

Final Report

도서 관리 시스템

과 목	임베디드시스템S/W설계
담당교수	김태석 교수님
학 과	컴퓨터정보공학부
학 번	2021202058
성 명	송채영
날 짜	2024. 06. 02 (일요일)



1. 요약

Kernel function 수	사용한 Task의 수	Semaphore 예제 수	Message queue 예제 수
2	5	3	3

위 표를 요약하여 설명하며, 자세한 설명은 본문에서 진행하겠다.

우선 2개의 kernel function을 구현하였다. WriteLoanRecord 함수는 OS_CORE.C에 구현하였으며, 대출 정보를 파일에 기록하는 역할을 한다. SetReturnDueDate 함수는 OS_TIME.C에 구현하였으며, 대출된 책의 반납 기한을 관리한다.

Task는 5개를 사용하였다. 먼저 Task 1은 사용자 인터페이스로 사용자에게 메뉴를 표시하고 사용자 입력을 처리한다. Task 2는 사용자가 도서를 대출할 때 호출된다. Task 3은 사용자가 도서를 반납될 때 호출된다. Task 4는 책을 검색하면 책에 대한 정보를 출력한다. Task 5는 도서를 예약할 때 호출하며 도서를 대출하거나 반납할 때 발생하는 알림을 처리한다.

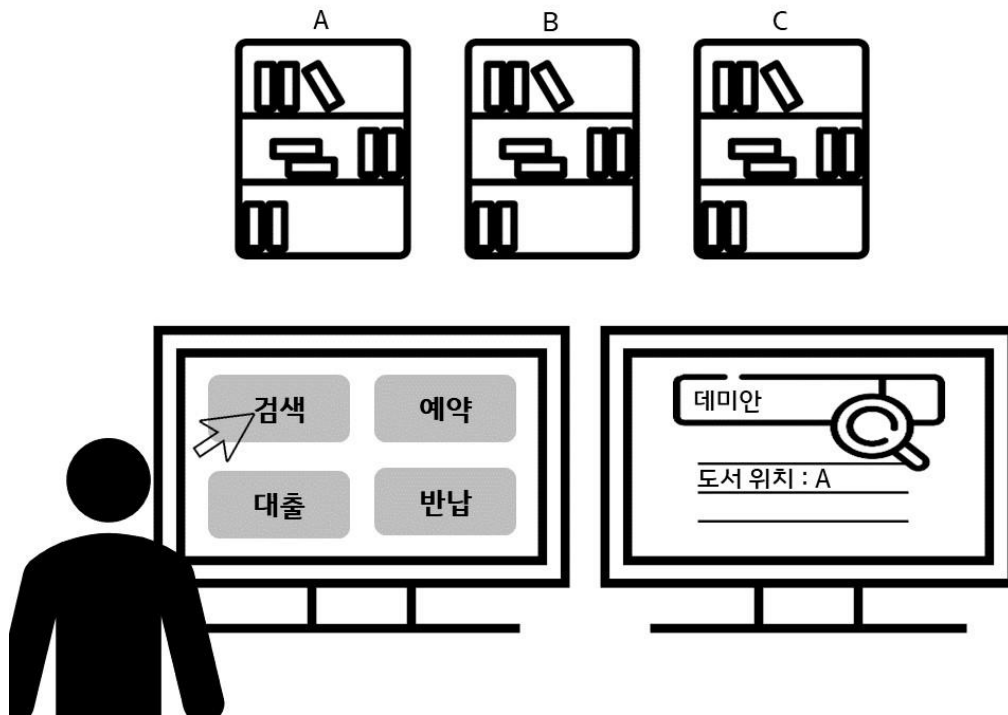
Semaphore는 Task 2와 Task 3 그리고 Task 4에서 사용했다. 먼저 첫 번째 경우는 Task 2에서 대기중인 사용자가 도서를 대출하려고 할 때 Semaphore를 획득하여 다른 task가 도서를 대출하는 동안 동시 접근이 안되도록 한다. Task 3에서는 사용자가 도서를 반납할 때 실행되며 Task 4에서는 사용자가 도서를 검색할 때 실행된다.

