Library Management Mini Project

Name: Sami Djellabi

Group: L2 G6

Project Overview

This project implements a basic library management system using a linked list in C.

The system supports the following functionalities:

1. Borrowing a book.

2. Returning a book.

3. Displaying the inventory.

4. Searching for a book.

The project is structured with clear functions for each feature and includes error handling for invalid operations.

Code Explanation

The program consists of the following key components:

- Data structure: The linked list node represents a book.

- Functions for borrowing, returning, displaying, and searching books.

- Main function to provide a menu-driven interface for users.

Each function is designed to perform a specific task while maintaining the integrity of the linked list.

Screenshots

note: the code is explained by comments in the source file

header files

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Data Structures

```c
// CODE EXPLANATION
// this struct represent a node in the linked list
// it has the book information and a pointer to the next book(node)
typedef struct Book {
    char title[100];
    char author[100];
    int copies;
    struct Book *next;
} Book;

// CODE EXPLANATION
// this is the head of the linked list
Book *head = NULL;
```

Helper Function

```c
// CODE EXPLANATION
// Create a new book node and allocate memory for it
Book* createBook(const char *title, const char *author, int copies) {
    // allocate enough memory to store the book data
    Book *newBook = (Book *)malloc(sizeof(Book));
    // copy the given title to the book->title
    strcpy(newBook->title, title);
    // copy the given author to the book->author
    strcpy(newBook->author, author);
    newBook->copies = copies;
    // set the next book to NULL
    newBook->next = NULL;
    return newBook;
}
```

main function

```c
// Main function
int main() {
    int choice;
    char title[100];

    // menu like program
    do {
        printf("\nLibrary Management:\n");
        printf("1. Borrow Book\n");
        printf("2. Return Book\n");
        printf("3. Display Inventory\n");
        printf("4. Search Book\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter title to borrow: ");
                scanf(" %[^\n]", title);
                BorrowBook(title);
                break;
            case 2:
                printf("Enter title to return: ");
                scanf(" %[^\n]", title);
                ReturnBook(title);
                break;
            case 3:
                DisplayInventory();
                break;
            case 4:
                printf("Enter title to search: ");
                scanf(" %[^\n]", title);
                SearchBook(title);
                break;
            case 5:
                printf("Exiting...\n");
                break;
            default:
                printf("Invalid choice.\n");
        }
    } while (choice != 5);

    return 0;
}
```

Function Outputs

```
caspeer@grey [proj] → ./assignment

Library Management:
1. Borrow Book
2. Return Book
3. Display Inventory
4. Search Book
5. Exit
Enter your choice:
```

BorrowBook function

```c
// CODE EXPLANATION
// we traverse the linked list and see if a book matches the given title
void BorrowBook(const char *title) {
    Book *current = head, *prev = NULL;
    while (current) {// traverse the linked list
// compare the current book title with the given title
        if (strcmp(current->title, title) == 0) {
            // check if we have enough copies
            if (current->copies > 0) {
                // decrement the copies number
                current->copies--;
                printf("Borrowed: %s\n", title);
                if (current->copies == 0) { // Remove if no copies left
                    if (prev) prev->next = current->next;
                    else head = current->next;
                    free(current);
                }
            } else {
                // if there is no copies of the book
                printf("All copies of '%s' are borrowed.\n", title);
            }
            return;
        }
        prev = current;
        current = current->next;
    }
    // we don't found the book
    printf("Book not found: %s\n", title);
}
```

Function Outputs

```
Library Management:
1. Borrow Book
2. Return Book
3. Display Inventory
4. Search Book
5. Exit
Enter your choice: 1
Enter title to borrow: Sami Djellabi Bio
Borrowed: Sami Djellabi Bio
```

ReturnBook function

```c
//CODE EXPLANATION
void ReturnBook(const char *title) {
    Book *current = head;
    // traverse the library linked list until we find the book
    while (current) {
// compare the current book title with the given title
        if (strcmp(current->title, title) == 0) {
            current->copies++; // increment the copies of the founded book
            printf("Returned: %s\n", title);
            return; // Done
        }
        current = current->next;
    }
    // Add new book if not found
    printf("Enter author for new book: ");
    char author[100];
    scanf(" %[^\n]", author);
    Book *newBook = createBook(title, author, 1);
    newBook->next = head;
    head = newBook;
}
```

Function Outputs

```
Library Management:
1. Borrow Book
2. Return Book
3. Display Inventory
4. Search Book
5. Exit
Enter your choice: 2
Enter title to return: Sami Djellabi Bio
Enter author for new book: Sami Dj

Library Management:
1. Borrow Book
2. Return Book
3. Display Inventory
4. Search Book
5. Exit
Enter your choice:
```

SearchBook function

```c
//CODE EXPLANATION
// Search a book
void SearchBook(const char *title) {
    Book *current = head;
    //we traverse the library linked list and compare
    //the book title with the given title (linear serach)
    while (current) // check if current ≠ NULL {
        // compare book->title and title
        if (strcmp(current->title, title) == 0) {
            // we found it
            printf("Found: Title: %s, Author: %s, Copies: %d\n",\
                   current->title, current->author, current->copies);
            return;
        }
    // next book
        current = current->next;
    }
// we don't found it
    printf("Book not found: %s\n", title);
}
```

Function Outputs

```
Library Management:
1. Borrow Book
2. Return Book
3. Display Inventory
4. Search Book
5. Exit
Enter your choice: 4
Enter title to search: Sami Djellabi Bio
Found: Title: Sami Djellabi Bio, Author: Sami Dj, Copies: 1
```

DisplayInventory

```c
//CODE EXPLANATION
// Display inventory
void DisplayInventory() {
    //if the head is NULL then the library is empty
    if (!head) {
        printf("Library is empty.\n");
        return;
    }
    // otherwise we traverse the library linked list and printf the information
    // of every node until we reach NULL
    Book *current = head;
    while (current) {
        printf("Title: %s, Author: %s, Copies: %d\n", current->title, current->author, current->copies);
        current = current->next;
    }
}
```

Function Outputs

```
Library Management:
1. Borrow Book
2. Return Book
3. Display Inventory
4. Search Book
5. Exit
Enter your choice: 3
Title: Sami Djellabi Bio, Author: Sami Dj, Copies: 1
```

the full code is available on github: https://github.com/caspee-r/algo-mini-project

THANKS