13강

파일 처리

동양미래대학교 강환수 교수

본 강의 사용 및 참조 자료

- Perfect C, 3판, 강환수 외 2인 공저, 인피니티북스, 2021

15장 파일 처리

목차

1 파일 기초

2 파일 입출력

3 파일 접근 처리

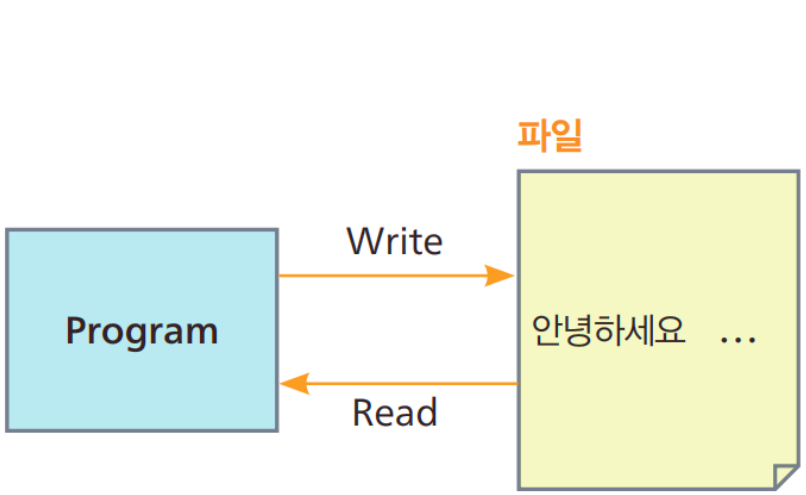
01

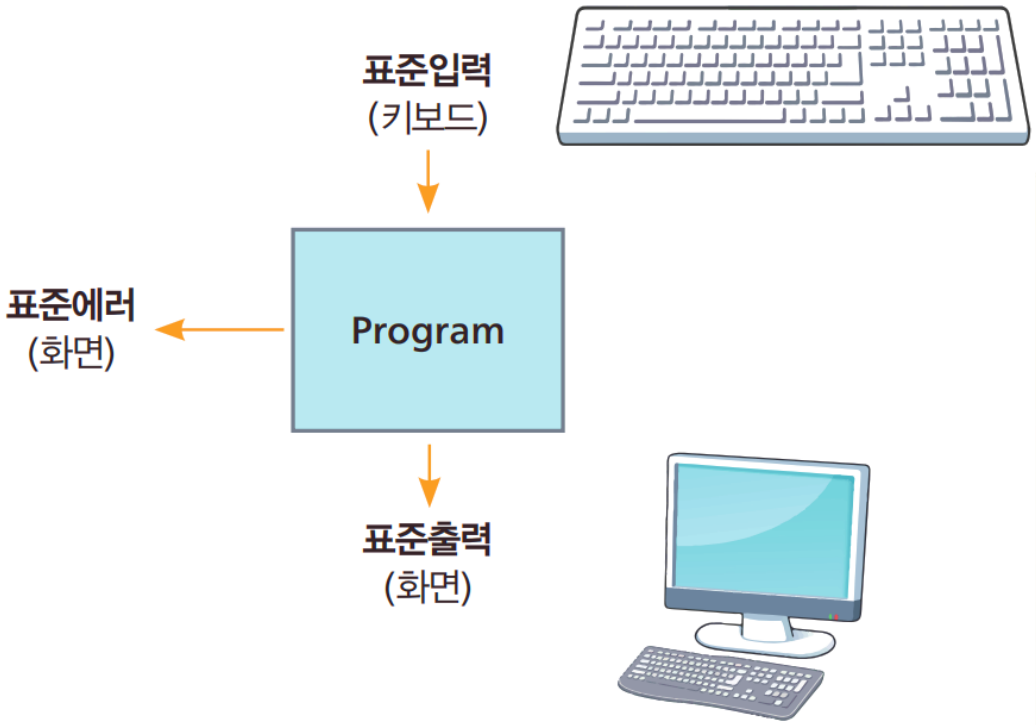
파일 기초

파일의 필요성

- 워드프로세서 없이 프로그램에서
  프로그램의 결과로 구성되는 파일을 직접 만들 수 있을까?
  - 프로그램에서 출력을 파일에 한다면 파일이 생성
  - 반대로 키보드에서 표준 입력하던 입력을
    파일에서 입력하면 파일 입력

파일 입출력과 표준 입출력





[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 646쪽

파일과 메모리

- 프로그램이 종료되면 모두 사라지는 자료
- 프로그램이 종료되더라도 계속 저장
  - 보조기억장치인 디스크에 저장되는 파일(file)은 직접 삭제하지 않은 한 계속 남음
- 프로그램에서 사용하던 정보를 종료 후에도 계속 사용하고 싶다면
  - 프로그램에서 파일에 그 내용을 저장
  - 학생 성적 처리를 프로그램을 통하여 결과를 얻어내 그 처리 결과를 지속적으로 저장하려면 파일에 저장

파일 분류

- 텍스트 파일(text file)과
  이진 파일(binary file) 유형으로 나뉨
- 텍스트 파일
  - 내용이 아스키 코드(ascii code)와 같은 문자 코드값으로 저장
    - 메모리에 저장된 실수와 정수와 같은 내용도
      문자 형식으로 변환되어 저장
    - 텍스트 편집기를 통하여 그 내용을 볼 수 있고 수정 가능
      - 메모장(notepad) 같은 편집기로 열기 가능

