

PART 14 ► 프로그래밍 실습

1. 도서에 대한 데이터 입력과 도서에 대한 열람 기능을 제공하는 간단한 도서관리 시스템을 구현해보자. 이 시스템은 도서 입력과 도서 열람을 위한 두 개의 프로그램으로 구성된다.

① 헤더 파일 book.h 작성

12장에서 정의한 struct book을 확장해서 book.h에 새로운 구조체를 작성한다.

② 도서 입력 프로그램

도서관에 새로운 책이 들어올 때마다 책에 대한 정보를 입력받아 파일에 저장한다. 책에 대한 정보는 이름, 출판일, 저자, 가격 등이며 책이 들어오는 순서에 따라 일련번호를 붙인다.

- ⓐ book.h 헤더 파일을 include 한다.
- ⓑ FILE 포인터, struct book 타입의 필요한 변수를 선언한다.
- ⓒ fopen() 함수를 이용하여 도서 정보를 저장하기 위한 파일을 연다.
- ⓓ scanf() 함수를 이용하여 새로 들어온 책에 대한 정보를 입력 받아 구조체 변수에 저장한다.
- ⓔ fwrite() 함수를 이용하여 구조체 변수에 있는 책 정보를 파일에 저장한다.
- ⓕ ㉔부터 ㉔의 과정을 더 이상 입력이 없을 때까지 반복한다.

③ 도서 열람 프로그램

책의 일련번호를 입력받아 해당 책 정보를 찾아 출력해 주는 프로그램을 작성한다.

- ⓐ book.h 헤더 파일을 include 한다.
- ⓑ FILE 포인터, struct book 타입의 필요한 변수를 선언한다.
- ⓒ fopen() 함수를 이용하여 도서 정보를 저장하기 위한 파일을 연다.
- ⓓ 책 일련번호를 입력받고 해당 책의 레코드 위치를 계산한다.
- ⓔ fread() 함수를 이용하여 해당 레코드를 읽어서 구조체 변수에 저장한다.
- ⓕ 읽어 온 책에 대한 정보를 서식에 맞게 출력한다.
- ⓖ ㉔부터 ㉔까지 과정을 반복한다.

book.h

```
#define MAXBOOK 100          // 최대 도서 수

struct book {
    char author[20];
    char title[50];
    char publisher[20];
    int year;                  // 단순화하기 위해 출판일은 년도로만 표시함
    int price;
};
```

book.c

```
#include <stdio.h>
#include "book.h"

int bookno=0;
struct book info;

void author_search(FILE *fp, char *author)
{
    int i;
    for (i = 0; i < bookno; i++) {
        fread(&info, sizeof(struct book), 1, fp);
        if (!strcmp(info.author, author))
            printf("%s, %s, %s, %d \n", info.author, info.title,
                info.publisher, info.year);
    }
}

void title_search(FILE *fp, char *title)
{
    int i;
    for (i = 0; i < bookno; i++) {
        fread(&info, sizeof(struct book), 1, fp);
        if (!strcmp(info.title, title))
            printf("%s, %s, %s, %d \n", info.author, info.title,
                info.publisher, info.year);
    }
}

void getbookinfo()
```

```

{
    struct book info;
    printf("\n저자: ");
    gets(info.author);
    getchar(); // skip newline
    printf("제목: ");
    gets(info.title);
    getchar(); // skip newline
    printf("출판사: ");
    gets(info.publisher);
    getchar(); // skip newline
    printf("출판년도: ");
    scanf("%d", &info.year);
    getchar(); // skip newline
    printf("가격: ");
    scanf("%d", &info.price);
    getchar(); // skip newline
}

int main()
{
    int menu, end = 0;
    char author[20], title[50];
    FILE *fp;

    while(end == 0) {
        printf("\n1. 도서 입력\n");
        printf("2. 저자별 검색\n");
        printf("3. 제목 검색\n");
        printf("4. 끝\n");
        printf("메뉴를 선택하세요: ");
        scanf("%d", &menu);
        getchar(); // skip newline

        switch (menu)
        {
            case 1: fp = fopen("books", "ab+");
                    getbookinfo();
                    fwrite(&info, sizeof(struct book), 1, fp);
                    bookno++;
                    fclose(fp);
                    break;
            case 2: fp = fopen("books", "rb");
                    printf("저자 이름: ");
                    gets(author);
                    author_search(fp, author);
                    fclose(fp);
                    break;
        }
    }
}

```

```

        case 2: fp = fopen("books", "rb");
                printf("저자 이름: ");
                gets(author);
                author_search(fp, author);
                fclose(fp);
                break;
        case 3: fp = fopen("books", "rb");
                printf("도서 제목 : ");
                gets(title);
                title_search(fp, title);
                fclose(fp);
                break;

        case 4: end = 1;
                break;

        default: printf("잘못된 메뉴입니다. 다시 입력하세요\n");

    }
}
return 0;
}

```