

```

1 #include<stdio.h>
2 #include<fcntl.h>
3 #include<stdlib.h>
4 #include<sys/types.h>
5 #include<unistd.h>
6 #define ERROR -1
7 /*
8 int main()
9 {
10     int filedес;
11     char pathname[]="temp.txt";
12     if((filedes=open(pathname, O_CREAT | O_RDWR | O_EXCL,0664))==ERROR) //temp.txt를 만들고 ,RDWR 읽기쓰기로 설정 ,EXCL temp.txt가 있는지 중복 확인
13     {
14         printf("File Open Error");
15         exit(1);//1반환
16     }
17     printf("fd=%d\n",filedes);
18     close(filedes);
19     return 0;
20     // 슬마시고 코딩하다 CREATE에서 E를 빼먹음
21     // & 교집합 , | 합집합
22     // 0664는 권한 나중에 다시 배운다 .
23 } // 0은 표준입력 , 1은 표준출력 , 2는 표준에러 그 이후 3번부터는 만들기 나름 .
24
25 -----

```

```

26 */
27
28 /*
29 int main()
30 {
31     int filedес1,filedes2;
32
33     filedес1=open("dataa1.txt",O_WRONLY|O_CREAT|O_TRUNC,0664);//data1.txt를 만들고 쓰기 전용으로 설정 , data1에 아무거나 치고 저장해도 다시 a.out 해보면 TRUNC 때문>
34     에 초기화 되어있다 .
35     filedес2=creat("dataa2.txt",0664);//data2를 생성하는데 계속 새로 생성해서 덮어쓰기 해준다 . open 으로 열어서 다른 옵션으로 해주면 덮어쓰기 안함 .
36
37     close(filedes1);//open 해줬으니 닫는다 .
38     close(filedes2);//open 해줬으니 닫는다 .
39     return 0;
40 }
41 -----

```

```

42
43
44 /*
45
46 int main()
47 {
48
49     int fdin,fdout;
50     ssize t nread;
51     char buf[1024];
52
53     fdin=open("temp1.txt",O_RDONLY);//temp1.txt를 읽기전용으로 생성
54     fdout=open("temp2.txt",O_WRONLY|O_CREAT|O_TRUNC,0664);//temp2.txt를 쓰기전용으로 만들고 실행시마다 초기화
55
56     while((nread=read(fdin,buf,1024))>0)//temp1.txt의 글자수를 읽는데 일단 0보단 크다 뭔가 퍼낼게 있으니까
57         // (fdin,buf,1024) -> buf라는 바가지가 있고 크기가 1024다 buf라는 바가지로 fdin 을 푼다는것 . 푼다는건 뭔가 있다는 뜻이니까
58         0보단 크다 .
59         {
60
61             if(write(fdout,buf,nread)<nread)//write는 뿌려주는건데 nread가 아까 푼 물이다 이거를 fdout에 뿌린다 .
62             { // nread를 뿌렸으니 nread와 같아야 한다 근데 < 이게 말이 안됨 그래서 빠져나간다 .
63                 close(fdin);//open 으로 열어줬으니까 닫아준다 .
64                 close(fdout);//open 으로 열어줬으니까 닫아준다 .
65             }
66
67         }
68
69         close(fdin);
70         close(fdout);
71
72         return 0;
73     }
74 }
75 -----

```

```

73 }
74 -----
75 */
76
77 /*
78
79 int main()
80 {
81     int filesdes;
82     off_t newpos;
83
84     filesdes=open("data1.txt",0_RDONLY);//data1.txt 읽기전용으로 읽고 그 37을 filesdes에 기억해준다?
85     newpos=lseek(filesdes,(off_t)0,SEEK_END);//0부터 끝까지 filesdes 3을 찾아서 newpos에 넣어준다?
86                                     //off_t0 : 시작을 0부터 / seek end 끝 까지 순회해라 = 파일 사이즈
87     printf("file size : %ld\n",newpos);//3인덱스가져 글자수가 몇인지를 본다 +1 은 널값
88     printf("%d",filesdes);//012 다음 이니까 3이야 그냥
89 }
90

```

```

94
95 int main(int argc,char **argv)
96 {
97     int i;
98     char buf[1024];
99     ssize_t nread;
100     int fdin,fdout;
101
102     if(argc!=3)
103     {
104         printf("인자 입력 3개\n");
105         exit(-1);
106     }
107
108     for(i=0;i<argc;i++)
109     {
110         printf("당신이 입력한 인자는 %s\n",argv[i]); // ./a.out 이 0 이고 복사할파일이 1이고 그다음 복사될파일명이 2이다 .
111     }
112
113     fdin=open(argv[1],0_RDONLY);//1번째 파일 그러니까 복사할파일을 읽기 전용으로 연다 근데 그 인덱스 ? 그게 3임
114     fdout=open(argv[2],0_CREAT|0_WRONLY|0_TRUNC);//2번째 파일 복사될파일명 그걸 만들고 읽기 전용 그리고 초기화 근데 그 인덱스 ? 그게 4임
115
116     while((nread=read(fdin,buf,1024))>0)//복사할파일을 퍼다가
117     {
118         if(write(fdout,buf,nread)<nread)//복사될파일에다가 뿌린다 .
119         {
120             close(fdin);
121             close(fdout);
122             printf("비정상 종료\n");
123             exit(-1);
124         }
125     }
126     close(fdin);//open 했으니 닫는다 .
127     close(fdout);//open 했으니 닫는다 .
128     printf("정상 종료\n");
129
130     return 0;
131 }
132

```

```

/*
int main()
{
    FILE *fp=fopen("mycat.c","r");
    char buf[1024]="\0";
    int ret;
    while(ret=fread(buf,1,sizeof(buf),fp))//1바이트씩 읽고          //느리다는 것을 보여줌
    {
        usleep(1000000); // usleep 마이크로세컨드 10 6제곱 =1초
        fwrite(buf,1,ret,stdout);//stdout 표준출력 (모니터)
    }
    fclose(fp);
    return 0;
}
-----
*/

/*
int main(int argc,char **argv)//argc는 012 argv 012 의 내용
{
    int fd,ret;
    char buf[1024];
    if(argc!=2)                //      ./a.out mycat.c 치니까 어짜피 2번째는 뭐가 없으니까 실행 x
    {
        printf("Usage: my cat filename\n");
        exit(-1);
    }
    fd=open(argv[1],O_RDONLY); //      ./a.out mycat.c 마이캣.c를 읽어서 fd로 보낸다
    while(ret=read(fd,buf,sizeof(buf)))// 버퍼의 크기만큼 퍼냄 그걸 렛에
    {
        write(1,buf,ret); // 1이 표준출력이니까 ret 값을 뿌려준다 그게 마이캣 내용
    }
    close(fd);
    return 0;
}

```

```

175
176 int main(int argc,char **argv)
177 {
178     int fd,ret;
179     int fdout,nread;
180
181     char buf[1024];
182
183     if(argc!=2)
184     {
185         printf("1개의 인자만 받는다");          //1개의 인자만 받으니까 실행 x 2개의 인자하면 오류값 출력
186         exit(-1);
187     }
188
189     fdout=open(argv[1],O_WRONLY|O_TRUNC); //      ./a.out 어쩌구저쩌구.c 하면 쓰기전용으로 부르고 계속 초기화
190
191     nread=read(0,buf,sizeof(buf));          // 표준입력 친걸 nread에
192
193     if(write(fdout,buf,nread)<nread) //      nread 에 친걸 뿌려준다
194     {
195         printf("ERROR\n");
196     }
197     close(fdout);
198
199     return 0;
200 }
201

```