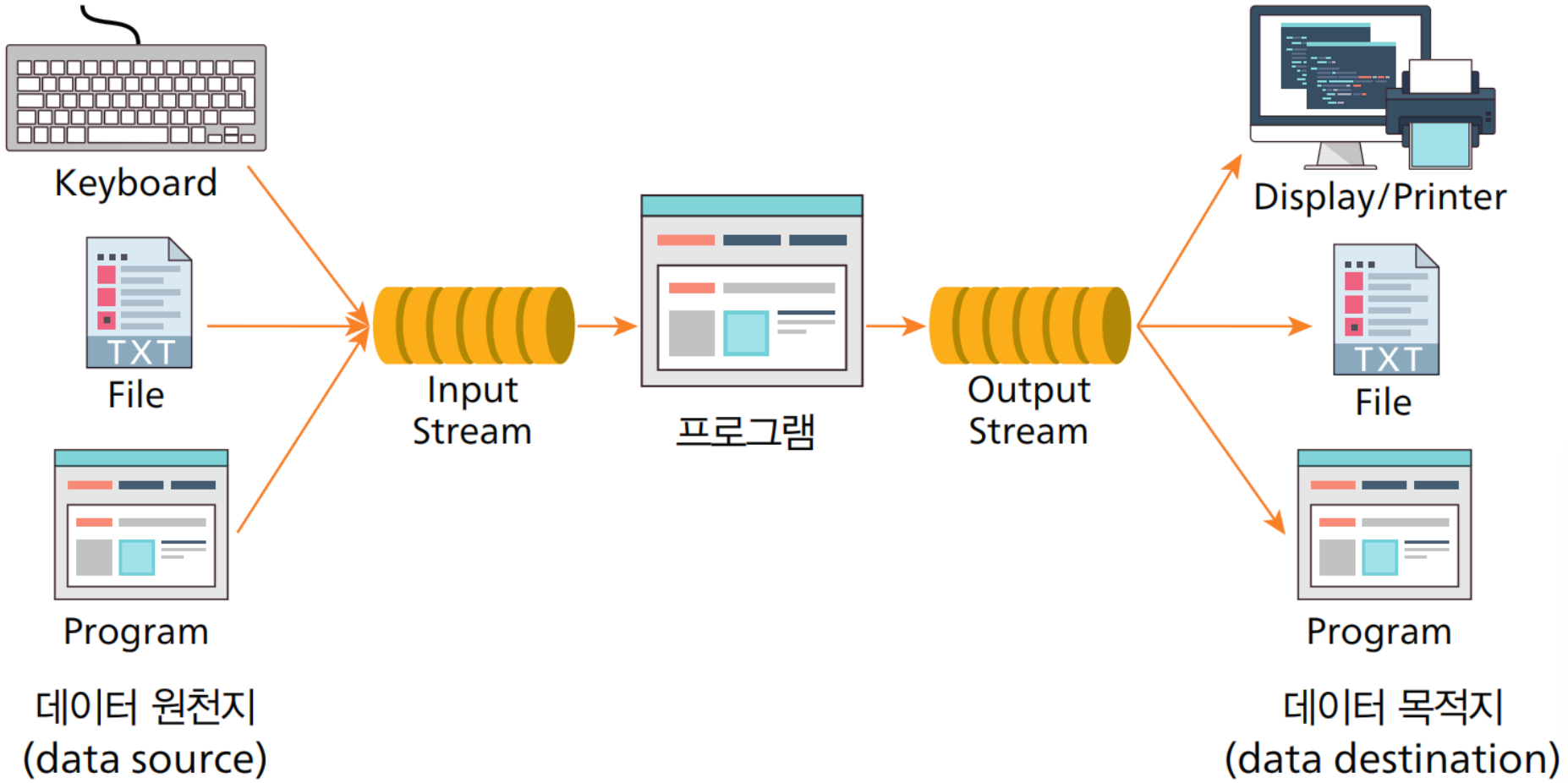
이진 파일(binary file)

- 목적에 맞는 이진 형태(binary format)로 저장되는 파일
  - 컴퓨터 내부 형식으로 저장되는 파일
  - 자료는 메모리 자료 내용에서 어떤 변환도 거치지 않고 그대로 파일에 기록
  - 입출력 속도도 텍스트 파일보다 빠름
    - 메모장과 같은 텍스트 편집기로는 그 내용을 볼 수 없음
    - 내용을 이미 알고 있는 특정한 프로그램에 의해 인지될 때 의미가 있음
- 사용 사례
  - 실행 파일과 그림 파일, 음악 파일, 동영상 파일 등

