Linked Lists Part 2

What we did:

- Concept Intro
- Insert at head
- Linked list traversal
- Insert at tail

What we'll do today:

- Inserting anywhere in LL
- Removing from LL
- In the middle
- With only one item in a list

Recap

A linked list is a chain of nodes

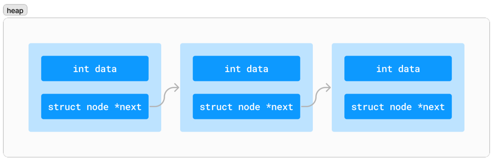
- A node is a struct, usually allocated on the heap
-

It contains a payload (some data), and a pointer to another node
Each node has some data and a pointer to the next node (of the same data type), creating a linked structure that forms the list

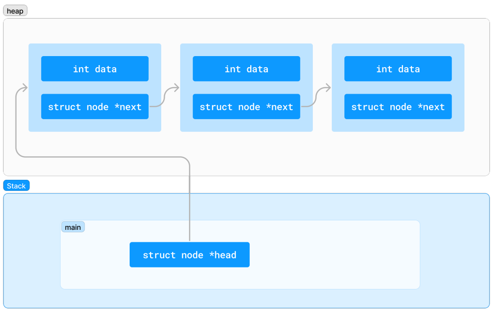
A node declaration in C

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

Visualisation of linked list

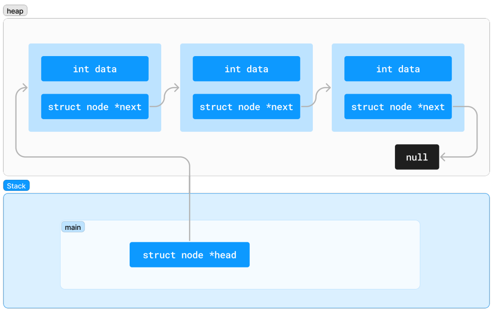


Need a reference to the linked list



Is a pointer to a struct node, but not a node itself.

How do we know we're at the end of the lined list?



To create a linked list, we:

- Define a struct for a node
- A pointer to keep track of where the start of the list
- A way to create a node and then connect it into our list

Demo?

Today's goals:

- insert_at_index
- delete_node_at_index
- remove_tail
- size_of_linked_list

Inserting in the middle of a linked list

1. Discuss
2. Whiteboard
3. Implement

Deleting in the middle of a linked list

1. Discuss
2. Whiteboard
3. Implement

Feedback

https://forms.office.com/r/Ze4admEWnR