Lab E

Goals-

Using pointers to create a linked structure

You will create a simple linked structure. You will use a simple node that has a pointer to a Node. The data for the Node will be a single data member of type char. You will build a structure where the last node you add will point to the first node created, i.e. it will be a circular linked structure.

You will create a program that stores characters provided by the user, stored in the order they were entered. Your program will give the user the option to enter a value, display the current first item, display all of the current items (in order), and quit the program. Items can be read ONLY by removing them from the structure you create.

You will need functions to:

1. create the structure, initially empty
2. add to the back of the structure
3. display the value stored at the head of the structure (leaving the node in place)
4. remove the value stored in the head of the structure

Do you need anything else?

Your program will NOT provide a function to go through the structure and just look at each element. If you find a need for this while debugging that is fine for unit testing. You will remove it or comment it out in the final version.

You will two additional pointers:

1. head will point to the front of the structure (i.e. the first item entered)
2. back will point to the end, where you add new items

You will NOT delete any nodes. When you add the first item all three pointers (i.e. head, back, and next) will point to that node. When you add an item head will point to that node, back will point to the original node, and that node's next pointer will remain unchanged. When you have removed more values than you inserted you will have nodes with no values. The head and back pointers will just go around your ring of nodes.

To display the contents of the entire structure you will create a loop that runs until all the values have been removed and displayed. You will alternate calls to display and remove. You will not display the contents without removing each item.

What's the point of this? You are implementing a simple circular queue. It minimizes memory operations. If you have a situation with MANY data items being stored and removed constantly the overhead of the (much) slower memory can impact the system performance. This structure only has memory operations when an item is added to a full structure.

NOTE: If it's easier for you to do this project you can use doubly-linked lists, but the programming is a little more complicated.

What to submit?

Your source code file(s)
Your makefile
Submit all files in a zip file

Grading

Programming style- 1 point

Create the linked structure- 2 points

Create the add back function- 2 points

Create the remove front function- 2 points

Create the display front function- 1 point

Correctly use the front pointer- 1 point

Correctly use the back pointer- 1 point