입출력 스트림

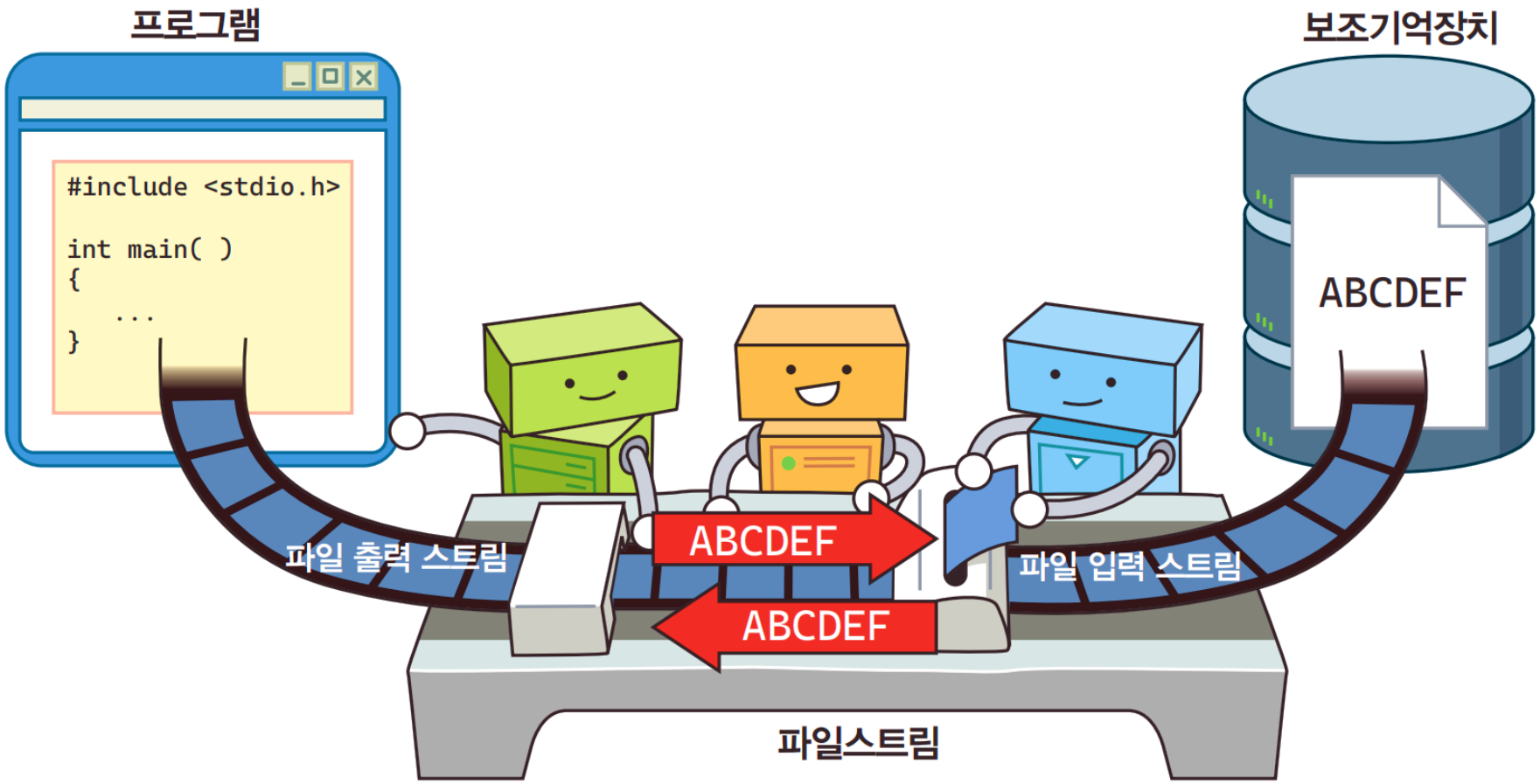
- 자료 입력과 출력은 자료의 이동
  - 자료가 이동하려면 이동 경로가 필요
- 입출력 스트림(IO stream)
  - 입출력 시 이동 통로
  - 표준입력 스트림: 키보드에서 프로그램으로 자료가 이동 경로
    - 함수 scanf( )
      - 표준 입력 스트림에서 자료를 읽을 수 있는 함수
  - 표준출력 스트림: 프로그램에서 모니터의 콘솔로 자료가 이동 경로
    - 함수 printf( )
      - 표준출력 스트림으로 자료를 보낼 수 있는 함수

스트림의 이해



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 648쪽

파일 스트림 개념



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 649쪽

함수 fopen( ): 파일 스트림 열기

- 함수 fopen( ) 또는 fopen_s( )를 이용
  - 프로그램에서 특정한 파일과 파일 스트림을 연결하는 함수
    - 헤더 파일 stdio.h 필요
  - 현재 비주얼 스튜디오
    - 함수 fopen( )는 fopen_s( )로 대체, 함께 사용 가능
  - 함수원형

```
FILE * fopen(const char * _Filename, const char * _Mode);
errno_t fopen_s(FILE ** _File, const char * _Filename, const char * _Mode);
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 650쪽

FILE 자료형

- 헤더 파일 stdio.h에 정의되어 있는 구조체 유형
  - 함수 fopen( )의 반환값 유형 FILE *
    - 구조체 FILE의 포인터 유형
  - 함수 fopen( )은 인자가 파일이름과 파일열기 모드
    - 파일 스트림 연결에 성공
      - 파일 포인터를 반환
    - 실패하면
      - NULL을 반환

```
FILE* f; //파일 포인터
char* fname = "basic.txt"; //파일이름

if ((f = fopen(fname, "w")) == NULL)
{
   printf("파일이 열리지 않습니다.\n");
   exit(1);
};
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 650쪽

함수 fopen_s( )

- FILE 포인터가 인자 f에 저장
  - 스트림 연결에 성공: 정수 0을 반환, 스트림 연결에 실패: 양수를 반환
- 첫 번째 인자
  - 파일 포인터의 주소값
- 두 번째 인자
  - 문자열은 처리하려는 파일 이름
- 세 번째 문자열
  - 파일열기 종류인 모드

```
if (fopen_s(&f, "basic.txt", "w") != 0)
//if ( (f = fopen(fname, "w")) == NULL )
{
    printf("파일이 열리지 않습니다.\n");
    exit(1);
};
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 651쪽

파일열기 종류(모드)

- 텍스트 파일인 경우 "r", "w", "a" 등의 종류
  - 읽기모드 r
    - 읽기가 가능한 모드, 쓰기는 불가능
  - 쓰기모드 w
    - 파일 어디나 쓰기가 가능한 모드, 읽기는 불가능
  - 추가모드 a
    - 파일 중간에 쓸 수 없으며,
    - 파일 마지막에 추가적으로 쓰는 것만 가능한 모드, 읽기는 불가능
    - 파일에 쓰는 내용은 무조건 파일 마지막에 추가

함수 fclose( )

- fopen( )으로 연결한 파일 스트림을 닫는 기능을 수행
- 파일 스트림을 연결한 후 파일 처리가 모두 끝났으면
  - 파일 포인터 f를 인자로 함수 fclose( )를 호출하여 반드시 파일을 닫도록
    - 내부적으로 파일 스트림 연결에 할당된 자원을 반납
    - 파일과 메모리 사이에 있던 버퍼의 내용을 모두 지우는 역할을 수행
- 성공하면 0을 실패하면 EOF을 반환

실습예제 1/2

Prj01 01fopen.c 이름과 성적 정보 내용으로 간단한 파일을 생성 난이도: ★

