



과목명	시스템 프로그래밍
담당교수	최종무 교수님
학과	소프트웨어학과
학번	32200185
이름	곽다은
제출일자	2021.10.15

1. mycp.c 소스코드

```
/*mycp program written by daeunkwak.misodaeun3@naver.com*/
#include <stdlib.h>
#include <unistd.h> // POSIX 운영 체제 API를 제공하는 헤더파일
#include <fcntl.h> // 리눅스 시스템에서 열려진 파일의 속성을 가져오거나 설정할 때 사용하는 헤더파일
#include <errno.h> // 오류를 감지하여 errno에 저장되는 기능을 제공하는 헤더파일
#include <sys/types.h>
#include <sys/stat.h> // struct stat 헤더파일
#define MAX_BUF 64

int main(int argc, char *argv[]) {
    int org_fd, des_fd, read_size, infoo;
    char buf[MAX_BUF];
    struct stat st;

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

    org_fd = open(argv[1], O_RDONLY);
    infoo = fstat(org_fd, &st);
    // infoo = st.st_mode;
    if (org_fd < 0) { // file system error handling
        printf("Can't open! %s\n", argv[1]);
        exit(-1);
    }

    // des_fd open
    des_fd = open(argv[2], O_WRONLY | O_CREAT | O_EXCL, st.st_mode &(S_IRWXU |
    S_IRWXG | S_IRWXO));
    if (des_fd < 0) {
        printf("Can't create %s file with errno %d\n", argv[2], errno);
        exit(-1);
    }

    // read_size만큼 org_fd내용을 des_fd에 write
    while(1) {
        read_size = read(org_fd, buf, MAX_BUF);
        if (read_size == 0) {
            break;
        }
        write(des_fd, buf, read_size);
    }
    close(org_fd);
    close(des_fd);
}
```

- mycp

수업시간에 배웠던 mycat프로그램을 상당부분 참고하여 작성한 mycp 프로그램이다. Org_fd(원본 파일)을 O_RDONLY 옵션으로 읽어들였고, des_fd(복사 파일)은 읽고 쓸 수 있게, O_EXCL옵션을 통해 파일이 이미 존재한다면 생성 실패 및 에러메세지를 출력하도록 했다. 마지막으로 while loop를 통해 파일내용을 한꺼번에 읽어 des_fd에 복사했다.

- 파일 속성 복사

수업시간에 언급되었던 stat을 사용하여 파일 속성을 복사하였다. <sys/stat.h> 헤더파일을 include하고, struct stat을 정의하여 사용하였다.

2. 리눅스 실행결과

```
sys32200185@embedded:~/report3$ vi mycp.c
sys32200185@embedded:~/report3$ gcc -o mycp mycp.c
sys32200185@embedded:~/report3$ ./mycp
USAGE : ./mycp source_file destination_file
sys32200185@embedded:~/report3$ ./mycp systempro.txt systempro5.txt
sys32200185@embedded:~/report3$ ls -l
total 32
-rwxrwxr-x 1 sys32200185 sys32200185 5864 10월 15 16:42 mycp
-rw-rw-r-- 1 sys32200185 sys32200185 1353 10월 15 16:42 mycp.c
-r----- 1 sys32200185 sys32200185 18 10월 15 16:24 systempro2.txt
----- 1 sys32200185 sys32200185 18 10월 15 16:34 systempro3.txt
----- 1 sys32200185 sys32200185 18 10월 15 16:40 systempro4.txt
-rw-rw-r-- 1 sys32200185 sys32200185 18 10월 15 16:43 systempro5.txt
-rw-rw-r-- 1 sys32200185 sys32200185 18 10월 15 16:19 systempro.txt
sys32200185@embedded:~/report3$ more systempro.txt
I love systempro!
sys32200185@embedded:~/report3$ more systempro5.txt
I love systempro!
sys32200185@embedded:~/report3$ whoami
sys32200185
sys32200185@embedded:~/report3$ date
2021. 10. 15. (금) 16:45:31 KST
sys32200185@embedded:~/report3$
```

접근권한까지 모두 복사에 성공한 파일은 systempro -> systempro5.txt이다.

3. Discussion

Systempro.txt, systempro5.txt를 제외한 다른 파일들은 모두 수행 중 시행착오를 겪은 파일들이다.

현재 mycp에서는 `infoo = fstat(org_fd, &st)`를 사용하여 `fstat`정보를 `infoo` 변수에 대입하여 작성했는데, 처음에는 `infoo = st.st_mode`로 파일속성을 저장하고, `des_fd = open(argv[2], O_WRONLY | O_CREAT | O_EXCL, infoo);` 코드를 통해 파일속성 저장을 시도했었다. 하지만 `ls -l` 결과에서 많은 `-----`를 보면 알 수 있듯, 이 코드는 파일 속성까지 복사하지 못한 코드이다. `st_mode`를 통한 파일 속성을 얻으려면 `st_mode & S_IFREG`와 같이 간단한 비트 연산을 통해 정보들을 얻어야 한다. 따라서 `des_fd`를 `open`할 때 `st.st_mode & (S_IRWXU | S_IRWXG | S_IRWXC)`를 통해 접근권한까지 복사할 수 있었다. 배운 내용을 리눅스 서버에서 직접 활용하니 이론으로만 공부하는것보다 훨씬 와닿았고, `stat`같은 system call들을 직접 적용해볼 수 있어서 재미있었다.