International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0205

e ISSN: 2584-2854
Volume: 02
Issue: 05 May 2024
Page No: 1515-1521

Linked List Data Structure and Library Management System

Nayana Joshi[1], Nisha Satpute[2], Komal Walgude[3], Dhanashri Korpad[4], Eeshwari Ransing[5]
[1,2,3,4,5]Assistant Professor, Vishwakarma College of Arts, Commerce and Science, Pune, Maharashtra, India.
Email Id: *nayanajoshi80@gmail.com[1], nisha.satpute3l@gmail.com[2], kswalgude16@gmail.com[3], dhanashrikorpad1991@gmail.com[4], ishika1912ransing@gmail.com[5]*

Abstract
Data Structure and Library Management System, focuses on the implementation of a linked list data structure for a library management, aiming to address the efficiency and functionality of managing library resources through the use of linked lists. The paper explores the advantages of using linked lists in the context of a library management system, discussing how this data structure can optimize operations such as adding or removing books from the library, tracking borrowing history, and managing book availability. By utilizing a linked list data structure, the library management system can efficiently organize and manipulate its resources, leading to improved search and retrieval times. The implementation of a linked list data structure in a library management system streamlines processes, enhances resource management, and ultimately improves the overall efficiency of the system.
Keywords: *Data Structure; Library Management; Linked List.*

1. Introduction

The library management system is a crucial component in organizing and accessing resources in a library. It is important to implement the appropriate data structure, such as a linked list, to efficiently manage and manipulate the library's resources [4]. By implementing a linked list data structure in the library management system, it becomes easier to add, delete, and search for books or materials. This data structure allows for efficient traversing and organizing of the library's resources, facilitating smooth operations and enhancing user experience. Additionally, the use of a linked list data structure ensures that the resources in the library are properly indexed and can be easily retrieved. Furthermore, the implementation of a linked list data structure in a library management system research paper allows for the exploration and evaluation of its effectiveness in terms of performance and scalability. The linked list data structure proves to be a suitable choice for a library management system research paper due to its efficient traversal and organization capabilities, as well as its potential to enhance the overall functionality of the system. Linked list, when compared, overcomes all the disadvantages of an array, as in the linked list the number of elements is not fixed nor the allocation of memory is needed, insertions and deletion of nodes is easy and quite simple.[1]

2. Role of Linked List

A linked list data structure can be a useful tool for building a library management system. Linked list basically classified into singly linked list, doubly linked list, circular linked list. In this research paper, a Singly linked list has been opted to know, how one can traverse, insert and delete the library related data in the form of nodes.[2] Here's how it can be applied in Figure 1 &2.

2.1. Storing Book Information

- Each node in the linked list can represent a book.
- The node can contain details like book ID (unique identifier), title, author, genre, availability (available/borrowed), etc.
- This allows for easy addition and removal of books from the system.

2.2. Efficient Insertion and Deletion

- Linked list insert and delete elements at any position.

e ISSN: 2584-2854
Volume: 02
Issue: 05 May 2024
Page No: 1515-1521

- This is beneficial when adding new books or removing borrowed ones.

2.3. Managing Borrowed Books (Separate Linked List)

- A separate linked list can track borrowed books.
- Each node can hold details like borrowed book ID, borrower ID, due date, etc.
- This simplifies searching for borrowed books and identifying overdue one

3. Diagram

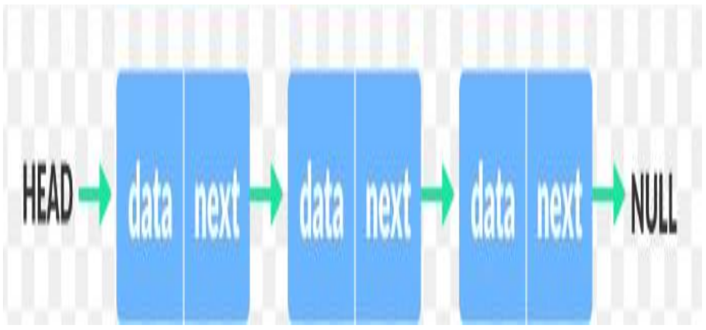


Figure 1 Representation of Node [3]