```
01  #define _CRT_SECURE_NO_WARNINGS
02  #include <stdio.h>
03  #include <stdlib.h> //for exit()
04
05  int main()
06  {
07      char* fname = "basic.txt"; //파일이름
08      FILE* f; //파일 포인터
09
10      char name[30] = "손혜진"; //파일에 쓰려는 자료
11      int point = 99;
12
13      //파일 열기 함수 fopen()과 fopen_s()
14      if ((f = fopen(fname, "w")) == NULL) //if (fopen_s(&f, fname, "w") != 0)
15      {
16          printf("파일이 열리지 않아 종료합니다.\n");
17          exit(1);
18      };
19
```

함수 fopen()의 호출 시, 첫 인자는 파일이름, 두 번째 인자는 모드로 "w"는 쓰기모드이며, 반환 값을 파일 포인터로 선언한 f에 대입, 만일 파일 열기에 실패하면 f에 NULL이 저장됨

파일 열기에 실패하면 메시지 출력하고 exit(1)으로 종료

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 652쪽

실습예제 2/2

```
20      //파일 "basic.txt"에 쓰기
21      fprintf(f, "이름 %s 학생의 성적은 %d 입니다.\n", name, point);
22      fclose(f);
23      //표준출력 콘솔에 쓰기
24      printf("이름 %s 학생의 성적은 %d 입니다.\n", name, point);
25      puts("프로젝트 폴더에서 파일 basic.txt를 메모장으로 열어 보세요.");
26
27      return 0;
28  }
```

지정한 파일 f에 출력 printf()를 하는 기능을 수행

파일 처리가 종료되었으면 fclose()로 스트림을 닫음

```
이름 손혜진 학생의 성적은 99 입니다.
프로젝트 폴더에서 파일 basic.txt를 메모장으로 열어 보세요.
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 653쪽

02

파일 입출력

텍스트 파일에 자료를 쓰거나 읽기 위한 함수

헤더 파일 stdio.h를 포함

- 첫 번째 인자는 입출력에 이용될 파일
  - stdin 또는 stdout를 이용하면 표준 입력, 표준 출력으로 이용이 가능
- 두 번째 인자는 입출력에 이용되는 제어 문자열
- 다음 인자들은 입출력될 변수 또는 상수 목록

함수 fprintf()와 fscanf() 함수원형

```
int fprintf(FILE * _File, const char * _Format, ...);
int fscanf(FILE * _File, const char * _Format, ...);
int fscanf_s(FILE * _File, const char * _Format, ...);
```

위 함수에서 _File은 서식화된 입출력 스트림의 목적지인 파일이며, _Format은 입출력 제어 문자열이며, 이후 기술되는 인자는 여러 개의 출력될 변수 또는 상수이다.

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 656쪽

기호 상수 stdin, stdout, stderr

- 헤더 파일 stdio.h에 정의되어 있는 값
  - 각각 표준 입력, 표준출력, 표준에러를 의미

표준파일	키워드	장치(device)
표준입력	stdin	키보드
표준출력	stdout	모니터 화면
표준에러	stderr	모니터 화면

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 656쪽

함수 fprinf( )와 fsanf_s( )

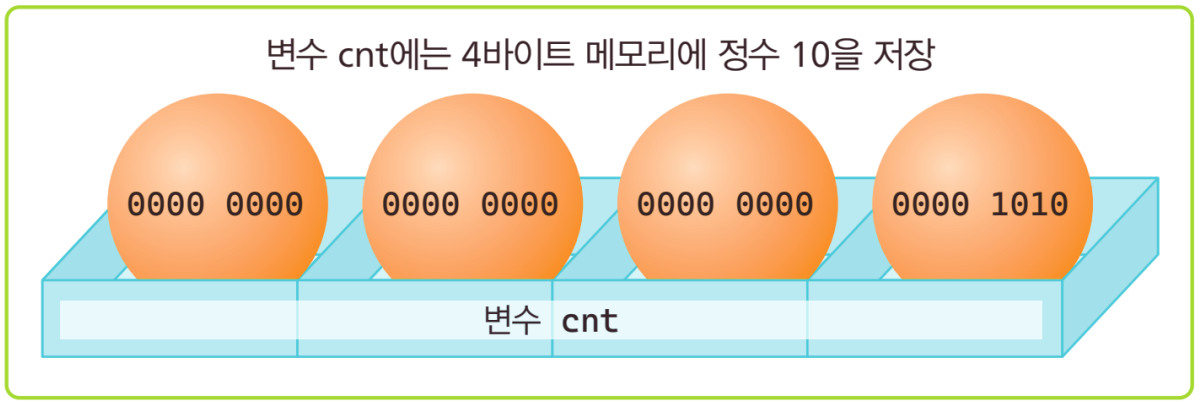
- 자료의 입출력을 텍스트 모드로 처리
  - 출력된 텍스트 파일
    - 텍스트 편집기로 그 내용을 볼 수 있으며
    - 텍스트 파일의 내용은 모두 지정된 아스키 코드와 같은 문자 코드값
- 함수 fprintf()를 이용
  - int 형 변수 cnt의 값을 파일 f에 출력하는 과정
  - 실제로 파일에 저장되는 자료는
    정수값 10에 해당하는 각 문자의 아스키 값
    - 각각의 문자 '1'과 '0'의 아스키 코드값이 저장

텍스트 파일 출력

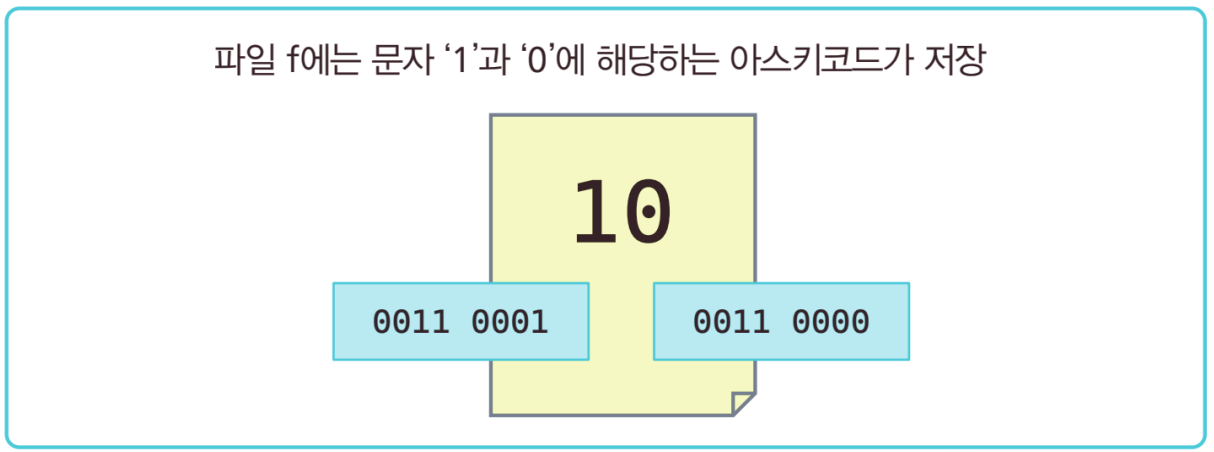
```
int cnt = 10;

fprintf(f, "%d", cnt);
```

