

4.11

파일 입출력

개요

프로그램은 한 번 실행하고 나면 그 안에 입력했던 데이터들이 모두 사라집니다. 프로그램이 변수를 만들고 그 안에 데이터를 넣어두거나 할 때 사용하는 메모리는 영구적으로 데이터를 저장할 수 있는 작업공간이 아니기 때문입니다. 데이터를 영구적으로 저장해두고 싶다면 프로그램이 실행될 때 파일에 저장해두면 됩니다. 이 파일에 저장된 데이터는 다음 번에 프로그램이 실행될 때 다시 읽어올 수 있습니다. C에는 **파일 입출력 (File IO)** 기능이 있어서 파일에 데이터를 저장하거나 읽어오도록 만들 수 있습니다.

핵심개념

- \* 파일 입출력
- \* FILE 구조체

파일 열기

```
1 FILE* file;  
2 file = fopen("file.txt", "r");
```

▲ <코드 1>

C의 표준 입출력 라이브러리인 **stdio.h**에서 파일을 읽고 쓰는 기능을 제공합니다. 먼저 파일에 접근해야 파일의 데이터를 읽어오거나 쓸 수 있습니다. 파일은 **FILE이라고 하는 구조체로** 구현되어 있습니다. 따라서 <코드 1>의 1행과 같이 FILE\* file; 코드를 이용해 변수를 선언할 수 있습니다.

2행의 fopen 함수는 파일을 열고 그 파일에 접근할 수 있는 포인터라는 것을 넘겨줍니다. (포인터는 변수나 다른 어떤 것이 저장된 메모리의 주소입니다. 즉 데이터 값 그 자체가 아니라 데이터에 접근할 수 있는 길이라고 생각하면 됩니다.) 만약 file에 NULL이 반환되었다면 요청한 파일을 열지 못한 것입니다.

fopen 함수를 호출할 때 전달하는 첫 번째 인자는 파일 이름이고 두 번째 인자는 모드입니다. 즉, 'r'은 읽기 모드, 'w'는 쓰기 모드, 'a'는 추가 모드입니다.

파일 열고 쓰기

읽기	구분	쓰기
fgetc	파일 내용 중 한 글자	fputc
fgets	파일 내용의 한 줄	fputs
fread	파일의 변수, 구조체, 배열 등	fwrite
fscanf	형식 문자열 사용	fprintf

파일을 열고 나면 파일의 내용을 읽거나 쓸 수 있습니다.

파일의 내용 중에서 한 글자만 읽어오고 싶다면 **fgetc** 함수를 사용합니다. 만약 한 줄을 읽어오고 싶다면 **fgets** 함수를 사용할 수 있고 정해진 길이만큼의 데이터를 읽어오고 싶다면 **fread** 함수를 사용할 수 있습니다. 형식문자열을 사용하여 데이터를 읽어오기 위해서는 **fscanf**를 이용합니다.

파일에 데이터를 한 글자만 쓰고 싶다면 **fputc** 함수를 사용하고 한 줄을 쓰고 싶다면 **fputs** 함수를 사용할 수 있습니다. **fprintf** 함수를 사용하면 **printf** 함수처럼 형식문자열을 이용해 데이터를 쓸 수 있습니다. **fwrite** 함수는 배열에 들어있는 데이터를 파일에 쓸 수 있도록 합니다.

## 파일 닫기

```
1 | fclose(file);
```

▲ <코드 2>

파일을 한 번 열면 그 파일은 반드시 닫아야 합니다. 왜냐하면 파일을 계속 열기만 하면 파일을 열 수 있는 시스템 리소스가 소진되어 더 이상 파일을 열 수 없게 될 수도 있기 때문입니다. UNIX 시스템에서는 프로그램이 종료될 때 열려있는 파일이 있으면 자동으로 닫아주기도 합니다. 하지만 파일을 사용하고 나면 항상 파일을 닫는 습관을 들이는 것이 좋습니다

