12\_파일 입출력

학습목표

스트림의 개념을 이해한다.

형식화된 입출력을 이해한다.

파일 입출력 함수들을 사용할 수 있다.

이진 파일을 사용할 수 있다.

파일에 임의 접근할 수 있다.

1. 스트림

일반적으로 프로그램에서는 화면이나 키보드, 파일등의 입출력 장치에 대하여 데이터를 쓰거나 읽게 된다. 입출력 장치들은 상당히 다양한 방식으로 데이터를 주고받지만 C에서는 스트림(stream)이라는 개념을 사용하여서 동일한 방법으로 입출력 할 수 있다.

스트림이란 모든 입력과 출력을 바이트(byte)들의 흐름으로 생각하는 것이다. 어떤 입출력 장치던지 상관없이 바이트 단위로 입출력이 이루어진다. 스트림의 최대 장점은 장치 독립성이다. 입출력 장치에 상관없이 동일한 함수를 사용하여 프로그램을 작성할 수 있다.

1. 표준 입출력 스트림

프로그램의 동작에 필수적인 몇 개의 기본적인 스트림은 프로그램 실행 시에 자동으로 생성된다. 이것을 표준 입출력 스트림(standard input/output stream)이라고 한다.

Stdin 표준 입력 스트림 키보드 scanf

Stdput 표준 출력 스트림 화면 printf

Stderr 표준 에러 스트림 화면

1. 파일도 스트림에 연결시킬 수 있다.

C에서 파일은 곧 스트림이다. 앞에서 스트림에 뭐든지 붙일 수 있다고 했는데 파일도 예외가 아니다. 운영 체제에서는 하나의 프로세스 당 100여개의 스트림이 지원되고 스트림 중 0, 1, 2번은 스트림은 stdin, stdout, stderr로 미리 정의된다. 3번 스트림부터 나머지 스트림은 개발자가 파일을 연결해서 사용할 수도 있고 여러 가지 통신 포트를 연결하여 사용할 수 있다.

1. 파일의 기초

지금까지는 키보드와 화면을 통해 스트림을 사용처를 알아봤지만 실제 응용 프로그램에서는 데이터나 현재의 설정 정보를 저장하기 위하여 디스크에 저장되는 파일을 많이 사용한다. 파일을 사용하는 이유는 무엇일까?

지금까지 우리는 모든 데이터를 변수에 저장하였고 이는 영구적인 기억장치가 아니다. 따라서 데이터를 영구적으로 보관하려면 디스크와 같은 보조 기억 장치에 보관해야 한다. C에서는 디스크에 파일을 생성시켜서 데이터를 보관할 수 있다.

1. 파일의 개념

C에서의 모든 입출력은 스트림을 통해 이루어지고 파일도 예외는 아니다. 파일도 스트림으로 취급되기 때문에 파일도 일련의 연속된 바이트라고 생각하면 된다. 따라서 파일에 대한 입출력도 표준 입출력과 동일한 함수들로 이루어진다. 이것이 바로 스트림의 장점이다.

1. 파일의 유형

파일의 유형에는 텍스트 파일과 이진 파일이 있다.

텍스트 파일(text file)은 사람이 읽을 수 있는 텍스트가 들어 있는 파일이다. C 프로그램 소스 파일이나 메모장 파일이 텍스트 파일의 예이다. 텍스트 파일에는 문자들을 나타내는 아스키 코드들이 들어 있다. 텍스트 파일이 중요한 이유는 모니터, 키보드, 프린터 등이 모두 문자 데이터만 처리하기 때문이다.

이진 파일(binary file)은 사람이 읽을 수는 없으나 컴퓨터는 읽을 수 있는 파일이다. 즉 문자 데이터가 아니라 이진 데이터가 직접 저장되어 있는 파일이다. 이진 파일은 텍스트 파일과는 달리 라인들로 분리되지 않는다. 라인의 끝을 표시할 필요도 없다. 모든 문자들은 특별한 의미를 가지지 않고 데이터로 취급된다. 이진 파일은 특정 프로그램에 의해서만 판독이 가능하다. C프로그램 실행 파일, 사운드 파일, 이미지 파일 등이 이진 파일의 예이다.

1. 파일 처리의 개요

프로그램에서 파일을 연다(open)는 것은 파일에서 데이터를 읽거나 쓸 수 있도록 모든 준비를 마치는 것을 의미한다. 내부적으로는 파일과 연결된 스트림을 만들게 된다. 파일을 연 다음에는 데이터를 읽고 수정할 수 있다. 파일을 사용한 후에는 파일을 닫아야(close) 한다.

파일 열기 -> 파일 읽기와 수정 -> 파일 닫기

1. 파일 열기