메모리



파일



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 667쪽

함수 fwrite( )와 fread( )

- 이진 모드로 블록 단위 입출력을 처리
- 이진파일(binary file)
  - C 언어의 자료형을 모두 유지하면서 바이트 단위로 저장되는 파일
- 헤더파일 stdio.h 필요

```
int cnt = 10;
fwrite(&cnt, sizeof(int), 1, f);
fread(&cnt, sizeof(int), 1, f);
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 668쪽

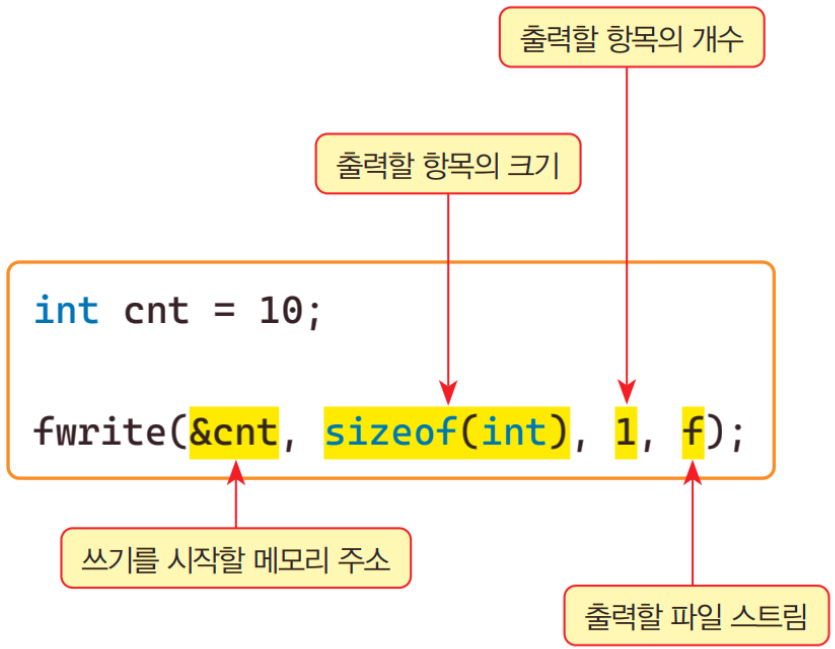
함수 fwrite( )

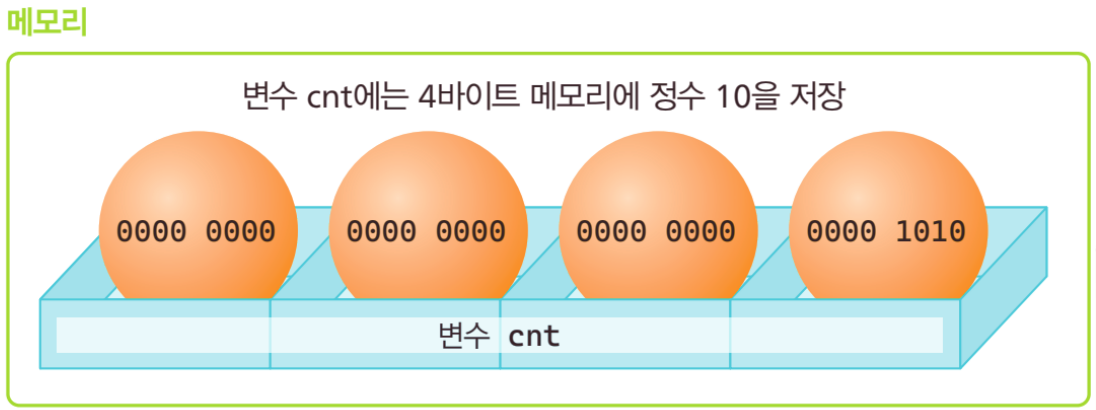
- 첫 번째 인자 ptr
  - 출력될 자료의 주소값
- 두 번째 인자 size
  - 출력될 자료 항목의 바이트 크기
- 세 번째 인자
  - 출력될 항목의 개수
- 마지막 인자
  - 출력될 파일 포인터
- 파일 f에 ptr에서 시작해서 size*n 바이트만큼의 자료를 출력
  - 반환값은 출력된 항목의 개수

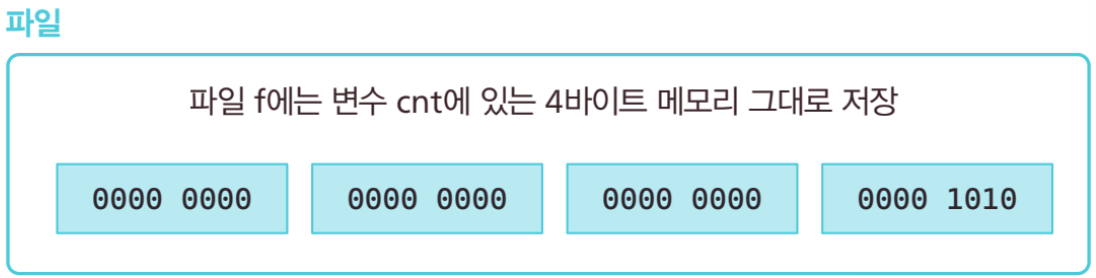
```
int cnt = 10;
fwrite(&cnt, sizeof(int), 1, f);
fread(&cnt, sizeof(int), 1, f);
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 668쪽

