리눅스

- 저수준 입출력 함수

※ open() -> 파일 열기

**int open(const char \*pathname, int flags, /\* mode\_t mode\*/);**

1. pathname : 가리키는 파일

2. flags : 파일 접근 모드를 지정하는 비트마스크

3. mode : open()호출에 O\_CREATE를 지정하면 mode를 생략해도 됨

리턴 : 성공하면 파일 디스크립터를 리턴, 에러가 발생하면 -1 리턴

※ creat() -> 새로운 파일을 만들고 열기 위해서 시스템 호출 사용

**int creat(const char \*pathname, mode\_t mode);**

1. pathname : 가리키는 파일

2. mode : 생성할 파일의 접근 권한

리턴 : 파일 디스크립터를 리턴, 에러가 발생하면 -1 리턴

※ read() -> 디스크립터 fd가 가리키는 열려 있는 파일에서 데이터를 읽음

**sizet\_t read(int fd, void\* buffer, size\_t, count)**

1. fd : 데이터를 쓸 파일을 가리키는 파일 디스크립터

2. buffer : 입력 데이터를 담을 메모리 버퍼의 주소. 최소한 count바이트만큼 커야함

3. count : 읽을 최대 바이트 수를 지정함

리턴 : 읽은 바이트 수를 리턴, EOF인 경우에는 0을, 에러가 발생하면 -1 리턴

※ write() -> 열려 있는 파일에 데이터를 쓴다.

**ssize\_t write(int fd, void \*buffer, size\_t count)**

1. fd : 데이터를 쓸 파일을 가리키는 파일 디스크립터

2. buffer : 쓸 데이터의 주소

3. count : 버퍼에서 읽어와 쓸 바이트 수

리턴 : 쓴 바이트 수를 리턴함. 에러가 발생하면 -1을 리턴

※ close() -> 열려 있는 파일 디스크립터를 닫고, 프로세스가 차후에 재사용 할수 있게 해줌.

**int close(int fd)**

1. fd : 데이터를 쓸 파일을 가리키는 파일 디스크립터

리턴 : 성공하면 0리턴, 에러가 발생하면 -1 리턴

※ lseek() -> 파일이 열렸을 때 파일의 시작을 가리키도록 설정됨

**off\_t lseek(int fd, off\_t offset, int whence)**

1. fd : 파일 디스크립터

2. offset : 바이트 단위로 지정

3. whence : offset의 기준점을 나타냄

리턴 : 성공하면 새로운 파일 오프셋을 리턴, 에러가 발생하면 -1을 리턴

※ fstat() -> 열려진 파일의 크기, 파일의 권한, 파일생성일시 등 파일의 상태나 파일의 정보를 얻을 수 있는 함수

**int fstat(int fd, struct stat \*buf)**

1. fd : 파일 디스크립터

2. buf : 파일 상태 및 정보를 저장할 buf 구조체

리턴 : 0이면 정상적으로 조회, 에러가 발생하면 -1 리턴

- 고수준 입출력 함수

※ fopen() -> 파일을 읽어 들이거나 쓰기를 하기 위해서 사용하는 함수

**\*fopen(const char\* path, const char\* mode)**

1. char\* path : 열고자하는 파일의 전체 이름

2. char\* mode : 파일 열기를 위한 옵션, 사용 형태

리턴 : 파일포인터를 반환, 열기에 실패하면 NULL을 반환

※ fwrite() -> 파일 버퍼에 내용 쓰기

**fwite(void\* ptr, size\_t size, size\_t nitems, FILE\* stream)**

1. void\* ptr : 파일 내용에 출력할 메모리 포인터

2. size\_t size : 데이터 하나의 크기

3. size\_t nitems : 쓰기할 데이터의 개수

4. FILE\* stream : 대상 파일 스트림

리턴 : 출력한 데이터 개수를 반환, 오류가 발생하면 -1 반환

※ fread -> 파일 버퍼에 내용 읽기

**fread(void\* ptr, size\_t size, size\_t nitems, FILE\* stream)**

1. void\* ptr : 파일 내용을 읽어 들일 메모리 포인터

2. size\_t size : 데이터 하나의 크기

3. size\_t nitems : 읽어 들일 데이터의 개수

4. FILE\* stream : 대상 파일 스트림

리턴 : 읽어들인 데이터 개수를 반환, 오류가 발생하면 -1 반환

※ fseek() -> 파일의 읽기/쓰기 위치를 파일의 처음 위치로 초기화

**fseek(FILE\* stream, long offset, int whence)**

1. FILE\* stream : 대상 파일 스트림

2. long offset : 이동할 바이트 수

3. int whence : 시작 지점

리턴 : 0을 반환, 오류가 발생하면 -1을 반환

※ ftell -> 파일 읽기/쓰기 위치 구하기

**ftell(FILE\* stream)**

1. FILE\* stream : 대상 파일 스트림

리턴 : 읽기/쓰기 위치 반환, 오류가 발생하면 -1 반환

※ 저수준/고수준 입출력 함수 비교