4. Implementation Through Coding

Linked List Program [4]

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>

struct book {
char name [30];
char author [30];
int id;
struct book *next;
};

struct student info {
char name [30];
char email [20];
char book [20];
char a [30];
int id;
struct student_info *next;
};

struct book *start_lib=NULL;
struct student_info *start=NULL;
struct book *initialize_lib (struct book *);
struct student_info *book_issue (struct student_info
*);
struct student_info *book_return (struct
student_info *);
struct book *diplay_lib(struct book *);
struct book *delete_book(int);
struct book *add_book (char [], char [], int);
void display (struct student_info *);
void greetings ();
void menu ();

int main () {
start_lib=initialize_lib(start_lib);
greetings ();
menu ();
return 0;
}

void greetings () {
printf("\n\n");
printf("\t\t\t
**************************************\
n");
printf("\t\t\t   *                              *\n");
printf("\t\t\t   *                              *\n");
printf("\t\t\t   *    ----------------------------    *\n");
printf("\t\t\t    *      WELCOME TO student_info
LIBRARY      *\n");
printf("\t\t\t   *    ----------------------------    *\n");
printf("\t\t\t   *                              *\n");
printf("\t\t\t   *                              *\n");
printf("\t\t\t
**************************************\
n");
printf("\n\n");
printf("\t\t\t
**************************************\
n");
printf("\t\t\t   *                              *\n");
printf("\t\t\t   *     ------------------------     *\n");
printf("\t\t\t     *           Student_info LIBRARY
*\n");
printf("\t\t\t   *     ------------------------     *\n");
```

e ISSN: 2584-2854
Volume: 02
Issue: 05 May 2024
Page No: 1515-1521

```
printf("\t\t\t     *                              *\n");
printf("\t\t\t     *                              *\n");
printf("\t\t\t    *     Pune,Maharashtra,India     *\n");
printf("\t\t\t
*******************************************\
n");
printf ("\n\n\t\t\t            Press to continue: ");
getch ();
}

void menu () {
int choice;
do {
printf("\n\n");
printf("\n\t\t\t*******************************
*******************\n");
printf("\n\t\t\t\t      MENU: ");
printf("\n\t\t\t\t     1.BOOK ISSUE ");
printf("\n\t\t\t\t     2.BOOK RETURN ");
printf("\n\t\t\t\t     3.DISPLAY ");
printf("\n\t\t\t\t     4.EXIT\n ");
printf("\n\t\t\t*******************************
*******************\n");
printf("\n\t\t\t\t      Enter your choice: ");
scanf("%d",&choice);
switch(choice) {
case 1: {
start=book_issue(start);
break;
}
case 2: {
start=book_return(start);
break;
}
case 3: {
display(start);
break;
}
case 4: {
exit (1);
}
default: {
printf("\n\t\t\t\t      ...Invalid Option!...\n");
printf("\n\t\t\t\t      Press any key to try again: ");
getch();
}
}
} while(choice!=4);
}

struct book *initialize_lib(struct book *start){
struct book *ptr, *n_book1, *n_book2, *n_book3,
*n_book4;

new_book1= (struct book *) malloc (sizeof (struct
book));
new_book1->next=NULL;
start_lib=new_book1;
strcpy(new_book1->name,"Let Us C");
strcpy(new_book1->author,"Yashavant Kanetkar");
new_book1->id=101;
ptr=new_book1;

new_book2= (struct book*) malloc (sizeof(struct
book));
new_book2->next=NULL;
strcpy(new_book2->name,"Object-Oriented
Programming with C++");
strcpy(new_book2->author,"E. Balagurusamy");
new_book2->id=102;
ptr->next=new_book2;
ptr=new_book2;

new_book3=(struct       book*)malloc(sizeof(struct
book));
new_book3->next=NULL;
strcpy(new_book3->name,"Java:   The   Complete
Reference");
strcpy(new_book3->author,"Herbert Schildt");
new_book3->id=103;
ptr->next=new_book3;
ptr=new_book3;

new_book4= (struct book*) malloc (sizeof(struct
book));
new_book4->next=NULL;
strcpy(new_book4->name,"A Byte of Python");
strcpy(new_book4->author,"Swaroop C H");
new_book4->id=104;
ptr->next=new_book4;
```

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2024.0205
e ISSN: 2584-2854
Volume: 02
Issue: 05 May 2024
Page No: 1515-1521