함수 fwrite( )에 의한 이진 파일 출력

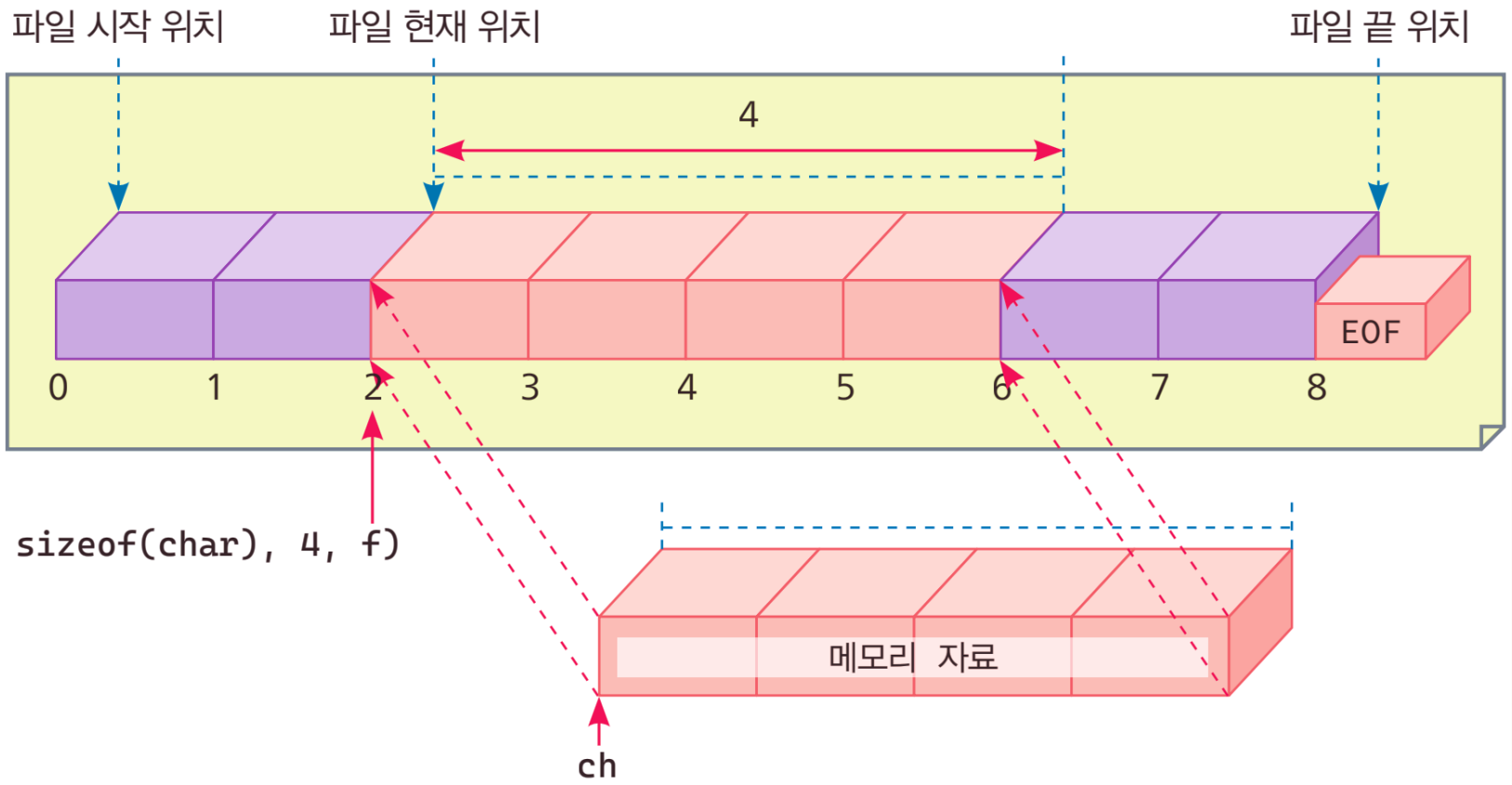






[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 668쪽

함수 fwrite( ) 여러 블록 쓰기



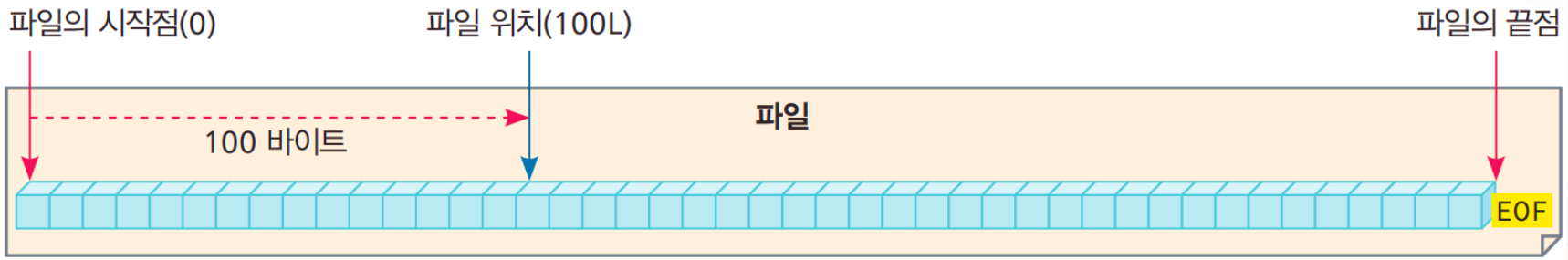
[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 669쪽

03

파일 접근 처리

파일 지시자(file indicator)

- 바이트 단위로 파일 내부 위치를 나타내는 값
  - '파일 표시자'라고도 부름
  - 파일의 시작점에서 파일 위치는 0이며 1바이트마다 1씩 증가
  - 파일 마지막에는 마지막임을 알리는 EOF(End Of File) 표시
- 파일을 열면
  - 파일 위치(file position)는 항상 파일의 시작 부분을 가리킴



