

기본 문법 파헤치기 .

1.open 파일 열기

파일을 사용하기 위해 열기(open)을 한다.

파일열기에는 표준함수인 fopen()도 있다. (둘의 차이점은 조금 있다가)

open을 사용할 때는 용도에 따라 읽기 전용, 쓰기 전용, 읽기와 쓰기 모두되는 옵션이 있다.

fcntl.h에 정의된다.

헤더 fcntl.h

형태 int open(const char *FILENAME, int FLAGS [, mode_t MODE]

인수 char *FILENAME 대상 파일 이름

int FLAGS 파일에 대한 열기옵션

[,mode_t MODE] O_CREAT 옵션 사용에 의해 파일이 생성될 때 지정되는 파일의 접근권한

옵션

O_RDONLY 읽기 전용

O_WRONLY 쓰기 전용

O_RDWR 읽기 쓰기

O_CREAT 해당 파일이 없으면 생성한다.

O_CREAT로 파일을 생성하게 되면 파일의 접근권한을 지정하기 위해
접근권한 값을 추가한다.

O_EXCL O_CREAT을 사용했을 때, 이미 파일이 있어도 열기가 가능하여 쓰기를 하면 원래내용이
삭제된다.

O_EXCL을 같이 사용하면, 이미 파일이 있을 경우 open()되지 않고 종료된다.

O_TRUNC 기존파일 내용을 삭제한다.

2.write 파일 쓰기

*open()함수로 열기를 한 파일에 쓰기를 합니다.

open()함수는 fcntl.h로 정의 되어있지만(cteat()도) write(), read(), close()는 unistd.h.에 정의되어있다.

형태 `ssize_t write(int fd, const void *buf , size_t n)`

인수 `int fd` 파일 디스크립터

`void *buf` 파일에 쓰기를 할 내용을 담은 버퍼

`size_t n` 쓰기할 바이트 개수

반환 `ssize_t` 정상적 쓰기를 했다면 쓰기를 한 바이트 개수를, 실패했다면 -1을 반환한다.

3.creat 파일 생성

*파일을 생성

이미 같은 이름의 파일이 있다면 파일의 내용을 모두 삭제하고 열기

그리고 *쓰기만 가능하다.

그리고 open()으로도 표현할 수 있다. (open이 읽기쓰기가 다 되서 유용하다)

예를들어

```
creat("./test.txt",0644);
```

```
open("./test.txt", O_WRONLY | O_CREAT | O_TRUNC ,0644);
```

헤더 `fcntl.h`

형태 `int creat(const char *file , mode_t mode)`

0320속제

인수 char *file 경로명포함 파일전체이름

mode_t mode 생성할 파일의 접근권한