```
ptr=new_book4;

return start_lib;
}

struct student_info *book_issue(struct student_info
*start){
struct book *ptr;
struct student_info *ptr2, *new_student_info;
int i=1, id,flag=0;
if(start_lib==NULL) {
printf("\n\t\t\t\t No books in the library to issue!\n");
} else {
system("cls");
ptr=start_lib;
printf("\n\t*************** Books Available:
***************\n");
while(ptr! =NULL) {
printf("\n\t_____________________________________
__________________\n");
printf("\n\t Book %d",i);
printf("\n\t Book Title: %s",ptr->name);
printf("\n\t Name of Author: %s",ptr->author);
printf("\n\t Book ID: %d",ptr->id);
printf("\n\t_____________________________________
__________________\n");
ptr=ptr->next;
i++;
}
printf("\n\n\t Enter the Book ID: ");
scanf("%d",&id);
ptr=start_lib;
while(ptr!=NULL){
if(ptr->id==id) {
flag=1;
break;
}
ptr=ptr->next;
}
if(flag==1){
ptr=start_lib;
while(ptr->id! =id) {
ptr=ptr->next;
}
new_student_info=(struct student_info
*)malloc(sizeof(struct student_info));
printf("\n\t Enter student_info Details:\n ");
printf("\n\t Enter your Name: ");
scanf("%s",new_student_info->name);
printf("\n\t Enter your Email: ");
scanf("%s",new_student_info->email);
strcpy(new_student_info->book,ptr->name);
strcpy(new_student_info->a,ptr->author);
new_student_info->id=ptr->id;
new_student_info->next=NULL;
printf("\n\t Issue of Book ID %d done
successfully!\n",new_student_info->id);
printf("\n\n\t*********************************
*****************\n");
if(start==NULL) {
start=new_student_info;
} else {
ptr2=start;
while(ptr2->next! =NULL) {
ptr2=ptr2->next;
}
ptr2->next=new_student_info;
}
start_lib=delete_book(new_student_info->id);
printf("\n\n\t Press any key to go to the main menu:
");
getch();
system("cls");
} else {
printf("\n\t\t     ...Invalid Option!...\n");
printf("\n\t\t     Press any key to try again: ");
getch();
system("cls");
}
}
return start;
}

struct student_info *book_return(struct
student_info *start){
struct student_info *ptr, *preptr;
char bname[30],auname[30];
int flag=0, id,identity,c=0,d=1;
printf("\n\n\t*************** Books Submission:
***************\n");
```

