Chapter - 10 : Pointers

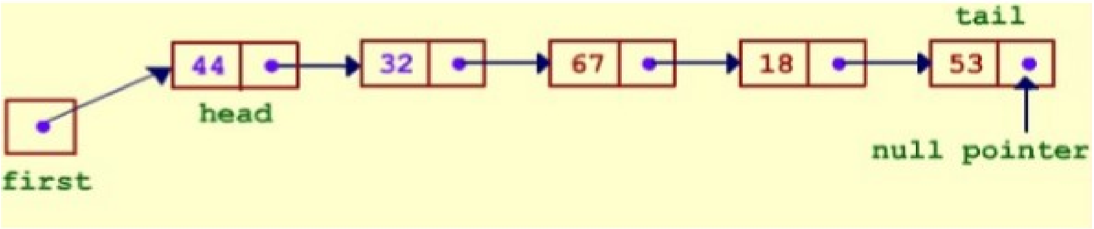
Insertion into a Linked List

Adding an Element to the Beginning of a List

- If the head pointer is NULL, the list is empty, and the new element will be its only member.
- If the head pointer is not NULL, the list already contains one or more elements.
- In either case, however, the procedure for adding a new element to the start of the list is the same:
- Create an instance of the structure, allocating memory space using malloc( ) or calloc( ).
- Set the next pointer of the new element to the current value of the head pointer.
- This will be NULL if the list is empty, or the address of the current first element otherwise.
- Make the head pointer point to the new element.

```
new =(struct node *) malloc( sizeof(struct node));
new->next = head;
head=new;
```

- Adding an Element to the Beginning of a List: Example

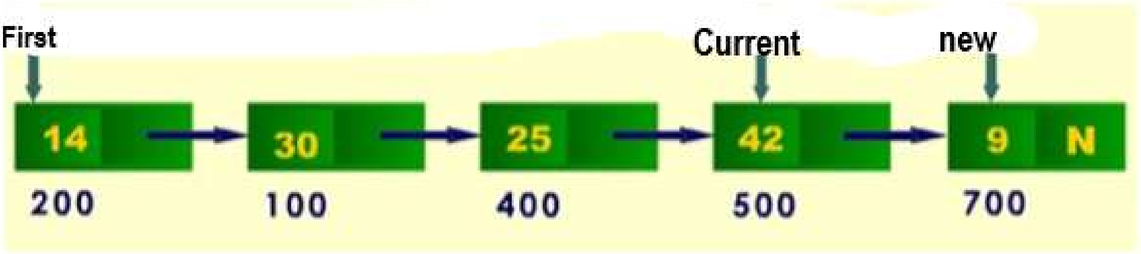


Adding an Element to the End of the List

- To add an element to the end of a linked list, one needs to start at the head pointer and go through the list until one finds the last element.
- Create an instance of the structure, allocating memory space using malloc( ).
- Set the next pointer in the last element to point to the new element (whose address is returned by malloc( )).
- Set the next pointer in the new element to NULL to signal that it is the last item in the list.

```
struct node *current;
current=head;
while( current -> next != NULL)
    current=current -> next;

new=(struct node*)malloc( sizeof( struct node));
current->next =new;
new->next=NULL;
```



Deletion from a Linked List

- Deleting an element from a linked list is a simple matter of manipulating pointers.
- The exact process depends on where in the list the element is located:
- To delete the first element, set the head pointer to point to the second element in the list.
- To delete the last element, set the next pointer of the next-to-last element to NULL.
- To delete any other element, set the next pointer of the element before the one being deleted to point to the element after the one being deleted.

The code to delete the first element in a linked list:

```
head = head->next;
```

The code to delete the last element and the element in the middle of the list

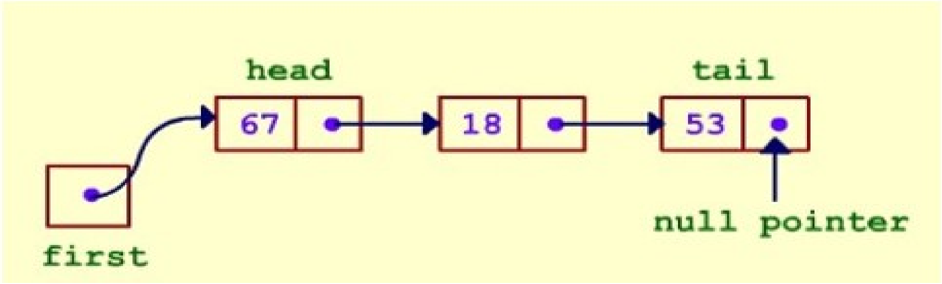
```
struct node *current1 , *current2;
current1=head;
current2=current1->next;

while( current2->next != NULL)
{        current1=current2;
         current2 = current1->next;
}
current1->next=NULL;
if (head==current1)
         head=NULL;
```

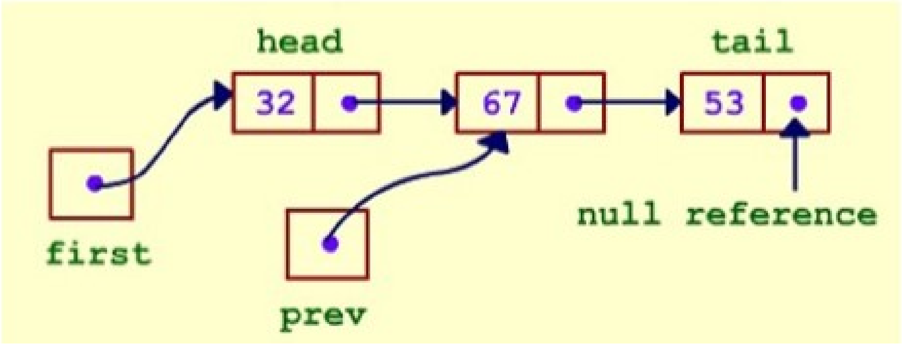
The code to delete the last element and the element in the middle of the list

```
struct node *current1 , *current2;
current2=current1->next;
current1->next= current2->next;
```

Deleting an element from the list



Deleting from the middle of a list


head
tail
32
67
53
first
prev
null reference