

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 문한나

mhn97@naver.com

리눅스 시스템 프로그래밍

```
mhn@mhn-Z20NH-AS51B5U: ~/linux/19/test
1 #include <unistd.h>
2 #include <fcntl.h>
3
4 int main(void){
5     int fdin,fdout;
6     char buf[1024];
7     ssize_t nread;
8
9     fdin = open("t1.txt",O_RDONLY);
10    fdout = open("t2.txt",O_CREAT | O_WRONLY);
11
12    while((nread = read(fdin,buf,1024)){
13        if(write(fdout,buf,nread) < nread){
14            close(fdin);
15            close(fdout);
16        }
17    }
18    close(fdin);
19    close(fdout);
20
21 }
22 }
```

t2이라는 이름의 txt파일을 만든다

O_CREAT : 파일 생성

O_RDONLY : 읽기 전용

O_WRONLY : 쓰기 전용

O_RDWR : 읽고 쓰기

O_EXCL : 파일이 있으면 생성x

O_TRUNC : 새로 갱신

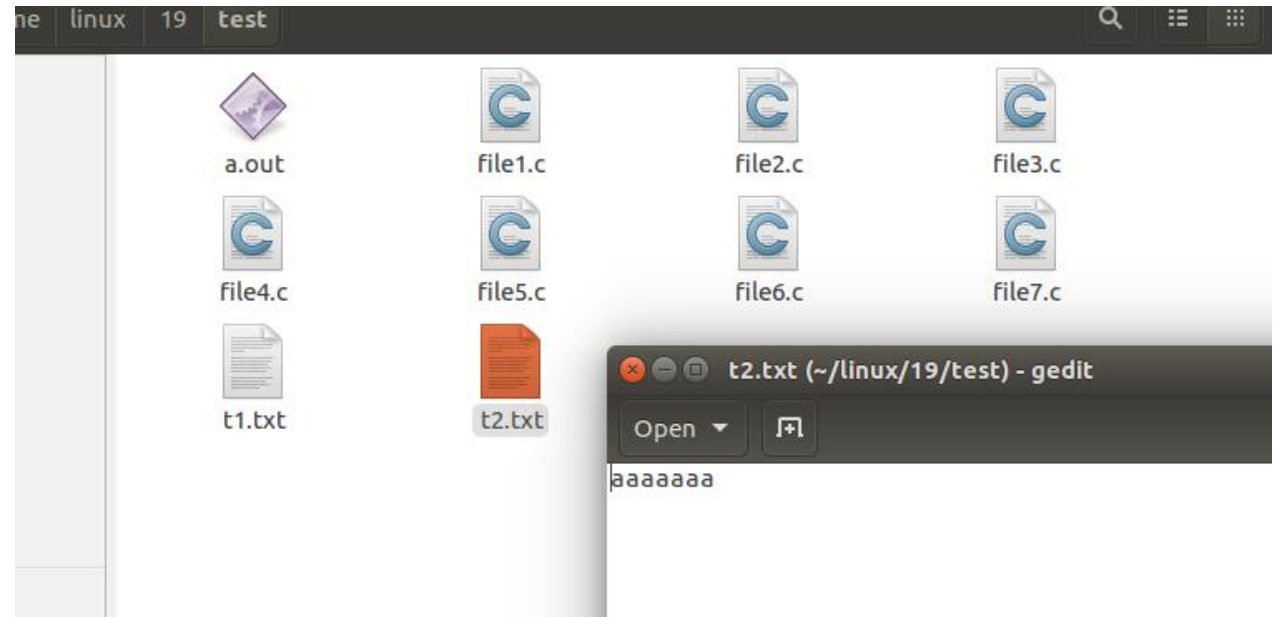
1024바이트만큼 fdin(t1.txt)을 읽어서
읽은 만큼 fdout(t2.txt)에 쓸 것이다

open과 close는 붙어 다닌다

t1.txt

```
mhn@mhn-Z20NH-AS51B5U: ~/linux/19/test
1 aaaaaaa
```

t2.txt



t2.txt 파일이 생성되면서 값이 복사된다

```

mhn@mhn-Z20NH-AS51B5U: ~/linux/19/test
1 #include <stdio.h>
2 #include <fcntl.h>
3 #include <unistd.h>
4
5 int main(void){
6
7 int file1;
8 off_t num;
9
10 file1 = open("t1.txt",O_RDONLY);
11 num = lseek(file1,(off_t)0,SEEK_END);
12
13 printf("%ld\n",num);
14
15
16
17
18 return 0;
19
20 }
~
~
~
~

