

Lab 2

2347139

November 15, 2023

Implement linked list and its operations Consider each node as structure representation of data for your domain. Perform all operations and implement different types of linked list

```
1  #include <stdio.h>
2  // #include <conio.h>
3  #include <stdlib.h>
4  #include "ll.h"
5
6  struct ll *head = NULL;
7
8  int main()
9  {
10     if (head != NULL)
11     {
12         loop();
13     }
14     else
15     {
16         head = newll();
17         loop();
18     }
19 }
20 void loop()
21 {
22     int choice, pos;
23     char c = 'n';
24     while (1)
25     {
26         system("cls");
27         printf("\n The List:\n");
28         display(head);
29         printf("\n -----");
30         printf("\n \n 1. Insert In Beginning");
31         printf("\n 2. Insert at End");
32         printf("\n 3. Insert In Between");
```

```

33     printf("\n 4. Delete");
34     printf("\n \n Enter your choice:");
35     scanf("%d", &choice);
36     switch (choice)
37     {
38     case 1:
39     {
40         inbegin(head);
41         break;
42     }
43     case 2:
44     {
45         inend(head);
46         break;
47     }
48     case 3:
49     {
50         if (count(head) <= 1)
51         {
52             printf("\n There is only one element in the list and can
53                 't insert inbetween");
54             break;
55         }
56         printf("\n Enter the position:(2-%d)", count(head));
57         scanf("%d", &pos);
58         inbetween(head, pos);
59         break;
60     }
61     case 4:
62     {
63         del(head);
64         break;
65     }
66     default:
67     {
68         printf("Invalid Input");
69         break;
70     }
71     }
72     printf("\n\n-----\n
73         Do you want to continue?(y/n)");
74     scanf(" %c", &c);
75     if (c == 'y' || c == 'Y')
76     {
77         system("cls");
78     }
79     else
80     {
81         break;
82     }
83 }

```

```

83
84 struct ll *newll()
85 {
86     printf("\n The List is Empty!!!!!!");
87     printf("\n A new List is being created---");
88     struct ll *newnode = (struct ll *)malloc(sizeof(struct ll));
89     printf("\n Enter the HoneypotId:");
90     fflush(stdin);
91     scanf("%d", &newnode->data);
92     printf("\n Enter the Honeypot Name:");
93     fflush(stdin);
94     scanf("%[^\\n]*c", newnode->name);
95     newnode->link = NULL;
96     printf("\n New List created successfully");
97     return newnode;
98 }
99 void inbegin(struct ll *temp)
100 {
101     struct ll *newnode = (struct ll *)malloc(sizeof(struct ll));
102     printf("\n Enter the HoneypotId:");
103     fflush(stdin);
104     scanf("%d", &newnode->data);
105     printf("\n Enter the Honeypot Name:");
106     fflush(stdin);
107     scanf("%[^\\n]*c", newnode->name);
108     newnode->link = temp;
109     head = newnode;
110     printf("\n Insertion at beginning is Successfull");
111 }
112
113 void display(struct ll *ptr)
114 {
115     while (ptr != NULL)
116     {
117         printf("%d. %s ---> ", ptr->data, ptr->name);
118         ptr = ptr->link;
119     }
120     printf("NULL\\n");
121 }
122
123 int count(struct ll *temp)
124 {
125     int count = 0;
126     while (temp != NULL)
127     {
128         count++;
129         temp = temp->link;
130     }
131     return count;
132 }
133
134 void inend(struct ll *temp)

```

```

135 {
136     struct ll *newnode = (struct ll *)malloc(sizeof(struct ll));
137     printf("\n Enter the HoneypotId:");
138     fflush(stdin);
139     scanf("%d", &newnode->data);
140     printf("\n Enter the Honeypot Name:");
141     fflush(stdin);
142     scanf("%[^\\n]*c", newnode->name);
143     newnode->link = NULL;
144     while (temp->link != NULL)
145     {
146         temp = temp->link;
147     }
148     temp->link = newnode;
149     printf("\n Insertion at End is Successfull");
150 }
151
152 void inbetween(struct ll *temp, int pos)
153 {
154     if (1 < pos <= (count(head)))
155     {
156         int i;
157         struct ll *newnode = (struct ll *)malloc(sizeof(struct ll));
158         printf("\n Enter the HoneypotId:");
159         fflush(stdin);
160         scanf("%d", &newnode->data);
161         printf("\n Enter the Honeypot Name:");
162         fflush(stdin);
163         scanf("%[^\\n]*c", newnode->name);
164         for (i = 2; i < pos - 1; i++)
165         {
166             temp = temp->link;
167         }
168         newnode->link = temp->link;
169         temp->link = newnode;
170         printf("\n Insertion inbetween Completed");
171     }
172     else
173     {
174         printf("\n Invalid Position");
175     }
176 }
177
178 void del(struct ll *temp)
179 {
180     printf("\n\n Enter the Emp-ID which you wish to delete:");
181     int id;
182     fflush(stdin);
183     scanf("%d", &id);
184     int pos = searchid(id);
185     printf("\n The position of the node is: %d", pos);
186     if (pos <= 0)

