



Bus Booking System

Major Concept used :

Doubly Circular Linked List

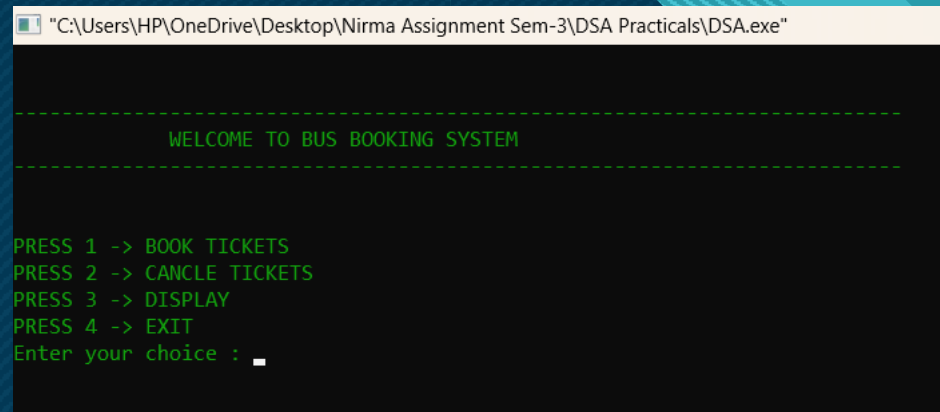
Code Explanation :

- In bus, every seat is represented as a node of Linked List
- Structure of Node :

```
typedef struct node
{
    struct node *next;
    struct node *prev;
    char name[50];
    long long int contact_no;
    int seat_no;
    char book;
    int pin;
} node;
```

- So that every node contains information of
 - Passenger name
 - Contact number
 - Seat number
 - char book : seat is booked or not
 - Unique pin number for every seat booking

Code Explanation :



```
"C:\Users\HP\OneDrive\Desktop\Nirma Assignment Sem-3\DSA Practicals\DSA.exe"

-----
WELCOME TO BUS BOOKING SYSTEM
-----

PRESS 1 -> BOOK TICKETS
PRESS 2 -> CANCEL TICKETS
PRESS 3 -> DISPLAY
PRESS 4 -> EXIT
Enter your choice : _
```

- In main function :
- First of all this window appears which asks user for
 - Book ticket
 - Cancel ticket
 - Display booked tickets arrangement
 - Exit from program

Book Ticket

```
<<---Seat Status--->>
-----
| 1 -> - | 2 -> - | 3 -> - | 4 -> - | 5 -> - |
| 6 -> - | 7 -> - | 8 -> - | 9 -> - | 10 -> - |
| 11 -> - | 12 -> - | 13 -> - | 14 -> - | 15 -> - |
| 16 -> - | 17 -> - | 18 -> - | 19 -> - | 20 -> - |
| 21 -> - | 22 -> - | 23 -> - | 24 -> - | 25 -> - |
| 26 -> - | 27 -> - | 28 -> - | 29 -> - | 30 -> - |
| 31 -> - | 32 -> - | 33 -> - | 34 -> - | 35 -> - |
| 36 -> - | 37 -> - | 38 -> - | 39 -> - | 40 -> - |
| 41 -> - | 42 -> - | 43 -> - | 44 -> - | 45 -> - |
| 46 -> - | 47 -> - | 48 -> - | 49 -> - | 50 -> - |
| 51 -> - | 52 -> - | 53 -> - | 54 -> - | 55 -> - |
-----
- = Available seats, * = Booked seats
-----

Enter No of Tickets You Want to Buy : 2

Enter your name : parth

Enter your mobile number : 8320685404

Enter seat number(s) :-
    Select Seat Number(1) = 1
    Select Seat Number(2) = 2
-----

THANK YOU..!

YOU HAVE SUCCESFULLY BOOKED THE SEATS

Seat No.(1) PIN : 11

Seat No.(2) PIN : 22

!!!!!!!!!!!!!!!!!!!!...KEEP PIN SAFELY...!!!!!!!!!!!!!!!!!!!!

PINS ARE REQUIRED AT THE TIME OF CANCELATION OF SEATS
-----
```

- When user enter 1 to book ticket...
- Program asks for **no of seats** passenger want to book
- And then take info of **name** & **contact** no of passenger
- Then asks passenger to enter the **seat numbers** user want to take
- And according to that it gives **unique PIN ID** to passenger (this unique PIN ID can be used at the time of cancel ticket)

How Book Ticket Works?