Message queue는 Task 2, Task 3, Task 4에서 사용되어 Task 5로 전달한다. 먼저 Task 2에서는 대출할 도서가 이미 대출 중이거나 존재하지 않는 경우 메시지를 전달한다. Task 3에서는 도서가 반납되었을 때 메시지를 전달한다. Task 4에서는 도서를 예약한 후 예약이 완료되었음을 메시지에 전달한다. 이후 Message queue를 사용하여 Task 5로 전달한다.

2. 본문

A) 프로젝트 개요

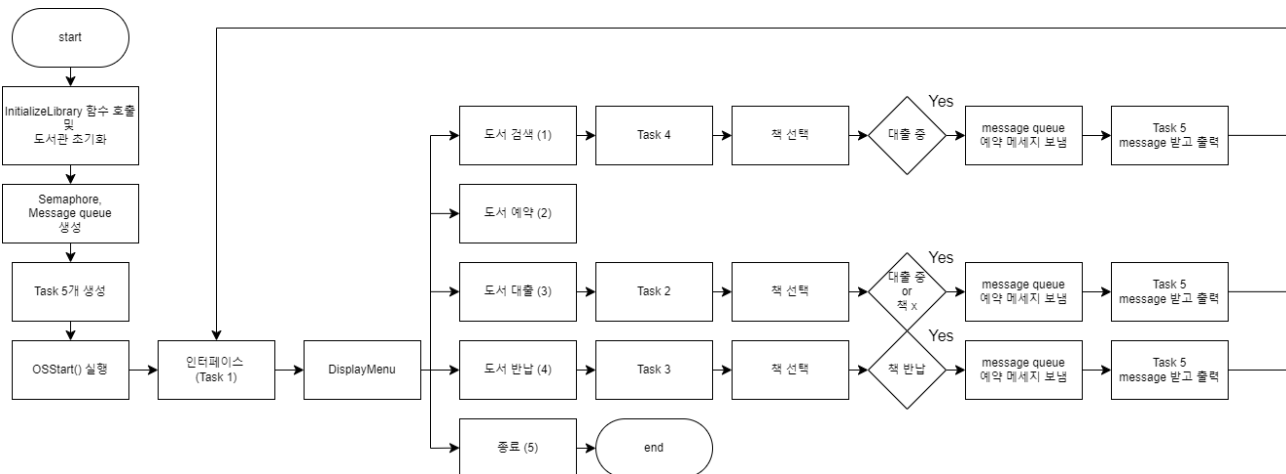
아래는 프로젝트 전반을 설명하는 그림이다.



μ C/OS를 이용해 임베디드 시스템을 구현하는 것이 목적이며 프로젝트의 주제는 도서관 관리 시스템이다. 도서관 관리 시스템은 사용자가 도서를 대출, 반납, 검색, 및 예약을 할 수 있다. 사용자는 인터페이스를 통해 각각의 기능을 선택할 수 있다. 시스템은 도서 대출 및 반납, 그리고 검색 처리를 동기화하여 동시에 여러 사용자가 동시에 책을 대출하거나 반납하는 것을 방지할 수 있다. 또한 책의 위치를 검색하고 책이 대출 중일 때 예약하는 기능을 추가하여 사용자가 원하는 책을 찾을 수 있고 예약할 수 있다. 책장은 A, B, C가 있으며 사용자가 원하는 책이 있을 경우 책의 위치(책장)를, 없을 경우 해당 책이 없음을 알려준다. 또한 도서 대출 및 반납 정보를 기록으로 남긴다.

B) 프로젝트의 구조

아래는 프로젝트의 전체적인 구조를 나타내는 flow chart이다.



