

CADT Cambodia Academy of
Digital Technology

IDT វិទ្យាស្ថានបច្ចេកវិទ្យាឌីជីថល
Institute of Digital Technology

Advance Programming

Direct by Dr.Kong Phutphalla

TA Mr Natt Korat

Group 7:

- 1.Khun Sophavisnuka*
- 2.Chhay Lymean*
- 3.Both Chealean*
- 4. Chea Kolthida*
- 5. Ek Sopheaksetheariny*

Table of contents

1. Introduction.....	1
i. Introduction	
ii. Background	
iii. Objective	
2. Solution	5
i. Data Structures and File Management	
ii. The system relies on three text files	
iii. Book Management Functions	
iv. Borrowing Management Functions	
vi. User Interface and Input Handling	
vii. Scalability and Extensibility	
3. Result	6
i. Book Management Functions	
ii. Borrowing Management Functions	
iii. User Interface and Input Handling	
iv. Scalability and Extensibility	
4. Conclusion	7

Introduction

1. Introduction

This C program is designed to manage the operations of a library, including viewing book information, adding new books, and allowing students to borrow books. The code utilizes various data structures and functions to handle the different functionalities of the library management system. The main functions include viewing books by name or ID, adding new books with details such as the book name, ID, type, and author, and allowing students to borrow books, including recording the student's name, the date of the borrowing, and the number of books borrowed. The program utilizes file handling to store and retrieve books and borrow information from text files, ensuring that the library's data is persistently stored and can be accessed across multiple sessions. The code provides a user-friendly interface with a menu-driven approach, allowing the user to navigate through the different functionalities of the library management system, and includes error handling and appropriate feedback to the user. Overall, this C code demonstrates the implementation of a comprehensive library management system, which can be further expanded and enhanced to meet the specific needs of a library's operations.

2. Background

The "Library-management.c" code presents a comprehensive library management system implemented in C programming language. This system aims to help an efficient and organized way for librarians to manage the library's book inventory and student book borrowing. The key features of the system include book management, where librarians can view the available books, search for books by name or ID, and add new books to the library's collection, with the book information stored in a text file for persistent storage. The system also allows students to borrow books by providing their personal details and the book ID they wish to borrow, with the program keeping track of the borrowed books and the number of books each student has borrowed. The system provides a clear and intuitive command-line interface, allowing users to navigate through the various functionalities with ease. The implementation of the library management system utilizes data structures, file handling, and control flow constructs to ensure the efficient management of book and student information, demonstrating good programming practices, such as modularity, error handling, and input validation. This report presents a detailed analysis of the "Library-management.c" code, exploring the design choices, implementation details, and potential areas for improvement, with the findings serving as a valuable reference for developers working on similar library management projects or those interested in enhancing the existing system.

3. Objective

The primary objective of the library management system code is to provide a centralized and efficient solution for managing the various operations and records of a library. The key goals and objectives of this system can be summarized as follows:

+Book Inventory Management:

- Maintain a comprehensive database of the library's book collection, including details such as book title, type, author, and unique identification numbers.
- Allow librarians to easily search for and view information about the available books, either by book name or unique ID.
- Facilitate the process of adding new books to the library's collection, ensuring that the records are accurately updated.

+Borrowing Process Automation:

- Streamline the process of students borrowing books from the library.
- Capture and store relevant student information, such as name, borrowing date, and number of books borrowed.
- Enable the library to monitor and track the borrowing activities of students, improving resource utilization and service delivery.

+Data Persistence and Integrity:

- Utilize file-based storage to maintain the library's records, ensuring the persistence of data across multiple sessions.
- Maintain the accuracy and consistency of the library's book and borrowing records by properly managing the storage and retrieval of information.

+User-friendly Interface:

- Provide a simple and intuitive command-line-based interface for both librarians and students to interact with the system.
- Ensure a seamless user experience by incorporating features like clear screen management and efficient input handling.

+Scalability and Flexibility:

- Design the system with the ability to accommodate the growing needs of the library, such as the expansion of the book collection or an increase in the number of students.
- Leverage the underlying data structures and file management techniques to enable the system to be easily modified and expanded as the library's requirements evolve over time.

By addressing these key objectives, the library management system code aims to transform the way libraries manage their operations, ultimately enhancing the overall user experience for both librarians and students. The efficient handling of book inventory, borrowing processes, and data

management enables the library to operate more effectively, optimizing resource utilization and improving service delivery.

4. Solution

+Data Structures and File Management:

- The code utilizes two custom structures, book and student, to represent the core entities of the library management system.
- The book structure stores the book's name, type, author name, phone number, and a unique book ID.
- The student structure holds the student's name, the date of borrowing, the student's phone number, and the number of books borrowed.

+The system relies on three text files to persistently store and manage the library's data:

- "Add-book.txt" - Stores the details of all the books in the library.
- "All-Book.txt" - Maintains a comprehensive record of the entire book collection.
- "Borrow-book.txt" - Keeps track of the books borrowed by students.

By using file-based storage, the system ensures that the library's records are maintained across multiple sessions, enabling continuous access and updates.