- `void book_seat()` method :
- In this method program asks passenger for
 - no. of seat he want to book - `n`
 - Name
 - Contact number
- Then for `n` times for loop runs in which...
 - Input seat numbers
 - And check if entered seat number is taken already by any other passenger then it shows "INVALID CHOICE" "SEAT NUMBER IS ALREADY TAKEN"
- And then asks if passenger want to
 - Check seat status
 - Book other seat exit booking portal

```
PRESS 1 to check Seat Status
PRESS 2 to Book other seat
PRESS 3 to Exit Booking Portal
Enter Your choice : 3_
```

How Book Ticket Works?

- And if we enter 1 to **check seat status** then it show booked and un-booked seat
- Here we can see that seat 1 & 2 are booked
- Then after entering 1 we can again go to the MAIN MENU
- Or by entering 2 WE CAN EXIT FROM BOOKING SYSTEM

```
PRESS 1 to check Seat Status
PRESS 2 to Book other seat
PRESS 3 to Exit Booking Portal
Enter Your choice : 1
```

```
<---Seat Status--->
```

1 -> *	2 -> *	3 -> -	4 -> -	5 -> -
6 -> -	7 -> -	8 -> -	9 -> -	10 -> -
11 -> -	12 -> -	13 -> -	14 -> -	15 -> -
16 -> -	17 -> -	18 -> -	19 -> -	20 -> -
21 -> -	22 -> -	23 -> -	24 -> -	25 -> -
26 -> -	27 -> -	28 -> -	29 -> -	30 -> -
31 -> -	32 -> -	33 -> -	34 -> -	35 -> -
36 -> -	37 -> -	38 -> -	39 -> -	40 -> -
41 -> -	42 -> -	43 -> -	44 -> -	45 -> -
46 -> -	47 -> -	48 -> -	49 -> -	50 -> -
51 -> -	52 -> -	53 -> -	54 -> -	55 -> -

```
- = Available seats, * = Booked seats
```

```
PRESS 1 to Main Menu
PRESS 2 To EXIT BOOKING SYSTEM
Enter your choice
```

Cancel Ticket

- `void cancel()` method :
- How it works?
 - First of all it shows the seat arrangement in which passenger gets the info about booked & un-booked seats
 - Then it takes passenger's name , contact no & PIN ID of seat & seat number
 - If `seat number` and `PIN ID matches` then it asks for confirmation and then `cancels` the seat
 - If `seat number` and `PIN ID does not match` then it shows `"Invalid Seat number & PIN combination..!"`
 - And then program asks passenger to...
 - PRESS 1 to check seat status
 - PRESS 2 to cancel more tickets
 - PRESS 3 to exit cancelation portal

```
PRESS 1 -> BOOK TICKETS
PRESS 2 -> CANCEL TICKETS
PRESS 3 -> DISPLAY
PRESS 4 -> EXIT
Enter your choice : 2
```

<<---Seat Status--->>

1 -> *	2 -> *	3 -> -	4 -> -	5 -> -
6 -> -	7 -> -	8 -> -	9 -> -	10 -> -
11 -> -	12 -> -	13 -> -	14 -> -	15 -> -
16 -> -	17 -> -	18 -> -	19 -> -	20 -> -
21 -> -	22 -> -	23 -> -	24 -> -	25 -> -
26 -> -	27 -> -	28 -> -	29 -> -	30 -> -
31 -> -	32 -> -	33 -> -	34 -> -	35 -> -
36 -> -	37 -> -	38 -> -	39 -> -	40 -> -
41 -> -	42 -> -	43 -> -	44 -> -	45 -> -
46 -> -	47 -> -	48 -> -	49 -> -	50 -> -
51 -> -	52 -> -	53 -> -	54 -> -	55 -> -

- = Available seats, * = Booked seats

SEAT CANCELTATION

Enter your name : parth

Enter your contact number : 8320685404

Enter the SEAT NUMBER you want to delete : 2

Enter PIN of the SEAT NUMBER : 22

Are you sure u want to cancel the seat (y/n) y

SEAT CANCELED SUCCESSFULLY..!