```
printf("\n\n\t Enter your Book ID: ");
scanf("%d",&identity);
ptr=start;
while(ptr!=NULL){
if(ptr->id==identity) {
flag=1;
break;
}
ptr=ptr->next;
}
if(flag==1){
ptr=start;
while(ptr!=NULL){
c++;
ptr=ptr->next;
}
ptr=start;
while(ptr->id! =identity) {
d++;
ptr=ptr->next;
}
ptr=start;
if (d==1) {
printf("\n\t__________________________________
__________________\n");
printf("\n\t student Name: %s",start->name);
printf("\n\t student Email: %s",start->email);
printf("\n\t Name of Book Issued: %s",start->book);
printf("\n\t__________________________________
__________________\n");
printf("\n\n\t Return of Book ID %d done
successfully!\n",identity);
printf("\n\n\t********************************
*****************\n");
strcpy(bname,start->book);
strcpy(auname,start->a);
id=start->id;
start=start->next;
free(ptr);
add_book(bname,auname,id);
} else {
ptr=start;
while(ptr->id! =identity) {
preptr=ptr;
ptr=ptr->next;
}
printf("\n\t__________________________________
__________________\n");
printf("\n\t student_info Name: %s",ptr->name);
printf("\n\t student_info Email: %s",ptr->email);
printf("\n\t Name of Book Issued: %s",ptr->book);
printf("\n\t Book ID: %d",ptr->id);
printf("\n\t__________________________________
__________________\n");
strcpy(bname,ptr->book);
strcpy(auname,ptr->a);
id=ptr->id;
preptr->next=ptr->next;
free(ptr);
add_book(bname,auname,id);
}
printf("\n\t Thank you! \n\t Do visit again! ");
printf("\n\n\t Press any key to go to the main menu:
");
getch();
system("cls");
} else {
printf("\n\tSorry the book doesn't exist! Please
recheck the entered ID");
printf("\n\t\t\t\t    Press any key to try again: ");
getch();
system("cls");
}
return start;
}

void display (struct student_info *start) {
struct student_info *ptr;
ptr=start;
while (ptr! =NULL) {
printf("\n\t*************        Details        of
student_infos: *************\n");
printf("\n\t__________________________________
__________________\n");
printf("\n\t\t student_info Name: %s",ptr->name);
printf("\n\t\t student_info Email: %s",ptr->email);
printf("\n\t\t Name of Book Issued: %s",ptr->book);
printf("\n\t\t Book ID: %d",ptr->id);
printf("\n\t__________________________________
__________________\n");
```

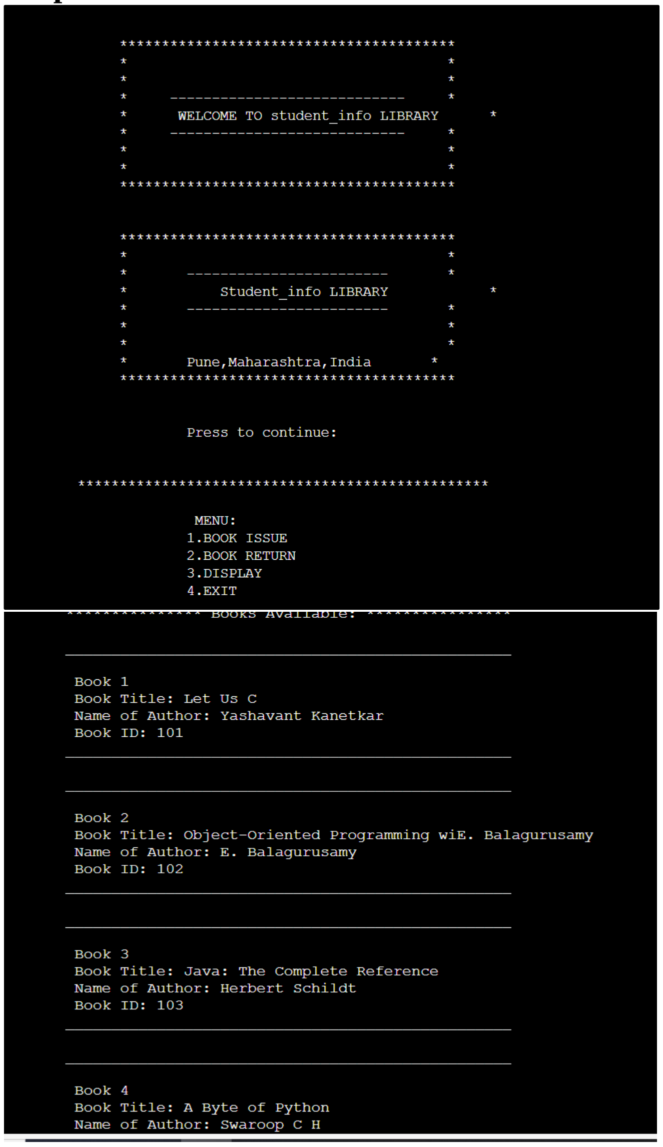
e ISSN: 2584-2854
Volume: 02
Issue: 05 May 2024
Page No: 1515-1521

