EE 306: Introduction to Computing

Programming Lab 3

Course Instructor: Prof. Nina Telang

TAs: Aniket Deshmukh, Shruthi Krish

All lab assignments must be completed individually. You are not permitted to seek help or clarification from anyone other than the instructor or the TAs.

Due date: Saturday, 4/7, 11:59pm

Objective:

The purpose of this assignment is to write a program in LC-3 assembly language to manage a directory of student *id* numbers, stored as a double linked-list: the two linked lists are for the UT directory and the 306 directory.

Addresses of the first nodes of the linked-lists corresponding to 306 and UT will be stored at memory locations x3800 and x3801, respectively, before the program is launched. Figure 1 shows the structure of nodes in the double linked-lists used in this assignment.
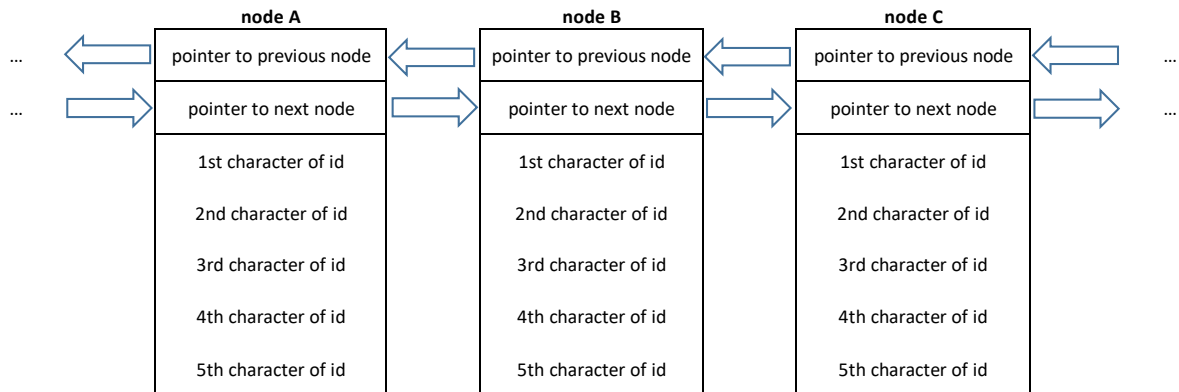

Figure 1. Structure of nodes in double linked-list used to store ids of class and school
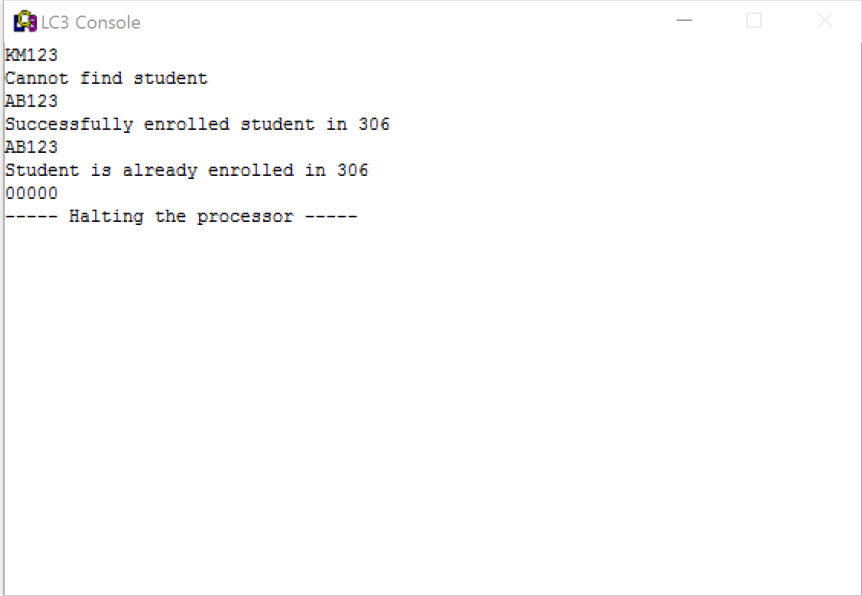
For example, if node A was stored at memory beginning at address x8000, address of node before node A would be at location x8000, address of node after node A would be at x8001 and id corresponding to this node would be stored at memory locations x8002-x8006.

Your program should read an id (exactly 5 characters) from the console. This id would specify the node that needs to be removed from school directory and added to class directory. This would

continue till user enters null id[1] ("00000"), at which point you would simply halt. Based on the state of the UT and 306 directories, we can imagine 3 scenarios.

1. Student was not registered for 306 before. Therefore, you would remove it from the UT directory and add it to the EE306 class directory. You also need to display the message "Successfully enrolled student in 306" on the console.
2. Student was registered before. In this case you need to inform the user by displaying message "Student is already enrolled in 306".
3. There is no student with given id in UT. If the id cannot be found in the school directory, you need to inform the user by displaying the message "Cannot find student".

Example: Tables 1 and 2 show state of the memory before and after execution of the program, respectively, for a console output of Figure 2.

```
LC3 Console                                    —    □    X
KM123
Cannot find student
AB123
Successfully enrolled student in 306
AB123
Student is already enrolled in 306
00000
----- Halting the processor -----
```

Figure 2. Sample console output

Table 1. State of memory before program is executed

Address	M[Address]
X3800	–
X3801	X4000
...	–

---

[1] It is assumed that none of the students in school directory have null id.

X4000	X0000
X4001	X4100
X4002	"A"
X4003	"B"
X4004	"1"
X4005	"2"
X4006	"3"
...	...
X4100	X4000
X4101	X0000
X4102	"C"
X4103	"D"
X4104	"4"
X4105	"5"
X4106	"6"

Table 2. State of memory after program is executed

Address	M[Address]
X3800	X4000
X3801	X4100
...	–
X4000	X0000
X4001	X0000
X4002	"A"
X4003	"B"
X4004	"1"
X4005	"2"
X4006	"3"
...	...
X4100	X0000
X4101	X0000
X4102	"C"
X4103	"D"
X4104	"4"
X4105	"5"
X4106	"6"

Hints:

1. Do not forget to modify the content of memory locations x3800 and x3801 while removing and adding nodes.
2. Do not forget to echo characters while reading from the console.
3. Pay attention to newlines in Figure 2.

4. You can reduce the size of your code by defining the common parts required for both directories as subroutines.

Notes:

1. *Id*s consist of any possible combination of ASCII codes. There may be non-character and non-number ASCII codes in ids. Characters in ids, if any, are case-sensitive.
2. Output of your program should exactly match the description and the example.
3. It is assumed that the null pointer (pointer that points to nowhere) would have value of 0. Therefore, the first field of the first node in linked-list and the second field of the last node in linked-list need to be zero.
4. You can test your program by manually constructing the double linked-list corresponding to the school directory in memory and putting the address of the first node in the memory location x3801.
5. The file that you will push onto the lab3 branch of github must be named lab3.asm.