus.c
-rw-rw-r-- 1 sys32202546 sys32202546  924 10월  15 17:09 mycp.c
sys32202546@embedded:~/HW3$ whoami
sys32202546
sys32202546@embedded:~/HW3$ date
2021. 10. 15. (금 ) 17:13:50 KST
```

- ./mycp dal.txt hyang.txt -> dal.txt 내용을 hyang.txt로 복사
- > cat dal.txt와 cat hyang.txt를 하면 두 파일의 내용이 같음을 확인!
- > mycp_bonus를 사용하니 mycp와 다르게 dal.txt와 hyang.txt가 접근권한이 같음을 확인!

4. mycp.c

```
/* mycp program, by jisung. dalssagi00@gmail.com */
#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 fd1, fd2, read_size;
    char buff[MAX_BUF];

    if(argc != 3){
        // 3개의 인자인지 검사
        printf("USAGE: %s source_file destination_file\n", argv[0]);
        exit(-1);
    }
    fd1 = open(argv[1], O_RDONLY);
    if(fd1 < 0){
        // error handling
        printf("Can't open %s\n", argv[1]);
        exit(-1);
    }

    fd2 = open(argv[2], O_WRONLY | O_CREAT | O_EXCL);
    if(fd2 < 0){
        // error handling
        printf("Can't create %s file with errno %d\n", argv[2], errno);
        exit(-1);
    }

    while(1){
        // read_size를 0으로 write ***
        read_size = read(fd1, buff, MAX_BUF);
        if(read_size == 0)
            break;
        write(fd2, buff, read_size);
    }
    close(fd1);
    close(fd2);
}
```

```
/* mycp program, by jisung. dalssagi00@gmail.com */
#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 fd1, fd2, read_size;
    char buff[MAX_BUF];

    if(argc != 3){
        // 3개의 인자인지 검사
        printf("USAGE:      %s      source_file      destination_file\n",
```

```

argv[0]);
        exit(-1);
    }
    fd1 = open(argv[1], O_RDONLY);
    if(fd1<0){
        // error handling
        printf("Can't open %s\n", argv[1]);
        exit(-1);
    }

    fd2 = open(argv[2], O_WRONLY | O_CREAT | O_EXCL);
    if(fd2 < 0){
        // error handling
        printf("Can't create %s file with errno %d\n", argv[2],
errno);
        exit(-1);
    }
    while(1){
        // read_size만큼 write 실행
        read_size = read(fd1, buff, MAX_BUF);
        if(read_size == 0)
            break;
        write(fd2, buff, read_size);
    }
    close(fd1);
    close(fd2);
}

```

```

/* mycp program, by jisung. dalssagi00@gmail.com */
#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 fd1, fd2, read_size, info;
    char buff[MAX_BUF];
    struct stat s;

    if(argc != 3){
        printf("USAGE: %s source_file destination_file\n", argv[0]);
        exit(-1);
    }
    fd1 = open(argv[1], O_RDONLY);
    info = stat(argv[1], &s);
    if(fd1 < 0){
        printf("Can't open %s\n", argv[1]);
        exit(-1);
    }

    fd2 = open(argv[2], O_WRONLY | O_CREAT | O_EXCL, s.st_mode & (S_IRWXU | S_IRWXG | S_IRWXO));
    if(fd2 < 0){
        printf("Can't create %s file with errno %d\n", argv[2], errno);
        exit(-1);
    }
    while(1){
        read_size = read(fd1, buff, MAX_BUF);
        if(read_size == 0)
            break;
        write(fd2, buff, read_size);
    }
    close(fd1);
    close(fd2);
}

```

```

/* mycp program, by jisung. dalssagi00@gmail.com */
#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 fd1, fd2, read_size, info;
    char buff[MAX_BUF];
    struct stat s;

    if(argc != 3){
        printf("USAGE: %s source_file destination_file\n",
argv[0]);
        exit(-1);
    }
}

```

```

    fd1 = open(argv[1], O_RDONLY);
    info = stat(argv[1], &s);
    if(fd1<0){
        // error handling
        printf("Can't open %s\n", argv[1]);
        exit(-1);
    }

    fd2 = open(argv[2], O_WRONLY | O_CREAT | O_EXCL, s.st_mode &
(S_IRWXU
    | S_IRWXG
    | S_IRWXO));
    // 접근권한 정보를 복사
    if(fd2 < 0){
        // error handling
        printf("Can't create %s file with errno %d\n", argv[2],
errno);
        exit(-1);
    }
    while(1){
        // read_size만큼 write 실행
        read_size = read(fd1, buff, MAX_BUF);
        if(read_size == 0)
            break;
        write(fd2, buff, read_size);
    }
    close(fd1);
    close(fd2);
}

```

기존에 잘 사용 해 보지 못하던 open(), read(), write()함수들을 여러 flag들을 알아보고, 사용 해 보면서 필요한 flag를 사용하는 재미를 느꼈습니다. 또한 접근권한을 옮겨오는 stat에 대한 내용도 잘 알지 못하였는데 과제를 수행하면서 이해를 높인 것 같아 좋았습니다. 전반적으로 프로그램의 구현과 실행이 어떻게 돌아가는지 일부를 느낄 수 있었고, 리눅스를 이용하여 구동하니 참 즐거운 실습이었습니다. 감사합니다!