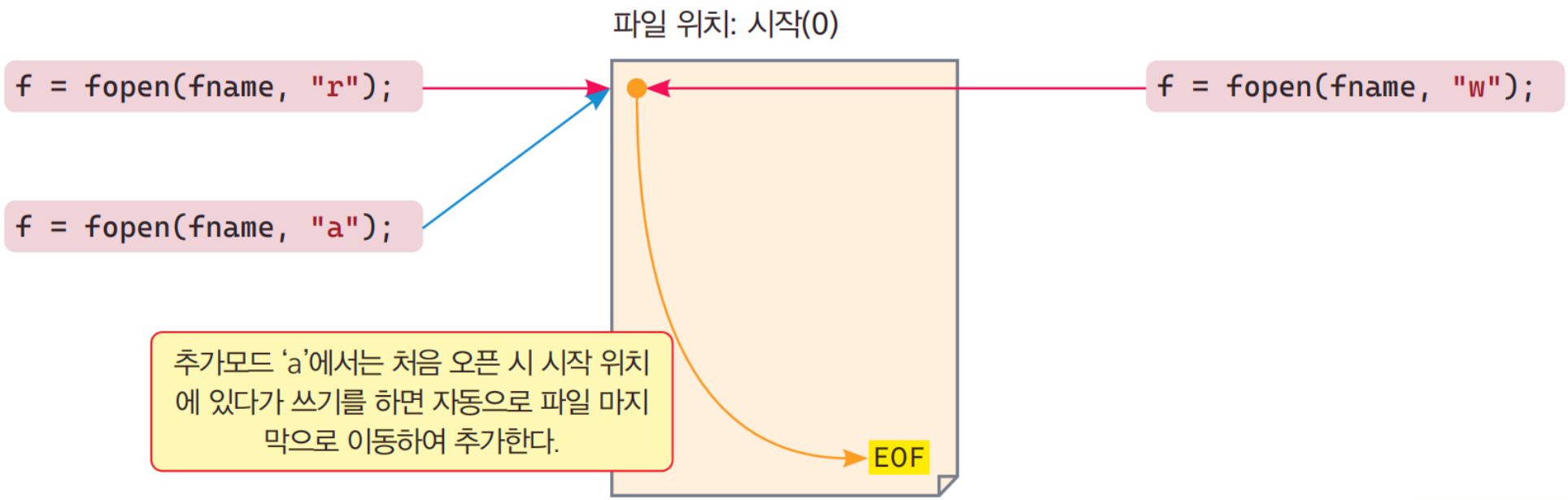
[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 675쪽

파일 스트림 연결 시 파일 위치

- 파일을 처음으로 열면 모드에 관계없이 파일 위치는 모두 0
- 파일 모드가 추가(a)
  - 파일을 처음 열면 파일 위치는 0
    - 자료를 파일에 쓰면 자동으로 파일 위치가 마지막으로 이동되어 추가
  - 파일 위치를 임의로 이동하였다면
    - 파일의 마지막으로 이동하여 추가

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 656쪽

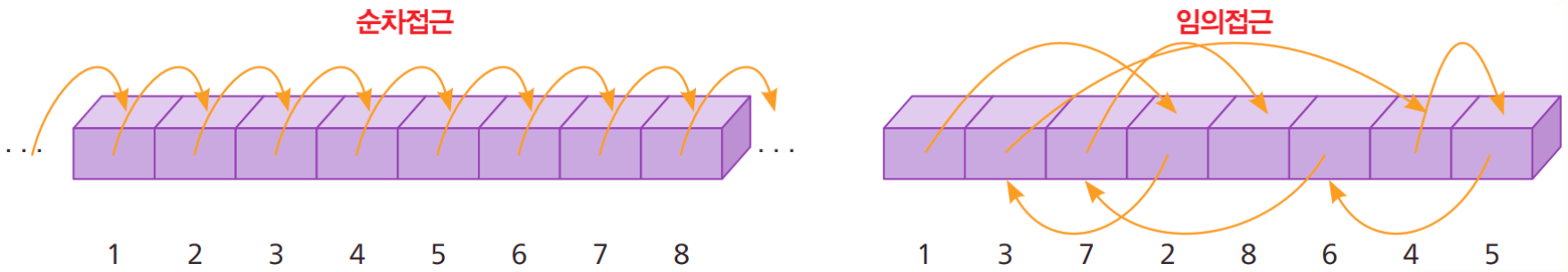
파일 스트림 연결 시 처음 파일 위치



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 675쪽

순차 접근과 임의 접근

- 순차적 접근(sequential access)
  - 파일 위치를 처음부터 하나씩 증가 하면서 파일을 참조하는 방식
- 임의 접근(random access)
  - 파일의 어느 위치든 바로 참조하는 방식
  - 관련 함수 fseek( ), ftell( ), rewind( ) 등을 활용하여 접근



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 676쪽

함수 fseek(): 파일의 임의 접근을 처리

- 파일 위치를 자유 자재로 이동, 헤더파일 stdio.h 필요
- 첫번째 인자
  - 파일 포인터
- 두 번째 인자: 흔히 오프셋(offset)이라 부름
  - long 유형으로 기준점으로부터 떨어진 값
- 세 번째 인자
  - 오프셋을 계산하는 기준점
  - 정수형 기호 상수로 다음 세 가지 중의 하나

함수 fseek( ) 함수 원형

함수 fseek() 함수원형

```
int fseek(FILE * _File, long _Offset, int _Origin);
```

함수 fseek()는 파일 _File의 기준점 _Origin에서 _Offest만큼 파일 포인터를 이동하는 함수 , 성공하면 0을 반환하며 실패하면 0이 아닌 정수를 반환