```

File1을 0부터 끝까지 읽은 후
순회한 바이트 수(파일 사이즈)를 리턴

t1.txt

```

mhn@mhn-Z20NH-AS51B5U: ~/linux/19/test
1 aaaaaaa
~
~
~
~

```

```

mhn@mhn-Z20NH-AS51B5U:~/linux/19/test$ vi file3.c
mhn@mhn-Z20NH-AS51B5U:~/linux/19/test$ gcc file3.c
mhn@mhn-Z20NH-AS51B5U:~/linux/19/test$ ./a.out
8

```

글자 +1만큼 반환됨

```

mhn@mhn-Z20NH-AS51B5U: ~/linux/19/test
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <stdlib.h>
4 #include <fcntl.h>
5
6 int main(int argc, char **argv){
7
8     int fd, ret;
9     char buf[1024];
10    if(argc != 2){
11        printf("filename\n");
12        exit(-1);
13    }
14
15    fd = open(argv[1], O_RDONLY);
16    while(ret = read(fd, buf, sizeof(buf))){
17        write(1, buf, ret); // 1 = 표준출력!!!
18    }
19    close(fd);
20    return 0;
21
22 }

```

입력을 2개 하라는 경고

2번째 입력한 파일을 출력

0 : 표준입력

1 : 표준출력

2 : 에러

*System call(S/W interrupt)

유저가 커널에 요청하면 커널이 처리한 후 응답해준다.

보안을 위해 파일 포인터 인덱스를 넘겨줌

```
mhn@mhn-Z20NH-AS51B5U:~/linux/19/test$ ./a.out file4.c
```

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <fcntl.h>
```

```
int main(int argc, char **argv){
```

```
    int fd, ret;
    char buf[1024];
    if(argc != 2){
        printf("filename\n");
        exit(-1);
    }
```

```
    fd = open(argv[1], O_RDONLY);
    while(ret = read(fd, buf, sizeof(buf))){
        write(1, buf, ret); // 1 = 표준출력!!!
    }
    close(fd);
    return 0;
```

```
}
```

```
mhn@mhn-Z20NH-AS51B5U:~/linux/19/test$
```

2번째 입력한 file4.c를 보여준다

mhn@mhn-900X3L: ~/my_proj/c/18_s/study

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

int main(int argc, char **argv){

    int fdin, fdout;
    char buf[1024]={0};
    ssize_t nread;
    int i;

    if(argc != 3){
        printf("3!!!\n");
        exit(-1);
    }

    for(i=0; i<argc; i++){
        printf("입력한 인자는 = %s\n", argv[i]);
    }

    fdin = open(argv[1], O_RDONLY);
    fdout = open(argv[2], O_WRONLY | O_CREAT);

    while((nread = read(fdin, buf, 1024)) > 0){
        if(write(fdout, buf, nread) < nread){
            close(fdin);
            close(fdout);
            printf("비정상종료\n");
            exit(-1);
        }
    }
    close(fdin);
    close(fdout);

    return 0;
}
```

입력한 인자 출력

입력한 2번째 인자를 열고, 3번째 인자를 생성한다

открытый файл копирует в новый созданный файл

```
mhn@mhn-900X3L: ~/my_proj/c/18_s/study
mhn@mhn-900X3L:~$ cd
mhn@mhn-900X3L:~$ cd my_proj/c/18_s/study
mhn@mhn-900X3L:~/my_proj/c/18_s/study$ vi file5.c
mhn@mhn-900X3L:~/my_proj/c/18_s/study$ gcc file5.c
mhn@mhn-900X3L:~/my_proj/c/18_s/study$ ./a.out t1.txt test.txt
입력한 인자는 = ./a.out
입력한 인자는 = t1.txt
입력한 인자는 = test.txt
mhn@mhn-900X3L:~/my_proj/c/18_s/study$
```

입력한 인자 출력

t1.txt

```
mhn@mhn-Z20NH-AS51B5U: ~/linux/19/test
1 aaaaaaa
~
~
~
```

test.txt

```
test.txt (~/my_proj/c/18_s/study) - gedit
Open [?]
test.txt
aaaaaaa
```

T1.txt의 내용이 새로 생성된 test.txt에 복사되었다


```

mhn@mhn-900X3L: ~/my_proj/c/18_s/study
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>

int main(void){

    int nr;
    char buf[1024] = {0};

    nr = read(0,buf,sizeof(buf));
    printf("nr = %d\n",nr);
    write(1,buf,nr);

    return 0;

}
~

```

표준입력으로 입력 받는다
입력 받은 것을 출력한다

```

mhn@mhn-900X3L:~/my_proj/c/18_s/study$ vi file6.c
mhn@mhn-900X3L:~/my_proj/c/18_s/study$ gcc file6.c
mhn@mhn-900X3L:~/my_proj/c/18_s/study$ ./a.out
123
nr = 4
123
mhn@mhn-900X3L:~/my_proj/c/18_s/study$

```

입력 받은 글자수 +1만큼 저장됨