프로그램이 시작되면 main 함수에서 OSInit을 한다. 이후 InitializeLibrary 함수를 호출하여 몇 가지 샘플 책을 포함하는 도서관을 초기화한다. OSSemCreate 함수와 OSQCreate 함수를 통해 semaphore와 message queue를 생성하고 OSTaskCreate 함수를 통해 다섯 개의 Task(Task 1 ~ Task 5)를 생성하고 우선순위를 준다. OSStart 함수를 통해 시작하며 Task 1의 우선순위가 가장 높기 때문에 Task 1부터 실행된다. Task 1은 사용자 인터페이스를 표시하고 사용자의 선택에 따라 해당하는 Task를 수행한다. Display menu를 통해 고를 수 있으며 도서 검색, 대출, 반납 및 종료에 대한 선택이 가능하다. 우선 도서 검색은 Task 4에 해당하며 사용자가 입력한 책 제목을 검색하여 대출 중인 경우 message queue를 활용하여 메시지를 전달하고 이를 Task 5가 수신하여 출력한다. 도서 예약의 기능은 따로 없으며 다른 Task에서 예약 기능이 포함되어 있다. 도서 대출은 Task 2에 해당하며 사용자가 선택한 책을 대출하고 이미 대출 중이거나 해당하는 책이 없을 경우 message queue를 활용하여 메시지를 전달하고 이를 Task 5가 수신하여 출력한다. 도서 반납은 Task 3에 해당하며 사용자가 선택한 책을 반납하고 반납이 됐다는 것을 message queue를 활용하여 메시지를 전달하고 이를 Task 5가 수신하여 출력한다. 5번을 입력 시 더 이상 displaymenu가 나오지 않는다. 각 task의 할 일이 끝나면 다시 Task 1로 돌아가 displaymenu가 출력되며 사용자가 종료할 때까지 이를 반복한다.

C) 각 Task의 정의

총 5개의 task가 있으며, task 1은 사용자 인터페이스, task 2는 도서 대출 처리, task 3은 도서 반납 처리, task 4는 책 위치 검색, task 5는 책 예약 처리에 해당한다.

우선 task1은 사용자 인터페이스로 사용자와 상호작용을 담당하는 task이다. task 1은 무한 루프를 통해 사용자에게 메뉴를 표시하고 사용자로부터 입력을 받는다. 사용자가 입력한 옵션에는 도서 대출, 도서 반납, 도서 검색, 도서 예약, 종료 등이 있으며 해당 옵션에 따라

해당하는 Task를 수행한다. task 1의 경우 사용자와 상호작용을 담당하고 사용자의 입력을 받고 처리하는 것이 시스템의 핵심 순위이므로 우선순위를 1로 설정하였다. task 1을 제외한 나머지 우선순위는 크게 상관이 없으므로 task의 순서대로 설정하였다.

task 2는 도서 대출 처리를 담당한다. 사용자가 대출을 요구할 때 semaphore를 획득하여 해당 작업을 수행하고 작업이 완료되면 semaphore를 해제하여 다음 요청이 처리될 수 있도록 한다. task 2는 입력 받은(대출하려는) 책을 찾고 대출이 가능할 경우 대출을 완료한다. 대출 시 SetReturnDueDate 커널 함수를 활용하여 대출 시간을 기록한다. 이후 WriteLoanRecord 커널 함수를 활용해 대출 기록을 남긴다. 이때 책이 이미 대출 중이거나 책이 존재하지 않는 경우 예약이 완료되었음을 알리는 메시지를 message queue를 활용해 보낸다. task 2의 우선 순위는 2로 설정하였다.

task 3은 도서 반납 처리를 담당한다. 사용자가 반납을 요구할 때 semaphore를 획득하여 해당 작업을 수행하고, 작업이 완료되면 semaphore를 해제하여 다음 요청이 처리될 수 있도록 한다. task 3은 반납하려는 책을 찾고, 책이 존재하는 경우 반납을 완료한다. 이때 반납이 일정 시간 늦었을 경우 경고 메시지를 출력한다. 책이 존재하지 않거나 현재 빌린 상태가 아닐 경우 적절한 출력문을 출력하고 책이 반납됐을 때 반납이 완료되었음을 알리는 메시지를 message queue를 활용해 보낸다. 우선 순위를 3으로 설정하였다.

task 4는 책의 위치를 검색하는 일을 처리한다. 사용자가 검색하려는 책을 입력 했을 때 해당 책이 있을 경우 책에 대한 정보를 출력하며, 빌린 여부에 따라 이미 대출중인 경우 예약이 완료되었다는 메시지를 message queue를 활용하여 보낸다. 대출 중이 아닐 경우 task 2를 재개한다. 책을 찾지 못했을 경우 적절한 출력문을 출력한다. 우선 순위를 4로 설정하였다.

task 5는 message queue로부터 수신된 메시지를 받아와 출력하며, 책을 예약이 가능함을 알리는 일을 한다. 사용자가 특정 책을 대출할 수 없는 경우 예약을 요청하면 해당 책을 대출 가능할 때까지 대기열에 넣어두고 대출이 가능해질 경우 사용자에게 알려준다. 우선 순위를 5로 설정하였다.