+Book Management Functions:

- The viewBook() function provides the ability to search for books based on either the book's name or the book's unique ID.
- It presents the user with a menu to choose the search method and then reads the "Add-book.txt" file to find the matching book information.
- If a book is found, the function displays the relevant details to the user. If no match is found, it informs the user that the book is not available.
- The addBook() function allows the librarian to add new books to the library's collection.
- It prompts the librarian to enter the book's details, which are then appended to both the "Add-book.txt" and "All-Book.txt" files, ensuring the library's records are updated.

+Borrowing Management Functions:

- The borrowBook() function facilitates the process of students borrowing books from the library.
- It prompts the student to enter their name, the date of borrowing, and their phone number.
- The function then updates the "Borrow-book.txt" file with the student's information and the number of books borrowed.

+User Interface and Input Handling:

- The code provides a command-line-based interface, allowing both librarians and students to interact with the library management system.
- It utilizes functions like `fflush(stdin)` to clear the input buffer and `system("cls")` to clear the console screen, enhancing the user experience.
- The input handling mechanisms ensure that the system can gracefully handle various user inputs, minimizing the risk of errors or unexpected behavior.

+Scalability and Extensibility:

- The modular design of the code, with well-defined functions for each operation, enables the system to be easily extended and scaled to accommodate the growing needs of the library.
- The file-based data storage approach allows the system to handle an increasing number of books and student records without significant performance degradation.
- The underlying data structures and file management techniques can be further optimized or expanded as the library's requirements evolve over time.

Overall, the library management system code presents a comprehensive solution that addresses the key objectives of book inventory management, borrowing process automation, data persistence, and user-friendliness. By leveraging file-based storage and well-structured functions, the system provides a reliable and scalable platform for managing the operations of a library effectively.

5. Result

+Book Management:

- When the user chooses the "View Book" option, the system will display a menu prompting the user to search for a book either by name or by unique book ID.
- If the user searches by name, the system will read the "Add-book.txt" file, scan through the book records, and display the details of any matching books found.
- If the user searches by book ID, the system will again read the "Add-book.txt" file, locate the book with the specified ID, and present the corresponding book information.
- If no matching book is found, the system will inform the user that the book is not available in the library's collection.
- When the user chooses the "Add Book" option, the system will prompt the user to enter the new book's details, such as the book name, type, author name, and phone number.
- The system will then append the new book information to both the "Add-book.txt" and "All-Book.txt" files, effectively updating the library's book collection records.

+Borrowing Management:

- When the user selects the "Borrow Book" option, the system will prompt the student to enter their name, the borrowing date, and their phone number.

- The system will then update the "Borrow-book.txt" file with the student's information and the number of books borrowed.
- This will allow the library to maintain a record of the books currently on loan and the students responsible for them.

+User Interface and Input Handling:

- The system will provide a clear and intuitive command-line-based interface, guiding the user through the various options and functionalities.
- The interface will display well-formatted menus and prompts, making it easy for both librarians and students to navigate the system.
- The input handling mechanisms will ensure that the system can gracefully handle different types of user inputs, such as string, integer, or float values, without crashing or producing unexpected behavior.
- The system will utilize functions like `fflush(stdin)` and `system("cls")` to maintain a clean and organized console display, enhancing the overall user experience.

+Data Persistence and Integrity:

- The system will rely on the three text files ("Add-book.txt", "All-Book.txt", and "Borrow-book.txt") to persistently store the library's book and borrowing records.
- Any changes made to the book collection or borrowing records, such as adding new books or updating borrowing information, will be reflected in the corresponding text files.
- This file-based storage approach will ensure that the library's data is maintained across multiple sessions, providing continuous access to the up-to-date information.

+Scalability and Extensibility:

- The modular design of the code, with well-defined functions for each operation, will allow the system to be easily extended or modified to accommodate the growing needs of the library.
- The file-based data storage solution can handle an increasing number of books and student records without significant performance degradation, ensuring the system's scalability.
- The underlying data structures and file management techniques can be further optimized or expanded as the library's requirements evolve over time, enabling the system to adapt to changing needs.

Overall, the library management system code will provide a comprehensive and reliable solution for managing the library's book collection, borrowing processes, and data records. The user-friendly interface, efficient data handling, and scalable architecture will contribute to the effective operation and management of the library's resources.

6. Conclusion

+Difficulties

- Since some of us still don't know well about the project and advance coding in C program, it becomes a difficulty for to do that project and we must research about the advance programming like File handling, Data storing, Structure...etc.

- The difficulty of this library management project is the borrow book function.

Since we must remove the book that student borrows from the file and return it back when the student returns the book, we find it hard to solve that problem. To solve that problem, we find a new solution. It is to make a new file to only store all book name instead of using information of new add book.

+Experience

- Understanding the needs of the library, including what features are necessary.

- We understand more clearly about File Handling, Array, String, Structure, and we learn more about how to debug the code and find the solution.

- Writing the code to implement the system. This involves programming the core functionalities such as searching for books, managing member records, handling transactions, etc.

- Implementing database interactions, including Create, Read, Update. Ensuring data integrity and efficient querying.