

Chapter 4. IO

Open

그냥 열 때 `int open (path, flag)`

새로 파일 만들때 int open (path, flag, mode)
 ↓ ↓ ↓
 file descriptor 반환 access mode 권한 부여

flag

필요

- O_RDONLY
- O_WRONLY
- O_RDWR

- O_APPEND file offset을 파일 맨뒤로
- O_TRUNC file 초기화
- O_CREAT 파일 새로 생성 (있으면 무시됨)
- O_EXCL 파일 이미 존재하면 오류 반환
- O_NOCTTY

~~X~~ O_CREAT | O_EXCL 3. 할당가능!

mode (Permission mask)

644, 755 ...

$$6 = \begin{pmatrix} 1 & 1 & 0 \end{pmatrix}_2 \quad 4 = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}_2 \quad 7 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}_2$$

$\begin{smallmatrix} \text{r} & \text{w} & . \end{smallmatrix}$

$\begin{smallmatrix} \text{r} & . & . \end{smallmatrix}$

$\begin{smallmatrix} \text{r} & \text{w} & \text{x} \end{smallmatrix}$

52 !

S_I · R { USR
 · W { GRP
 · X { OTH
 · RWX { U
 G
 O

→ (S_IRUSR | S_IWUSR | S_IROTH)
 처럼 활용 가능

close

`r_close` → interrupt 되었을 때 처리해 줌.

lseek

`lseek (file_descriptor, offset, start_flag)`

|
파일지정

|
± 몇만큼
이동

|
SEEK_SET : file 시작

SEEK_CUR : 현재 offset 위치

SEEK_END : file 끝

read

`read (file_descriptor, buffer, n_byte)`

buffer의 크기는 $\geq n$ 이어야 함!

* 요청한 n-byte 보다 적게 읽힐 수 있다!

맞은 byte 수 반환, EOF면 0 반환, -1은 오류.

char *buf; ← 버퍼를 댄 포인터로만 만들면, 컴파일은 되지만, 실행오류!

write

write (file_descriptor, buffer, n-byte)

read block : 요청한 byte 수 만큼 읽도록 강제.

file descriptor

프로세스마다 하나씩 존재. fork 하면 그대로 복사됨.

0		:	STDIN_FILENO	} 기본적으로 열려있음.
1		:	STDOUT_FILENO	
2		:	STDERR_FILENO	
:		:		

↓ 포인터

System file table

 : file offset 정보

↓

inode table

 : 실제 파일 정보 (파일당 하나!)

Filters and Redirection

ls > a.txt stdout 에 write 될 것을 a.txt 파일 디스크립터로 redirect.

sort < b.txt

dup2 (file_descriptor 1, file_descriptor 2)

복사
X 닫힘