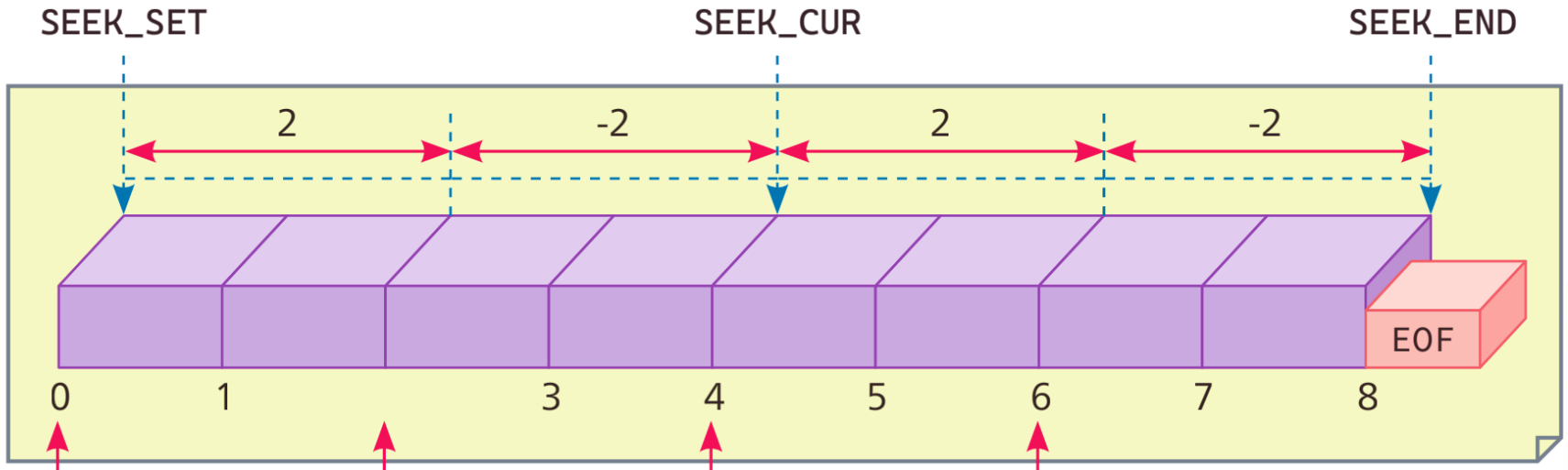
```
fseek(f, 0L, SEEK_SET);
fseek(f, 100L, SEEK_CUR);
fseek(f, -100L, SEEK_END);
```

기호	값	의미
SEEK_SET	0	파일의 시작 위치
SEEK_CUR	1	파일의 현재 위치
SEEK_END	2	파일의 끝 위치

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 676쪽

함수 fseek(f, 0L, SEEK_SET)

파일의 맨 처음으로 이동




```
fseek(f, 2L, SEEK_SET)
fseek(f, -2L, SEEK_CUR)
```

```
fseek(f, 2L, SEEK_CUR)
fseek(f, -2L, SEEK_END)
```

```
rewind(f)
fseek(f, 0L, SEEK_SET)
```

```
fseek(f, 0L, SEEK_CUR)
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 678쪽

함수 ftell( )과 rewind( )

함수 ftell( )

- 인자인 파일의 파일 위치를 반환

함수 rewind( )

- 파일 위치를 무조건 가장 앞으로 이동

함수	기능
int fseek(FILE *, long offset, int pos)	파일 위치를 세 기준점(pos)으로부터 오프셋(offset)만큼 이동
long ftell(FILE *)	파일의 현재 파일 위치를 반환
void rewind(FILE *)	파일의 현재 위치를 0 위치(파일의 시작점)로 이동

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 677쪽

파일열기 종류(모드)

- 함수 fopen( )과 fopen_s( )의 인자인 파일열기 종류(모드)
  - 텍스트 파일인 경우 “r”, “w”, “a”, “r+”, “w+”, “a+” 등
    - 읽기모드 r
      - 읽기가 가능한 모드이며, 쓰기는 불가능
    - 쓰기모드 w
      - 파일 어디에든 쓰기가 가능한 모드이나 읽기는 불가능
    - 추가모드 a
      - 파일 중간에 쓸 수 없으며 마지막에 추가적으로 쓰는 것만 가능한 모드
      - 읽기는 불가능
      - 파일에 쓰는 내용은 무조건 파일 마지막에 추가

파일모드에서 +의 삽입 1/2

- 수정(update) 모드 의미
  - 원래의 모드에서 읽기 또는 쓰기가 추가되는 모드
    - 모드 간의 전환이 가능
- 파일모드 r+
  - 처음에 읽기 모드로 파일을 열어 쓰기 모드로 전환 가능
- 파일모드 w+
  - 처음에 쓰기 모드로 파일을 열어
    - 필요하면 읽기 모드로 전환 가능
  - 만일 파일이 존재한다면 이전의 내용은 모두 사라짐

파일모드에서 +의 삽입 2/2

- 파일모드 a+
  - 처음에 추가 모드로 파일을 열어
    - 필요하면 읽기 모드로 전환 가능
- 수정모드에서 모드전환
  - 추가모드와 읽기모드 간, 쓰기모드와 읽기 모드간의 전환이 가능
  - 파일모드 전환 사이에는
    fseek( ) 또는 rewind( )와 같은 함수 호출이 필요
- 이진 파일을 위한 파일 열기 모드
  - 문자 'b'를 추가 가능, "rb+"

정 리 하 기

정리하기

- 텍스트 파일은 내용이 아스키 코드(ascii code)와 같은 문자 코드값으로 저장되는 파일이다.
- 이진 파일은 메모리 자료 내용이 어떤 변환도 거치지 않고 그대로 파일에 이진 형태(binary format)로 저장되는 파일이다.
- 함수 fopen( )으로 파일 스트림 열고 fclose( )로 파일 스트림을 닫는다.
- 함수 fprintf( ) 등으로 텍스트 파일을 작성하고 fwrite( )로 이진파일을 작성한다.
- 파일 내부에서 파일 지정자 위치는 함수 fseek( )으로 이동하며, 함수 ftell( )는 파일의 현재 위치를 반환하고 함수 rewind( )는 파일 위치를 무조건 가장 앞으로 이동한다.

14강

다음시간안내

동적 메모리