

```
read(fd, buf, 읽을크기)
write(fd, buf, 글쓰기)
printf("fd =%d\n", filedes)
```

O\_RDONLY /읽기전용으로 파일을 열겠다  
O\_WRONLY /쓰기전용으로 파일을 열겠다  
O\_RDWR /읽기쓰기전용으로 파일을 열겠다

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

```
int main(void)
{
    int fdin, fdout;
    ssize_t nread;
    char buf[1024];

    fdin = open("temp1.txt", O_RDONLY);          //읽기 용 읽을게없으면 안열림
    fdout = open("temp2.txt", O_WRONLY|O_CREAT|O_TRUNC,0644);
                                                //실행할때마다 파일 밀어버림

    while((nread = read(fdin, buf, 1024)) >0) //nread 는 파일을 읽음
                                                //read(fd,buf,읽을크기)
                                                //nread 는 내가 몇 바이트를 읽었나를 받게됨

    //0 보다 작으면 읽을게 없음

    {
        if(write(fdout, buf, nread) <nread)
        // write(fd,buf,쓸크기)
        {
            close(fdin);                      //close 는 시스템콜
            close(fdout);
        }
    }
    close(fdin);
    close(fdout);
    return 0;
}
```

OPEN([경로&이름], [설정], 0644)  
open-파일 리턴

CREAT([경로&이름],0644)  
creat-파일리턴

CLOSE([경로&이름]) → 파일닫기

fd

=

files\_struct 를 보면 files → f\_pos → file\_operation → inode → path

files 는

f\_pos 의 \*\*fd → files \*fd[]

[]배열의 인덱스를 유저한테 넘겨줌

////////////////////////////////////

```
#include <sys/types.h>
```

```
#include <unistd.h>
```

```
#include <fcntl.h>
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int filedес;
```

```
    off_t newpos;
```

```
    filedес = open("data1.txt", O_RDONLY);
```

```
    newpos = lseek(filedес, (off_t)0, SEEK_END);
```

```
    printf("file size : %d\n", newpos);
```

```
}
```

//////// text.1 에 hello 입력 했더니

결과

file size : 6

```
#include <stdio.h>
```

```
int main(void)
```

```

{
    FILE *fp = fopen("mycat.c", "r");    //fopen
    char buf[1024] = "\0";
    int ret;
    while(ret = fread(buf, 1, sizeof(buf), fp))    //fread 는 읽어서 버퍼에 저장 1 바이트씩읽어서 sizeof 1
    바이트씩 1024 번 읽어라 buf 에 넣어라
    {
        usleep(1000000);                //us = 10^-6 승 , 1 초후에 모니터에 써라
        fwrite(buf, 1, ret, stdout);    //
    }
    fclose(fp);
    return 0;
}

```

```

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

```

```

int main(int argc, char **argv)
{
    int fd, ret;
    char buf[1024];
    if(argc != 2)                                // argc 가 두개아니면 종료
    {
        printf("Usage: mycat filename\n");        //argc
        exit(-1);
    }
    fd = open(argv[1], O_RDONLY);                //onpe argv 1 번
    while(ret = read(fd, buf, sizeof(buf)))        //read
    {
        write(1, buf, ret);
    }
    close(fd);
    return 0;
}

```

./a.out 하면

Usage : my cat filename

./a.out cat1.c 하면 안에 코드 내용이 나옴

```

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

```

```

ret = read(0, buf, sizeof(buf));                //바이트 수를 읽어들임(글자수)를 읽어들임

```

//키보드 읽어서 바로 모니터출력

```
write(1, buf, ret);
```

“je 파일명

vs = 수직으로 나눔

sp = 커서있는 창을 수평으로 나눔

q = 커서있는 창 종료

:e 파일명 -> 파일명으로 들어감 (생성)

ctrl + ww ->창 넘어가기

gcc -o 파일이름 [컴파일 1.c] [컴파일 2.c] 헤더파일은 안넣어도됨

cat1.h 하나만들고

// 애는 선언해주는애고

cat1.c 총 두개를 만들어야 하나임

//애는 함수내용임

헤더파일에서는

< > 시스템 헤더

“ “ 내가만든거 커스텀

리눅스 커널

wget https:

다깔고

tar zxvf linux-4.4.tar.gz

cd linux-4.4

```

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

int main(int argc, char **argv)
{
    int fd = open(argv[1], O_RDONLY);
    int line = 0;
    int word = 0;
    int flag = 0;
    int cnt = 0;
    char ch;
    if(argc != 2)
    {
        printf("You need 1 more parameter\n");
        printf("Usage : mywc filename\n");
        exit(-1);
    }
    if((fd = open(argv[1], O_RDONLY)) < 0)
    {
        perror("open() ");
        exit(-1);
    }
    while(read(fd, &ch, 1))
    {
        cnt++;
        if(ch == '\n')
            line++;
        if(ch != '\n' && ch != '\t' && ch != ' ')
        {
            if(flag == 0)
            {
                word++;
                flag = 1;
            }
        }
        else
        {
            flag = 0;
        }
    }
    close(fd);
    printf("%d %d %d %s\n", line, word, cnt, argv[1]); // 라인, 단어종류수, 단어수
    return 0;
}

```

