O_TRUNC-임시적 데이터 저장할때 사용
        밀어버린다는 뜻 /새로 갱신한다는 뜻

RDONLY - 읽기전용 (읽을거리가없음 열리지않음)

fd - filedescriptor의 약자
  파일의 식별자

read(fd,buf,읽을크기)
write(fd,buf,쓸크기) -> 버프에 있는 내용이 fd로 간다 /거기에 쓰는 크기.

file_io4.c는 복사하는거 코드

system call(SW intrpt)

newpos - lseek(fildes.(off_t)0, SEEK_END);

#ifndef랑 #endif랑은 쌍임.
————————————————————————————————————

*****WC

```
#include<fcntl.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<errno.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 parameterWn");
                printf("Usage: mywc filenameWn");
                exit(-1);
        }

        if((fd= open(argv[1],O_RDONLY)) <0)
        {
                perror("open()");
                exit(-1);
        }
        while(read(fd,&ch,1))
        {
                cnt++;
                if(ch == 'Wn')
                        line++;
                if(ch != 'Wn' && ch != 'Wt' && ch != ' ')//숫자거나 문자
                {
                        if(flag == 0)
                        {
                                word++;
                                flag = 1;
                        }
                }
                else
                {
```

```
                        flag = 0;
                }
        }
        close(fd);
        printf("%d %d %d %sWn",line,word,cnt,argv[1]);
        return 0;
}
```

---

***cat

```
#include<stdio.h>

int main(void)
{
        FILE *fp = fopen("mycat.c","r");
        char buf[1024] = "WO";
        int ret;
        while(ret = fread(buf, 1, sizeof(buf),fp))
        {
                usleep(1000000);
                fwrite(buf,1,ret,stdout);
        }
        fclose(fp);
        return 0;
}
~
```

---

**********file_io3.c

```
#include<fcntl.h>

int main(void)
{
        int filedes1, filedes2;

        filedes1 = open("data1.txt", O_WRONLY | O_CREAT | O_TRUNC, 0644);
        filedes2 = creat("data2.txt",0644);

        close(filedes1);
        close(filedes2);

        return 0;
}
```

---

**********file_io4.c

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

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

        fdin = open("temp1.txt", O_RDONLY);
        fdout = open("temp2.txt",O_WRONLY | O_CREAT | O_TRUNC, 0644);

        while((nread = read(fdin, buf, 1024)) > 0)// fdin을 버퍼시킨다.
        {
                if(write(fdout, buf, nread) < nread)//:
                {
                        close(fdin);
                        close(fdout);
                }
        }
        close(fdin);
        close(fdout);
```

```
        return 0;
}
```

————————————————————————————————————————

**********창 3개나눈거

```
#include<stdio.h>
#include<unistd.h>
#include<stdio.h>
#include"my_scanf.h"

int main(void)
{
        int nr;
        char buf[1024] = {0};

        nr = my_scanf(buf,sizeof(buf));
        printf("nr = %dWn", nr);
        write( 1, buf , nr);

        return 0;
}
```

이거랑

```
#include "my_scanf.h"

int my_scanf(char *buf, int size)
{
        int nr = read(0, buf, size);
        return nr;
}
```

이거

```
#ifndef __MY_SCANF_H__
#define __MY_SCANF_H__

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

int my_scanf(char *,int);

#endif
```

————————————————————————

뭔지모르는거

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

int main(int argc, char **argv)
{
        int fd,ret;
        char buf[1024];
        int fdout,nread;

         if(argc !=2)
        {
         printf("1개의 인자만 받는다Wn");
         exit(-1);
        }
```

```
        fdout = open(argv[1], O_WRONLY | O_TRUNC);

        nread = read(0 , buf, sizeof(buf));

        if(write(fdout,buf,nread) <nread)
        {
                printf("ERRORWn");
        }
        close(fdout);

        return 0;

}
```