

Self Referencing Data Structures – Introduction to Linked List

By Anand George

Introduction

- Structures which contains pointer to same structure type.
- Self referencing structures are mainly used to implement data structures like linked list, stack, queue, tree, graph etc.
- Note:
 1. While we going to see a couple of them now and next couple of presentations please note these sessions are no way an introduction to algorithms or data structures but just a foundation for both on a C language perspective.
 2. None of the presentations on self referencing data structures **may** not implement the best algorithm for the sample operations it does.
 3. From a language standpoint this is all you going to need for any data structures or algorithms to be implemented or understood in C. So the focus is on the language part.

Self referencing structure

```
struct MyNode  
{  
    int data;  
    MyNode* link;  
};
```

Linked List

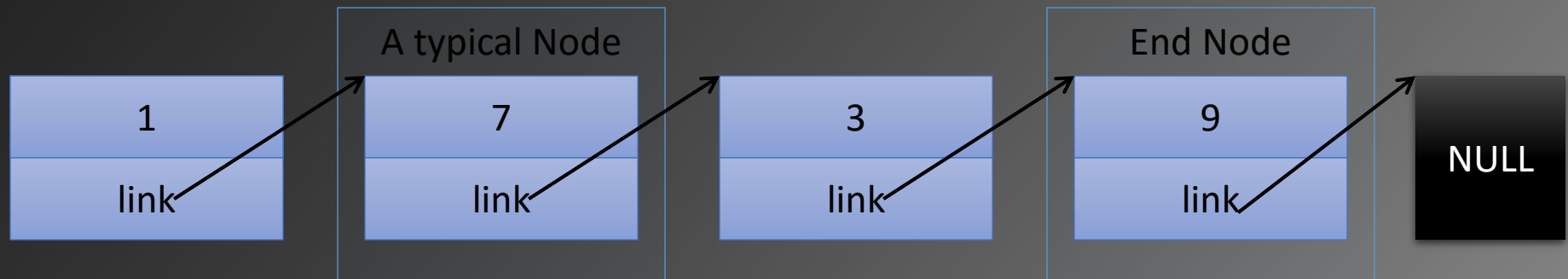
- Like array can be used to save multiple items of same type.
- Unlike array is not contiguous.
- Discontiguous memory chunks linked together.
- A chain of self referencing structure variables.
- Total memory can be dynamically allocated.

Node

- One link in the chain.
- Is a structure in C
- Will have pointer to its own type.
- Can have any amount of data.
- End of the list is designated by a pointer to NULL.

```
struct MyNode
{
    int data;
    MyNode* link;
};
```

Linked list.



Demo

- Linked list.
- Add 10 elements to the list.
- print the list.

Difference between array and linked list

- Indexing (getting the nth element)
 - Is faster in array as we can calculate the address of nth element if the first address is given.
 - Slow in linked list as traversal is need to nth element as we don't know the address of nth element only (n-1)th element knows it.
- Sorting and similar operations which need data to be moved between elements.
 - Normally operation which need change in elements position linked list is better as links can be exchanged instead of data itself. Sorting is just an example.
- Serialization(save to devices like disk/NIC etc)
 - Linked list cannot serialize by itself like array as memory is not contiguous.
- Expansion
 - Array cannot expand.
- Adding something in the middle or anywhere.
 - Array cannot do this operation if it is full or very costly.

Summary

- Self referencing structures.
- Linked List

Thank you