OSTaskSuspend와 OSTaskResume 함수를 통해 각 task의 상황에 맞게 우선 순위를 조정할 예정이다.

D) task 간 semaphore와 message queue의 활용방안

Semaphore는 Task 2와 Task 3 그리고 Task 4에서 사용했다. 먼저 첫 번째 경우는 Task 2에서 대기중인 사용자가 도서를 대출하려고 할 때 Semaphore를 획득하여 다른 task가 도서를 대출하는 동안 동시 접근이 안되도록 한다. Task 3에서는 사용자가 도서를 반납할 때 실행되며 Task 4에서는 사용자가 도서를 검색할 때 실행된다. OSSemPend와 OSSemPost를 사용하여 세마포어를 획득 및 해제해주었다.

Message queue는 Task 2, Task 3, Task 4에서 사용되어 Task 5로 전달한다. 먼저 Task 2에서는 대출할 도서가 이미 대출 중이거나 존재하지 않는 경우 메시지를 전달한다. Task 3에서는 도서가 반납되었을 때 메시지를 전달한다. Task 4에서는 도서를 예약한 후 예약이 완료되었음을 메시지에 전달한다. 이후 Message queue를 사용하여 Task 5로 전달한다. OSQPost와 OSQAccept를 활용하여 메시지를 전달 및 받았다.

E) 제안서 대비 변경사항

먼저 kernel function을 제안서 때는 OS_TASK.C 코드에서의 OSTaskCreate() 부분을 수정한다고 하였다. 기존에는 OSTaskCreate() 함수를 수정하여 도서 대출 및 반납 task의 우선 순위를 설정하고, 대출 및 반납 task 내에서 도서의 대출 상태를 확인하는 동기화 기능을 추가하려고 하였지만, 이는 task 내에서 구현이 가능했기 때문에 수정하게 되었다. 수정한 부분은 우선 OS_TIME.C 코드에 SetReturnDueDate 함수를 구현하여, 도서 대출 시 반납 기한을 설정하여 대출한 사용자가 책을 반납 시 메시지를 출력하도록 하였다. 구현 시 동작함을 보이기 위해 도서 반납 기한을 10으로 설정하였고 이는 10초 이내에 도서를 반납해야 함을 의미한다. 현재 시간에 도서 반납 기한을 설정하기 위해 OSTimeGet 함수를 사용하여 현재 시간을 가져왔고 10초를 계산하여 더해주었다. 또한 OS_CORE.C 코드에 도서 대출 및 반납 정보를 파일에 기록하는 기능을 하는 WriteLoanRecord 함수를 구현하였다. LOAN_RECORD_FILE에 대출 기록을 저장할 파일을 정의하고 도서 제목과 동작 그리고 시간을 기록하였다. 이를 OS_CORE.C에 구현한 이유는 시스템의 코어 기능을 관리하는 파일이기 때문에 제일 적절하다고 생각했다. 또한 기존에는 task 5를 예약을 하는 기능을 구현하겠다고 했지만, 최종적으로는 예약 메시지를 받는 종착지의 역할을 하는 task가 되었다. 하지만 구현 방법은 제안서의 내용과 동일하다.

F) 동작 예시

책 정보는 아래와 같이 존재한다.

```
// Initialize the library with some sample books
void InitializeLibrary()
{
    strcpy(library[0].title, "Demian");
    strcpy(library[0].author, "Hermann Hesse");
    library[0].shelf = 'A';
    library[0].is_borrowed = 0;

    strcpy(library[1].title, "Animal Farm");
    strcpy(library[1].author, "George Orwell");
    library[1].shelf = 'B';
    library[1].is_borrowed = 0;

    strcpy(library[2].title, "1984");
    strcpy(library[2].author, "George Orwell");
    library[2].shelf = 'C';
    library[2].is_borrowed = 0;
}
```

