

Experiment 7

Linked Lists

(Duration: 120 mins)

Author: Metin SAĞIK

metinsagik@iyte.edu.tr

06.07.2020

Purpose: The aim of this experiment is to implement operations of Linked Lists.

Problem Statement

You are asked to write a code that stores and retrieves student names and student ids in a linked list. The nodes of the linked list will consist of an integer which stores the student id and a fixed length char array which stores the student name. You will create a menu with 5 options which test linked list operations in the main function (**Insert, Listing, Search, Remove and Exit**).

(30 points) In the main function, options should be selected at first. Then, the command prompt should be cleared by calling the “clearscreen” function given in the lab procedure. If the selected operation requires inputs such as name and/or id, they should be asked from the user. Finally, “wait” function, which is given in the lab procedure, should be called to expect pressing ‘Enter’ to continue in the main loop. The loop in the main should be terminated once the “Exit” option is selected.

Option details:

1) (15 point) Insert: “Insert” function should be created to add the name and id of a student to the link list within a node. In the main, after getting the inputs in the corresponding option, “Insert” function should be called with the required arguments. Name and id of the added student should be printed on the screen as demonstrated in the final page of this document.

2) (10 point) Listing: Student information which currently exists in the list should be printed using “Listing” function. If the list is empty, a notification of "The list is empty. Please create the Listing after adding student number ..." should be printed in the main function.

3) (15 point) Search: The user is asked to enter the id of the student to be searched. If the student id is in the list, the id and the name of the student should be printed on the screen. If it is not listed, the message "Student number has not been found in the list ..." should appear. These messages should be printed in the “Search” function.

4) (20 point) Remove: The user is asked to enter the id of the student to be deleted from the list. If the student number you want to delete is in the list, the student should be removed, and " Student %s with an id number of %d has been deleted from the list....\n" should be printed on the screen. If

not, then "Student number you want to delete is not listed ..." should be printed on the screen. You must do these operations in the Remove function.

You should display the list after each insertion and removal. If the list is empty, the "Remove" and "Listing" output should display a screen output as in Figure 2.

5) (5 point) Exit: Finally, when option 5 is entered, the process will end with the message of "Finish Programming" appearing on the screen.

(5 point) If an option that is not available in the menu is entered by the user, a message should be written on the screen, such as "FALSE CHOOSE...".

All process steps are given in Figure 1.

Lab Procedure

- Use the "Wait" and "Cleanscreen" functions in your coding as follows:

```
void Cleanscreen () {  
    // For Windows:  
    system("cls");  
    // For Linux -->"clear"  
}  
  
void Wait () {  
    char temp_data;  
    printf ("Press Enter to continue!!!\n");  
    temp_data = getchar ();  
    temp_data = getchar ();  
    Cleanscreen ();  
}
```

- You can use a structure similar to "typedef struct" given below to create a node. (You don't have to use "typedef". You might also use the "struct" without "typedef".)

```
typedef struct node {  
    int id;  
    char name[N]; //N should be defined as 32 at the beginning  
    struct node *next;  
} list;  
list *head = NULL, *end = NULL;
```

- Write four functions with the prototypes shown below.

```
void Insert (int id, char name[N]);  
void Listing ();  
void Remove (int key);  
void Search (int wanted);
```

- ❖ Alternatively, you can use the following function structures.

```
struct node {  
    int id;  
    char name[N];    // N should be defined as 32 at the beginning  
    struct node *next;  
};  
  
struct node *Insert (struct node *head, int value, char name[N]);  
void Listing (struct node *head);  
void Search (struct node *head, int value);  
struct node *Remove (struct node *head, int value);
```

- Submission address: eelab204@gmail.com and CMS



Example Program outputs

<pre>----- 1-Insert 2-Listing 3-Search 4-Remove 5-Exit ----- Please choose an option. --> 1</pre>	<pre>Enter the student number: 47 Enter the student name: Ahmet Student Ahmet and number 47 added to the list Press Enter to continue!!!</pre>	<pre>----- 1-Insert 2-Listing 3-Search 4-Remove 5-Exit ----- Please choose an option. --> 1</pre>	<pre>Enter the student number: 73 Enter the student name: Gizem Student Gizem and number 73 added to the list Press Enter to continue!!!</pre>
<pre>----- 1-Insert 2-Listing 3-Search 4-Remove 5-Exit ----- Please choose an option. --> 2</pre>	<pre>Names Numbers ----- Ahmet 47 Sena 84 Gamze 59 Alper 25 Gizem 73 ----- Press Enter to continue!!!</pre>	<pre>----- 1-Insert 2-Listing 3-Search 4-Remove 5-Exit ----- Please choose an option. --> 3</pre>	<pre>Enter wanted student number: 25 Student Alper with an id number of 25 has been found... Press Enter to continue!!!</pre>
<pre>----- 1-Insert 2-Listing 3-Search 4-Remove 5-Exit ----- Please choose an option. --> 3</pre>	<pre>Enter wanted student number: 54 Student number has not been found in the list... Press Enter to continue!!!</pre>	<pre>----- 1-Insert 2-Listing 3-Search 4-Remove 5-Exit ----- Please choose an option. --> 4</pre>	<pre>Student number: 59 Student number 59 has been deleted from the list... Press Enter to continue!!!</pre>
<pre>----- 1-Insert 2-Listing 3-Search 4-Remove 5-Exit ----- Please choose an option. --> 4</pre>	<pre>Student number: 97 Student number you want to delete is not listed ... Press Enter to continue!!!</pre>	<pre>----- 1-Insert 2-Listing 3-Search 4-Remove 5-Exit ----- Please choose an option. --> 2</pre>	<pre>Names Numbers ----- Ahmet 47 Sena 84 Alper 25 Gizem 73 ----- Press Enter to continue!!!</pre>
<pre>----- 1-Insert 2-Listing 3-Search 4-Remove 5-Exit ----- Please choose an option. --> 7</pre>	<pre>FALSE CHOOSE... Press Enter to continue!!!</pre>	<pre>----- 1-Insert 2-Listing 3-Search 4-Remove 5-Exit ----- Please choose an option. --> 5</pre>	<pre>Finish Programming! Process exited after 1243 seconds with return value 0 Press any key to continue . . .</pre>

Figure 1. An example screen output

<pre>----- 1-Insert 2-Listing 3-Search 4-Remove 5-Exit ----- Please choose an option. --> 2</pre>	<pre>The list is empty. Please create the Listing after adding eleman... Press Enter to continue!!!</pre>	<pre>----- 1-Insert 2-Listing 3-Search 4-Remove 5-Exit ----- Please choose an option. --> 4</pre>	<pre>The list is empty. Please Remove after you add eleman... Press Enter to continue!!!</pre>
--	--	--	---

Figure 2. Remove and Listing output if list is empty