

# C언어 기반 도서관리 프로그램

기술 명세서 (Tech Spec)

## 1. ☐ 프로젝트 개요

- **프로젝트명:** C언어 기반 도서관리 시스템 (Library Management System)
- **개발 환경:** C언어 (Standard C)

### 목표

- struct (구조체)를 사용한 효율적인 데이터 모델링
- 바이너리 파일 입출력 (fwrite, fread)을 통한 데이터의 영구적 보존
- 포인터, 배열, 문자열 함수 등 C언어 핵심 문법의 종합적 활용
- 일반 사용자 / 관리자 모드 분리를 통한 다중 사용자 환경 구현

## 2. ☐ 핵심 데이터 구조 (struct)

모든 데이터의 뼈대가 되는 구조체입니다.

### 2.1. 도서 구조체 (struct Book)

도서 한 권의 정보를 저장합니다.

Listing 1: book.h

```
// book.h ( )
#define MAX_TITLE 100
#define MAX_AUTHOR 50
#define MAX_GENRE 30

struct Book {
    int id; // ID ( )
    char title[MAX_TITLE]; // 
    char author[MAX_AUTHOR]; //
    char genre[MAX_GENRE]; //
    int status; // (0: , 1: )
    int borrowedUserID; // ( ) ID
};
```

## 2.2. 사용자 구조체 (struct User)

회원 한 명의 정보를 저장합니다.

Listing 2: user.h

```
// user.h ( )
#define MAX_USERNAME 50
#define MAX_PASSWORD 50
#define MAX_NAME 30
#define MAX_BORROW 5 // 1

struct User {
    int userID;                      // ID
    char username[MAX_USERNAME];      // ID
    char password[MAX_PASSWORD];     // PW
    char name[MAX_NAME];            //
    int borrowedBookIDs[MAX_BORROW]; // ID
    int borrowedCount;               //
};
```

### 3. ☐ 데이터 영구 저장 (파일 입출력)

#### 전략

텍스트 파일(.txt)이 아닌 **바이너리 파일(.dat)**을 사용합니다.

**이유:** struct 데이터를 '덩어리'째로 읽고 쓸 수 있어 속도가 빠르고 파싱(Parsing)이 필요 없어 코드가 간결하며 데이터 관리가 매우 효율적입니다.

#### 대상 파일

- books.dat: 모든 struct Book 정보를 배열 형태로 저장
- users.dat: 모든 struct User 정보를 배열 형태로 저장

#### 핵심 C언어 함수

- fopen("rb"): 프로그램 시작 시, 바이너리 읽기 모드로 파일을 열어 데이터를 메모리(배열)로 로드.
- fopen("wb"): 프로그램 종료 시, 바이너리 쓰기 모드로 파일을 열어 메모리(배열)의 데이터를 파일에 저장.
- fread(buffer, sizeof(struct Book), count, fp): 파일에서 구조체 데이터를 읽어옴.
- fwrite(buffer, sizeof(struct Book), count, fp): 구조체 데이터를 파일에 씁.
- fclose(fp): 파일 핸들러 해제.

---

### 4. ☐ 기능별 상세 구현 (함수 명세)

프로그램의 실제 동작을 함수 단위로 나눕니다.

#### 4.1. 메인 프로그램 (main.c)

- int main():
  - loadAllData(): 프로그램 시작 시 books.dat, users.dat 파일에서 모든 데이터를 전역 배열로 불러옵니다.
  - showMainMenu(): "1. 사용자", "2. 관리자", "3. 종료" 메뉴 표시.
  - 사용자 입력에 따라 userMode() 또는 adminMode() 함수 호출.
  - (3. 종료 선택 시) saveAllData(): 전역 배열의 모든 데이터를 다시 파일에 저장하고 프로그램을 종료 합니다.

#### 4.2. ☐ 일반 사용자 모드 (user.c)

- void userMode(): 사용자 모드의 메인 메뉴 (회원가입, 로그인, 종료) 표시.
- void signUp():
  - 사용자에게 username, password, name을 입력받고 중복 검사 후 g\_users 배열에 추가합니다.

- **핵심 C언어 함수:** scanf, strcpy, strcmp
- **int login():**
  - ID와 PW가 일치하는 사용자를 찾아 userID를 반환합니다.
  - **핵심 C언어 함수:** strcmp
- **void searchBook():**
  - "1. 제목 검색", "2. 저자 검색" 메뉴 제공.
  - **핵심 C언어 함수:** strstr (문자열 내 포함 여부 검사)
- **void loanBook(int loggedInUserID):**
  - 대출 가능 상태와 본인의 대출 한도를 확인 후 대출 처리합니다.
  - **핵심 C언어 개념:** 포인터 (구조체 배열 요소 수정), 배열 관리
- **void returnBook(int loggedInUserID):**
  - 반납할 도서를 찾아 상태를 변경하고 내 대출 목록에서 제거합니다.
  - **핵심 C언어 개념:** 배열 요소 삭제
- **void showMyInfo(int loggedInUserID):**
  - 나의 정보와 대출 중인 도서 목록을 출력합니다.
  - **핵심 C언어 개념:** 데이터 크로스-참조

#### 4.3. ☐ 관리자 모드 (admin.c)

- **void adminMode():** 관리자 로그인 및 메인 메뉴 표시.
- **void manageBooks():** 도서 CRUD 기능 (추가, 조회, 수정, 삭제).
- **void manageUsers():** 회원 CRUD 기능 (조회, 수정, 삭제).
- **void viewLoanStatus():**
  - 대출 중인 도서와 대출한 회원의 이름을 연결하여 출력합니다.
  - **핵심 C언어 개념:** 고급 데이터 크로스-참조

#### 4.4. ☐ 유ти리티 및 시각화 (utils.c)

- **void displayBookCover(struct Book\* book):**
  - 도서 검색 결과나 상세 정보 조회 시, 책 표지를 ASCII 아트로 시각화하여 출력합니다.
  - 책 제목 길이를 계산하여 박스 내 **중앙 정렬**하는 알고리즘을 구현합니다.
  - **출력 예시:**

```
|-----|
|           |
|       Sunday Morning   |
|       Breakfast        |
|           |
|-----|
[ID: 101]
```

- **핵심 C언어 개념:** `strlen`, 반복문을 이용한 공백(Padding) 계산

© 2025 Library Management System. All rights reserved.