C:\SOFTWARE\PROJECT>test.exe

```
--- Library System ---
1. Search for a book
2. Reserve a book
3. Borrow a book
4. Return a book
5. Exit
Select an option: You selected book search.
Enter the title of the book to search:
1984
Book found!
Title: 1984
Author: George Orwell
Shelf: C
Borrowed: No
Borrowing book: 1984

--- Library System ---
1. Search for a book
2. Reserve a book
3. Borrow a book
4. Return a book
5. Exit
Select an option: You selected book search.
Enter the title of the book to search:
1984
Book found!
Title: 1984
Author: George Orwell
Shelf: C
Borrowed: Yes
Alert: Reservation complete.
```

1984라는 책을 search 했을 때 책의 정보를 출력하며, 책이 대여되지 않았으므로 대여를 진행한다. 이후 해당 책을 search 했을 때 대여가 되어 있기 때문에 Reservation

complete가 뜨는 것을 확인할 수 있다.

```

--- Library System ---
1. Search for a book
2. Reserve a book
3. Borrow a book
4. Return a book
5. Exit
Select an option: You selected book borrowing.
Enter the title of the book to borrow:
Demian
Borrowing book: Demian

--- Library System ---
1. Search for a book
2. Reserve a book
3. Borrow a book
4. Return a book
5. Exit
Select an option: You selected book borrowing.
Enter the title of the book to borrow:
Demian
Book 'Demian' is not available for borrowing or already borrowed.
Reservation complete.
Alert: Book not available for borrowing or already borrowed

```

Demian이라는 책을 borrow 했을 때 책을 빌린 것을 확인할 수 있고 같은 책을 또 borrow 하려고 했을 때 책이 이미 빌려졌다는 문구와 예약이 완료되었다는 것을 알 수 있고 해당 문구가 alert 된(task 5로 잘 전달됨) 것을 확인할 수 있다.

```

--- Library System ---
1. Search for a book
2. Reserve a book
3. Borrow a book
4. Return a book
5. Exit
Select an option: You selected book returning.
Enter the title of the book to return:
Demian
Book 'Demian' was not returned on time.
Returning book: Demian
Alert: Book returned

--- Library System ---
1. Search for a book
2. Reserve a book
3. Borrow a book
4. Return a book
5. Exit
Select an option: You selected book returning.
Enter the title of the book to return:
1984
Book '1984' was not returned on time.
Returning book: 1984
Alert: Book returned

```

위에서 빌렸던 두 책을 반납하려고 하는 과정이다. 두 책 모두 반납기한이 지났기 때문에 Book was not returned on time이라는 출력문이 나온 것을 확인할 수 있다. 책이 성공적으로 반납된 후 Alert : Book returned라는 구문을 통해 task 5로 메시지가 잘

전달되었음을 확인할 수 있다.

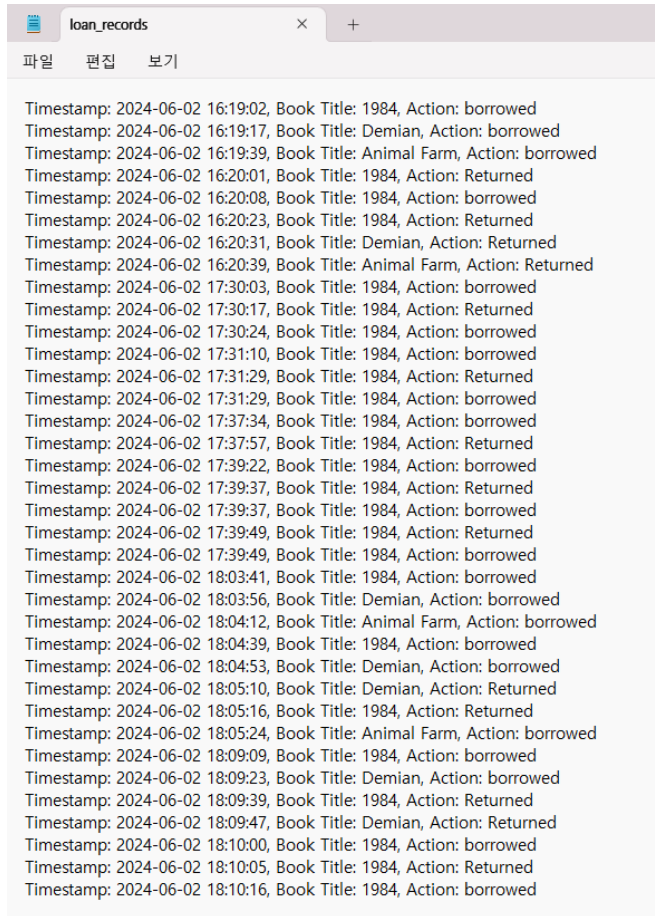
```
--- Library System ---
1. Search for a book
2. Reserve a book
3. Borrow a book
4. Return a book
5. Exit
Select an option: 1984
Book found!
Title: 1984
Author: George Orwell
Shelf: C
Borrowed: No
Borrowing book: 1984
You selected book returning.
Enter the title of the book to return:
1984
Returning book: 1984
Alert: Book returned

--- Library System ---
1. Search for a book
2. Reserve a book
3. Borrow a book
4. Return a book
5. Exit
Select an option: You selected book borrowing.
Enter the title of the book to borrow:
1984
Borrowing book: 1984
```