```
printf("\n\n\t********************************
*****************\n");
ptr=ptr->next;
}
printf("\n\n\t Press any key to go to the main menu:
");
getch();
system("cls");
}

struct book *delete_book(int id){
struct book *ptr,*preptr;
int c=0;
ptr=start_lib;
while (ptr! =NULL) {
c++;
ptr=ptr->next;
}
if(c==1) {
ptr=start_lib;
start_lib=NULL;
free(ptr);
} else if(start_lib->id==id) {
ptr=start_lib;
start_lib=start_lib->next;
free(ptr);
} else {
ptr=start_lib;
while(ptr->id! =id) {
preptr=ptr;
ptr=ptr->next;
}
preptr->next=ptr->next;
free(ptr);
}
return start_lib;
}

struct book *add_book(char bname[30],char
auname[30],int id){
struct book *ptr, *new_book;
new_book= (struct book *) malloc (sizeof(struct
book));
strcpy(new_book->name,bname);
strcpy(new_book->author,auname);
new_book->id=id;
new_book->next=NULL;
if(start_lib==NULL) {
start_lib=new_book;
} else {
ptr=start_lib;
while(ptr->next! =NULL) {
ptr=ptr->next;
}
ptr->next=new_book;
}
return start_lib;
}
```

Output



```
        ****************************************
        *                                      *
        *                                      *
        *     ----------------------------     *
        *      WELCOME TO student_info LIBRARY      *
        *     ----------------------------     *
        *                                      *
        *                                      *
        ****************************************


        ****************************************
        *                                      *
        *       ------------------------       *
        *           Student_info LIBRARY            *
        *       ------------------------       *
        *                                      *
        *                                      *
        *       Pune,Maharashtra,India       *
        ****************************************


                Press to continue:


   **************************************************


                 MENU:
                1.BOOK ISSUE
                2.BOOK RETURN
                3.DISPLAY
                4.EXIT
    *************** Books Available: ****************

    ______________________________________________________


     Book 1
     Book Title: Let Us C
     Name of Author: Yashavant Kanetkar
     Book ID: 101
    ______________________________________________________


    ______________________________________________________


     Book 2
     Book Title: Object-Oriented Programming wiE. Balagurusamy
     Name of Author: E. Balagurusamy
     Book ID: 102
    ______________________________________________________


    ______________________________________________________


     Book 3
     Book Title: Java: The Complete Reference
     Name of Author: Herbert Schildt
     Book ID: 103
    ______________________________________________________


    ______________________________________________________


     Book 4
     Book Title: A Byte of Python
     Name of Author: Swaroop C H
```

Figure 2 Output of the Program

e ISSN: 2584-2854
Volume: 02
Issue: 05 May 2024
Page No: 1515-1521

Conclusion

Library management system is an application using linked list in the C programming language. One can perform library management operations like book issue, book return and display of records. The user issues the book by entering the book ID and the user details. Each user can issue only one book at a time. When the user returns the issued book, the book is available in the library for issuing again. The record of the issued book with user details can also be viewed. Overall, linked lists offer a dynamic approach to managing book information in a library system, especially for smaller libraries or for specific functionalities like tracking borrowed books.

References

[1]. ASharma, A.A. (2018). Review Paper on Dynamic Implementation Using Linked List Chandigarh university.3.
[2]. Garima Gupta, K. (. (2014). Dynamic Implementation Using Linked List. 5.
[3]. Programiz. (n.d.). Retrieved from https://www.programiz.com: https://www.programiz.com/dsa/linked-list
[4]. smita3199. (n.d.). https://github.com/smita3199/Library Management-System-using-Data-Structures/blob/main/Library_Management_System.c. From https://github.com/smita3199/Library-Management-System-using-Data-Structures: https://github.com