```

```

187     {
188         printf("\n Emp-ID doesn't exist to delete");
189     }
190     else if (pos == 1)
191     {
192         head = head->link;
193         printf("\n\n Successfully removed the first node");
194     }
195     else if (pos == count(head))
196     {
197         delend(head);
198         printf("\n Successfully removed the last node");
199     }
200     else
201     {
202         delbetween(head, pos);
203         printf("\n Successfully removed the node");
204     }
205 }
206
207 int searchid(int id)
208 {
209     int pos = 0;
210     struct ll *temp;
211     temp = head;
212     if (id == temp->data)
213     {
214         pos = pos + 1;
215         return pos;
216     }
217     else
218     {
219         pos++;
220         while (temp != NULL)
221         {
222             temp = temp->link;
223             pos++;
224             if (id == temp->data)
225             {
226                 return pos;
227             }
228         }
229     }
230 }
231
232 struct ll *delbegin(struct ll *temp)
233 {
234     struct ll *head = NULL;
235     head = temp->link;
236     return head;
237 }
238

```

```

239 void delend(struct ll *t1)
240 {
241     struct ll *t2;
242     t2 = t1->link;
243     while (t2->link != NULL)
244     {
245         t1 = t1->link;
246         t2 = t2->link;
247     }
248     t1->link = NULL;
249     free(t2);
250 }
251
252 void delbetween(struct ll *t1, int pos)
253 {
254     struct ll *t2;
255     t2 = t1->link;
256     int count = 1;
257     while (count < pos - 1)
258     {
259         t1 = t1->link;
260         t2 = t2->link;
261         count++;
262     }
263     t1->link = t2->link;
264     free(t2);
265 }

```

Output

```
lab2.c - 02lab - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT TERMINAL PORTS
C/C++ Compile Run - output + - X

The List:
67. ---> 90. ---> 544. ---> NULL

-----

1. Insert In Beginning
2. Insert at End
3. Insert In Between
4. Delete

Enter your choice:3

Enter the position:(2-3)1

Enter the HoneypotId:989

Enter the Honeypot Name:
Insertion inbetween Completed

-----

Do you want to continue?(y/n)y
sh: line 1: cls: command not found
sh: line 1: cls: command not found

The List:
67. ---> 989. ---> 90. ---> 544. ---> NULL

-----

1. Insert In Beginning
2. Insert at End
3. Insert In Between
4. Delete

Enter your choice:4

Enter the Emp-ID which wish to delete:[]
```

```
lab2.c - 02lab - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT TERMINAL PORTS

-----

1. Insert In Beginning
2. Insert at End
3. Insert In Between
4. Delete

Enter your choice:2

Enter the HoneypotId:544

Enter the Honeypot Name:
Insertion at End is Successfull

-----

Do you want to continue?(y/n)y
sh: line 1: cls: command not found
sh: line 1: cls: command not found

The List:
67. ---> 90. ---> 544. ---> NULL

-----

1. Insert In Beginning
2. Insert at End
3. Insert In Between
4. Delete

Enter your choice:3

Enter the position:(2-3)1

Enter the HoneypotId:989

Enter the Honeypot Name:
Insertion inbetween Completed
```

```
lab2.c - 02lab - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT TERMINAL PORTS
C/C++ Compile Run - output + - X

In ~w/8/d/lab/02lab/output > ./"lab2"

The List is Empty!!!!!!
A new List is being created---
Enter the HoneypotId:90

Enter the Honeypot Name:
sh: line 1: cls: command not found
New List created successfully
The List:
90. ---> NULL

-----

1. Insert In Beginning
2. Insert at End
3. Insert In Between
4. Delete

Enter your choice:1

Enter the HoneypotId:67

Enter the Honeypot Name:
Insertion at beginning is Successfull

-----

Do you want to continue?(y/n)y
sh: line 1: cls: command not found
sh: line 1: cls: command not found

The List:
67. ---> 90. ---> NULL

-----

1. Insert In Beginning
2. Insert at End
```