1984 책을 다시 빌린 후 10초 안에 반납 했을 때의 결과로 return 시 반납 기한이 지났다는 출력문이 없는 것을 확인할 수 있고, 반납된 책 역시 다시 빌릴 수 있음을 확인하였다.

```
--- Library System ---
1. Search for a book
2. Reserve a book
3. Borrow a book
4. Return a book
5. Exit
Select an option: Exiting the menu...
```

5. Exit를 선택 시 더 이상 메뉴가 나오지 않음을 확인하였다.



```
loan_records
파일 편집 보기

Timestamp: 2024-06-02 16:19:02, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 16:19:17, Book Title: Demian, Action: borrowed
Timestamp: 2024-06-02 16:19:39, Book Title: Animal Farm, Action: borrowed
Timestamp: 2024-06-02 16:20:01, Book Title: 1984, Action: Returned
Timestamp: 2024-06-02 16:20:08, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 16:20:23, Book Title: 1984, Action: Returned
Timestamp: 2024-06-02 16:20:31, Book Title: Demian, Action: Returned
Timestamp: 2024-06-02 16:20:39, Book Title: Animal Farm, Action: Returned
Timestamp: 2024-06-02 17:30:03, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 17:30:17, Book Title: 1984, Action: Returned
Timestamp: 2024-06-02 17:30:24, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 17:31:10, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 17:31:29, Book Title: 1984, Action: Returned
Timestamp: 2024-06-02 17:31:29, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 17:37:34, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 17:37:57, Book Title: 1984, Action: Returned
Timestamp: 2024-06-02 17:39:22, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 17:39:37, Book Title: 1984, Action: Returned
Timestamp: 2024-06-02 17:39:37, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 17:39:49, Book Title: 1984, Action: Returned
Timestamp: 2024-06-02 17:39:49, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 18:03:41, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 18:03:56, Book Title: Demian, Action: borrowed
Timestamp: 2024-06-02 18:04:12, Book Title: Animal Farm, Action: borrowed
Timestamp: 2024-06-02 18:04:39, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 18:04:53, Book Title: Demian, Action: borrowed
Timestamp: 2024-06-02 18:05:10, Book Title: Demian, Action: Returned
Timestamp: 2024-06-02 18:05:16, Book Title: 1984, Action: Returned
Timestamp: 2024-06-02 18:05:24, Book Title: Animal Farm, Action: borrowed
Timestamp: 2024-06-02 18:09:09, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 18:09:23, Book Title: Demian, Action: borrowed
Timestamp: 2024-06-02 18:09:39, Book Title: 1984, Action: Returned
Timestamp: 2024-06-02 18:09:47, Book Title: Demian, Action: Returned
Timestamp: 2024-06-02 18:10:00, Book Title: 1984, Action: borrowed
Timestamp: 2024-06-02 18:10:05, Book Title: 1984, Action: Returned
Timestamp: 2024-06-02 18:10:16, Book Title: 1984, Action: borrowed
```

마지막으로 도서 대출 및 반납 정보의 기록이다. 기록은 지워지지 않고 계속 남는 것을 확인 할 수 있었다.