EE306 Introduction to Computing

Programming Assignment # 3

The purpose of this assignment is to write a program in LC-3 assembly language to manage id directory, stored as double linked-list, of UT and 306.

Address of first nodes of 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 structure of nodes in double linked-lists used in this assignment.

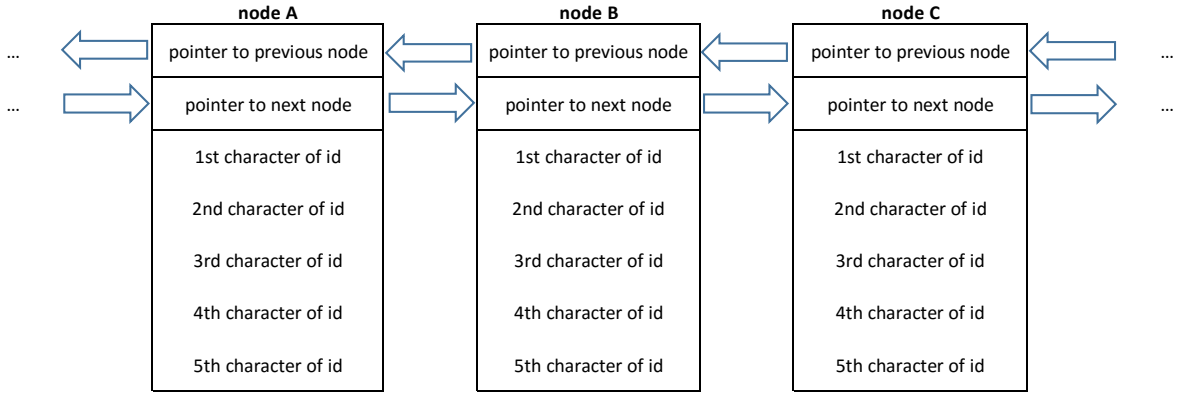

Figure 1. Structure of nodes in double linked-list used to store ids of class and school

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 lc3. Based on state of UT and 306 directories, we can imagine 3 scenarios.

1. Student was not registered for 306 before. Therefore, you would remove it from UT directory and add it to class directory. You also need to display message "Successfully enrolled student in 306" on the console.
2. Student was registered before. In this case you need to inform 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 user by printing message "Cannot find student".

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

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.

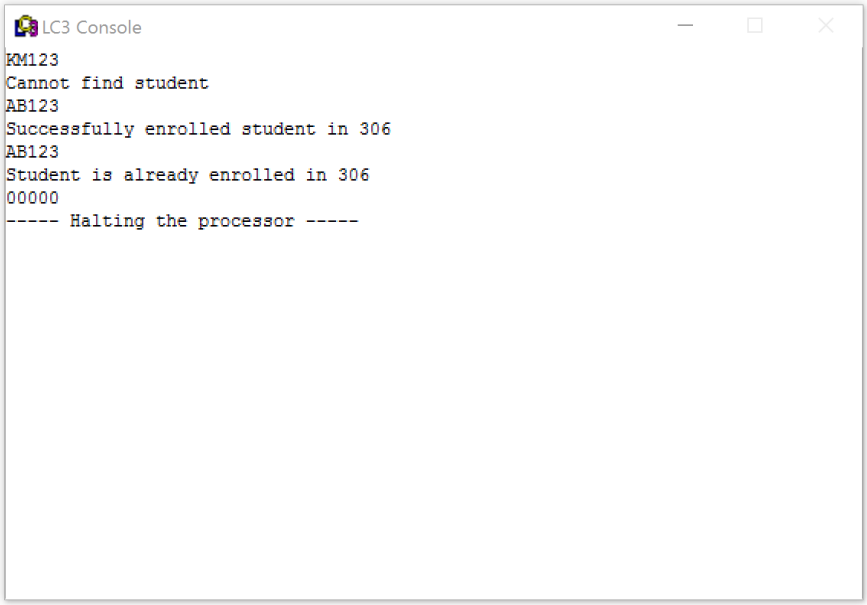

Figure 2. Sample console output

Table 1. State of memory before program is executed

Address	M[Address]
X3800	X0000
X3801	X4000
…	–
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 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 size of code by defining the common parts required for both directories as functions.

Notes:

1. Ids 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 null pointer (pointer that points to nowhere) would have value of 0. Therefore, first field of first node in linked-list and second field of last node in linked-list needs to be zero.
4. You can test your program by manually constructing double linked-list corresponding to school directory in memory and putting address of first node in the memory location x3801.
5. The file that you will upload to Canvas for this assignment must be named **linkedlist.asm**.