CSE 6010 - Computational Problem Solving
Assignment 2 - Software Modules

Ting Liao / GTID: 903278773

Slide 1

I structured my program by following these steps to translate the idea of Linked List Queue into code.

First, I put “#include <stdio.h>” and “#include <math.h>” at the top of my code, which can let me use the function of “printf”, “scanf” and the math function. Then, I set the program environment as “int main(int argc, const char * argv[])” to allow me to execute the program.

Second, I had to define the data structure of the LLQueue and the QNode, which the first one is for the Linked List structure and the second one is for the node. In the “LLQ_create” function, I first initialized the head and the tail of the LLQ in order to prevent any segmentation fault related to the memory problem. Then, use an if-statement to check if the LLQ is successfully created. Also, in the “LLQ_insert” function, I used two if-statement to help me verify the status of the queue. To be more specific, I can know that the queue now is empty, contains some items or fails to insert items. Besides, I used a if-statement in the “LLQ_delete” function to verify whether the queue is empty or not. Moreover, in the “LLQ_Search” function, I used a while-loop to look through each node and then used an if-statement to check whether the item we want to search is inside the queue or not. What’s more, the “LLQ_minimum” and the “LLQ_maximum” function have similar structures. They both contains an if-statement to check whether the queue is empty or not and a while-loop to look through the queue and get the min (or max) item.

After these basic functions are set up, I made the “LLQ_count” function which contains a while loop to count the number of the items inside the queue. The “LLQ_print” function has one if-statement to check whether the queue is empty or not and a while-loop to print all the items inside the queue. Finally, the “LLQ_free” function will loop through each node and store into a temporary node (temp). Then, it will free the temp node every time to release the memory of each node. At last, it will free the LLQ and release all the memory.

Slide 2

Unit test:

```
Last login: Wed Sep  2 20:19:48 on ttys001
/Users/andyliao/Desktop/GT\ CEE/2020\ Fall\ courses/CSE\ 6010/HW2/tliao32/main ; e
xit;
(base) liudingde-MacBook-Pro:~ andyliao$ /Users/andyliao/Desktop/GT\ CEE/2020\ Fal
l\ courses/CSE\ 6010/HW2/tliao32/main ; exit;
Welcome to our Linked List Queue Menu!!
======================
 1 - LLQ_Insert
 2 - LLQ_Delete
 3 - LLQ_Search
 4 - LLQ_Minimum
 5 - LLQ_Maximum
 6 - LLQ_Count
 7 - LLQ_Print
 8 - LLQ_free
 9 - Exit
======================
 Select your Choice : 7
3.500 10.600 22.300 31.000
 Select your Choice : 1

Input: 431.99

Successfully insert the data, return 0.
 Select your Choice : 7
3.500 10.600 22.300 31.000 431.990
 Select your Choice : 5

The maximum item is 431.990.
 Select your Choice : 4

The minimum item is 3.500.
 Select your Choice : 3

Input the data you want to search : 3.5

Here's the pointer which you try to search, 0x7fe0aed02690
 Select your Choice : 3

Input the data you want to search : 22.3

Here's the pointer which you try to search, 0x7fe0aed026b0
 Select your Choice : 3

Input the data you want to search : 431.99

Here's the pointer which you try to search, 0x7fe0aed026d0
 Select your Choice : 3

Input the data you want to search : 100

Here's the pointer which you try to search, 0x0
 Select your Choice : 2
```

First, I made an user interface to let my program be more user friendly. Besides, the user can type in a number related to the LLQ menu. Also, if anyone type in a wrong number, it will handle it by showing 404 not found. Beginning with the unit test, from the first print (choice number 7), we can see that I have already create an empty queue and then inserted four items (3.5, 10.6, 22.3, 31.0) into the empty queue. Jump to the most important test, I spent quite a while on testing my search function because it only showed the number I just input or a segmentation fault. Then, I found that the prior print format was wrong, so I fixed it by modifying “printf(“%lf”)” to “printf(“%p”)”, that the %p can let me print the pointer. After that, when I input an item that doesn’t exist in the queue, it will return NULL, and the pointer is 0x0.

```
Successfully delete. The delete item is 3.500.
 Select your Choice : 7
10.600 22.300 31.000 431.990
 Select your Choice : 2

Successfully delete. The delete item is 10.600.
 Select your Choice : 7
22.300 31.000 431.990
 Select your Choice : 2

Successfully delete. The delete item is 22.300.
 Select your Choice : 7
31.000 431.990
 Select your Choice : 2

Successfully delete. The delete item is 31.000.
 Select your Choice : 7
431.990
 Select your Choice : 2

Successfully delete. The delete item is 431.990.
 Select your Choice : 7

This queue now is empty!!!

 Select your Choice : 2

Return 0.000000. The queue is empty, nothing can be deleted.
 Select your Choice : 6

There are 0 items currently in the queue.
 Select your Choice : 8

 Select your Choice : 7

This queue now is empty!!!

 Select your Choice : 9

Do you sure you want to quit the program? (1 (yes) / 2 (no)) : 1
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
```

I can use the LLQ_delete function to delete every node or simply use LLQ_free to free everything in this queue. After deleting everything, now when I select LLQ_print, it will show the queue now is empty. Finally, I have set a function “Exit” to let the user leave the program.

Slide 3

In this program, the most challenging part is handling the segmentation fault. Due to that I'm not familiar with handling any stuff related to the memory storing or address, I don't really understand why it shows this error at the very beginning. However, Thanks to Darby and Dr. Cherry help me figure out the problem. It is because I didn't initialize the new node which I created in some functions. After I modified those initializations, my program works pretty well. Even though it cost me about two days trying to do the debugging by myself, I truly learned a lot via searching on the internet.

All in all, it's a very useful practice for making a Linked list queue data structure program. Let me gain more deep understanding of the C programming language.