C291 – System Programming in C and UNIX

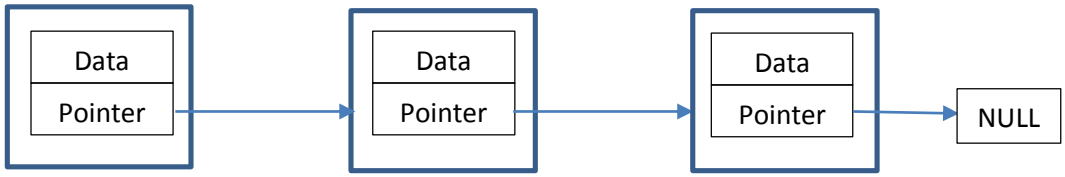
Assignment 5

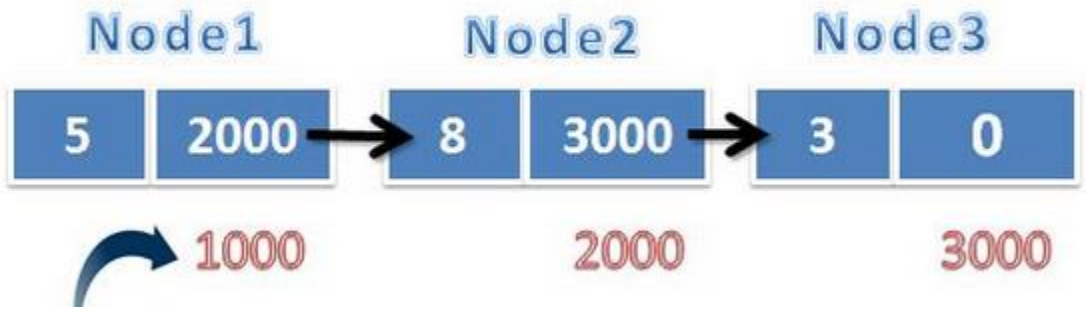
Task:

Write a C program to create and manipulate singly linked list.

What is a linked list?

Linked lists are a way to store data with structures so that the programmer can automatically create a new place to store data whenever necessary. One way to visualize a linked list is as though it were a train. The programmer always stores the first node of the list in a pointer he won't lose access to. This would be the engine of the train. The pointer itself is the connector between cars of the train. Every time the train adds a car, it uses the connectors to add a new car. This is like a programmer using malloc to create a pointer to a new struct.





Each element is a node in linked list. Elements is going to be a struct. Elements has two data members, data and pointer. Data will hold a value, in this assignment integer value and pointer will hold value to address of next element. The last element in linked list will have NULL pointer.

```
struct node {

 int x;

struct node *next;

};
```

How do you make a pointer point to next element?

```
struct node {
  int x;
  struct node *next;
};


int main()
{
   /* This will be the unchanging first node */
   struct node *root;

   /* Now root points to a node struct */
   root = (struct node *) malloc( sizeof(struct node) );

   /* The node root points to its next elements. Since we don't have any next element set it to
null */
   root->next = 0;
   /* By using the -> operator, you can modify what the node,
     a pointer, (root in this case) points to. */
   root->x = 5;
}
```

Create a linked list program using which user can add elements to linked list and search a number in the linked list.

Sample Output:

Enter your option:

1. Add element
2. Search
3. Display
4. Exit

1

Enter the value: 30

Value added to the linked list

Enter your option:

1. Add element
2. Search
3. Display
4. Exit

1

Enter the value: 40

Value added to the linked list

Enter your option:

1. Add element
2. Search
3. Display
4. Exit

3

30->40

Enter your option:

1. Add element
2. Search
3. Display
4. Exit

2

Enter value to search

30

Found value in linkedlist

Enter your option:

1. Add element
2. Search
3. Display
4. Exit

3

Enter value to search

100

Value not found

Enter your option:

1. Add element
2. Search
3. Display
4. Exit

4

Exiting…

NOTE: You are required to code only in open terminal. You should not use any IDE.

Bonus Points:

If you have the struct definition in node.h, linkedlist operations in operations.c and client.c to interact with client you will get bonus points. Maximum bonus points 30 pts.

Due Date:

The submission is due on Tuesday 10/13/2015 11:59 PM

What to turn in:

Upload .c and .h file in canvas. We'll grade whatever version you've put there at 11:59PM on the due date.

Academic Integrity:

You may discuss the assignment with other people at a high level, e.g. discussing general strategies to solve the problem. You may also consult printed and/or online references, including books, tutorials, etc., but you must cite these materials in report. However, if you are submitting the code, then it must be your own work, which you personally designed and wrote. You may not share written code with any other students, nor may you possess code written by another student either in whole or in part, regardless of format. The professor and AI's are always available to help, so reach out through canvas if you need one!

Rubric:

- You will get 50% of total marks on successful compilation of program without any errors
- You will get 70% of total marks on successful execution of program
- You will get 90% of total marks on passing all test cases
- You will get 100% of total marks based on your code clarity
- The following falls under clean code
    - Proper names for variables.
    - Follow camel case patterns.
    - Comment where ever needed
    - Check for valid inputs & valid range
    - Output unambiguous & user friendly messages