Cancel Ticket

```
SEAT CANCELED SUCCUESFULLY..!  
  
PRESS 1 to check seat status  
PRESS 2 to cancle more tickets  
PRESS 3 to exit cancelation portal  
Enter your choice : 2  
  
Enter your name : parth  
  
Enter your contact number : 8320685404  
  
Enter the SEAT NUMBER you want to delete : 1  
Enter PIN of the SEAT NUMBER : 123  
  
Invalid Seat number & PIN combination..!  
  
PRESS 1 to check seat status
```

- Here we can see that when we enter **wrong PIN ID** it does not allow us to cancel the **ticket**

Cancel Ticket

```
PRESS 1 to check seat status
PRESS 2 to cancle more tickets
PRESS 3 to exit cancelation portal
Enter your choice : 1

<<---Seat Status--->>

| 1 -> * | | 2 -> - | | 3 -> - | | 4 -> - | | 5 -> - |
| 6 -> - | | 7 -> - | | 8 -> - | | 9 -> - | | 10 -> - |
| 11 -> - | | 12 -> - | | 13 -> - | | 14 -> - | | 15 -> - |
| 16 -> - | | 17 -> - | | 18 -> - | | 19 -> - | | 20 -> - |
| 21 -> - | | 22 -> - | | 23 -> - | | 24 -> - | | 25 -> - |
| 26 -> - | | 27 -> - | | 28 -> - | | 29 -> - | | 30 -> - |
| 31 -> - | | 32 -> - | | 33 -> - | | 34 -> - | | 35 -> - |
| 36 -> - | | 37 -> - | | 38 -> - | | 39 -> - | | 40 -> - |
| 41 -> - | | 42 -> - | | 43 -> - | | 44 -> - | | 45 -> - |
| 46 -> - | | 47 -> - | | 48 -> - | | 49 -> - | | 50 -> - |
| 51 -> - | | 52 -> - | | 53 -> - | | 54 -> - | | 55 -> - |

- = Available seats, * = Booked seats
```

- And then we enter 1 to check seat status we can see that...
 - Seat number 2 is deleted : we entered correct PIN ID
 - Seat number 1 is not deleted : we entered wrong PIN ID

Display Seat Arrangement

- `void display()` method :
- It displays the seat arrangement in this manner

```
PRESS 1 -> BOOK TICKETS
PRESS 2 -> CANCEL TICKETS
PRESS 3 -> DISPLAY
PRESS 4 -> EXIT
Enter your choice : 3
```

<---Seat Status--->

1 -> *	2 -> -	3 -> -	4 -> -	5 -> -
6 -> -	7 -> -	8 -> -	9 -> -	10 -> -
11 -> -	12 -> -	13 -> -	14 -> -	15 -> -
16 -> -	17 -> -	18 -> -	19 -> -	20 -> -
21 -> -	22 -> -	23 -> -	24 -> -	25 -> -
26 -> -	27 -> -	28 -> -	29 -> -	30 -> -
31 -> -	32 -> -	33 -> -	34 -> -	35 -> -
36 -> -	37 -> -	38 -> -	39 -> -	40 -> -
41 -> -	42 -> -	43 -> -	44 -> -	45 -> -
46 -> -	47 -> -	48 -> -	49 -> -	50 -> -
51 -> -	52 -> -	53 -> -	54 -> -	55 -> -

- = Available seats, * = Booked seats

```
PRESS 1 to Main Menu
PRESS 2 To EXIT BOOKING SYSTEM
Enter your choice
2
```

EXIT SUCCESSFULLY

```
Process returned 0 (0x0)   execution time : 596.644 s
Press any key to continue.
```

THANKYOU!