

CSE 220 – C Programming

Linked lists

Linked List

- A linked list is a chain of structures (nodes), with each node containing a pointer to the next node in the chain.
- The last node in the list contains a null pointer.



Declaring a Node Type

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

```
struct point {  
    double x, y;  
}
```

```
struct vertex {  
    struct point element;  
    struct vertex *next;  
}
```

Building Linked List

- First, create an empty list

```
struct node *first=NULL;
```

- Then create nodes one by one:
 - Allocate memory for the node

```
struct node *new_node;
```

```
new_node=malloc(sizeof (struct node));
```



- Store data into the node

- (*new_node).value = 10;
- new_node->value = 10;



- Insert the node into the list

Inserting a Node at the Beginning of the List

- If first points to the first node of the linked list:

```
new_node->next=first;
```

```
first=new_node;
```

```
struct node *first=NULL, *new_node;
```

first 

new_node 

```
new_node=malloc(sizeof (struct node));
```

first 

new_node 

Inserting a Node at the Beginning of the List

```
new_node->value=10;
```

first 

new_node 

```
new_node->next=first;
```

first 

new_node 

```
first=new_node;
```

first 

new_node 

Inserting a Node at the Beginning of the List

```
new_node=malloc(sizeof (struct node));
```



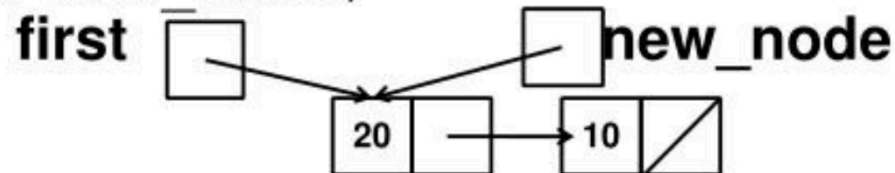
```
new_node->value=20;
```



```
new_node->next=first;
```



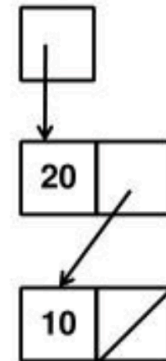
```
first=new_node;
```



Searching a Linked List

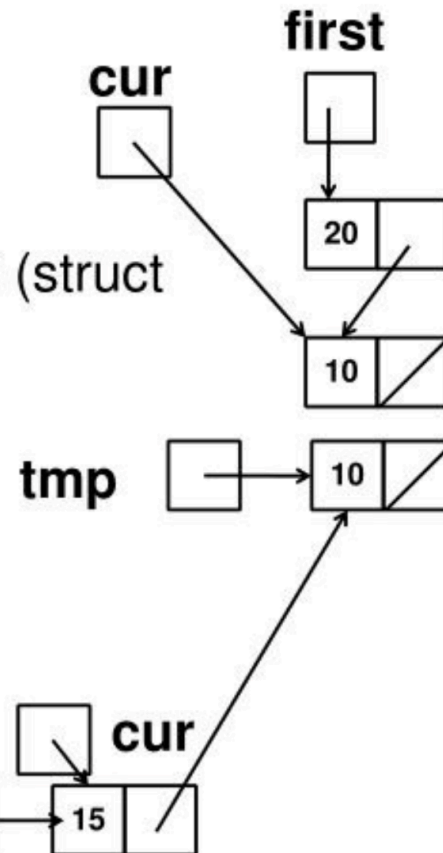
- for (p=first; p!=NULL; p=p->next)
{... }
- int value=20; struct node *p;
• for (p=first; p!=NULL; p=p->next)
 { if (p->value== value) return p; }
- struct node *find(struct node *list, int n){
 while (list!=NULL and list->value!=n)
 p=p->next;
 return list; }

first



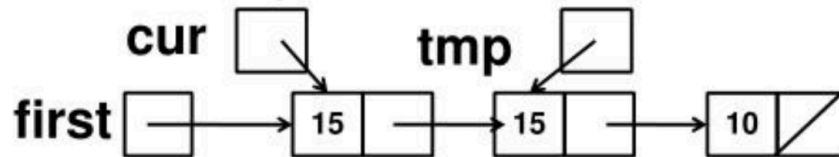
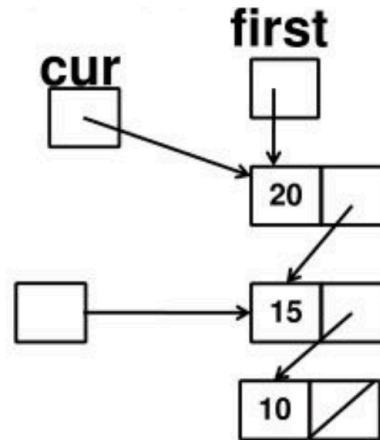
Insert a Node in the Middle of the List

- `struct *node p=first;`
- `struct node * cur= find(p, 10);`
- `struct node *tmp =malloc(sizeof (struct node));`
- `tmp->value= cur->value;`
- `tmp->next = cur->next_ptr;`
- `cur->value = 15;`
- `cur->next = tmp;`

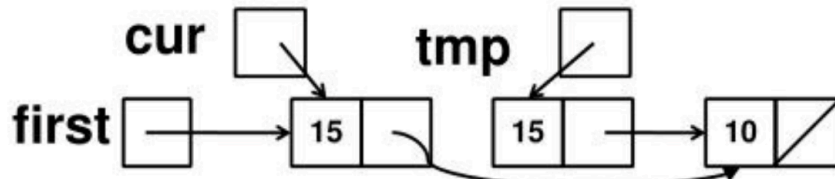


Deleting a Node from the List

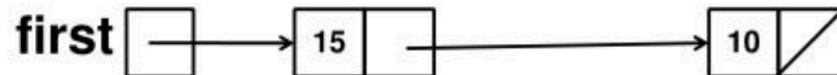
- `struct *node p=first;`
- `struct node * cur= find(p, 20);`
- `struct node *tmp;`
- `tmp = cur->next;`
- `cur->value = tmp->value;`



- `cur->next = tmp->next;`



- `free(tmp);`



SIRS Forms

- You grade me and Manni online.
- You are encouraged to write comments, not just filling in those bubbles
- The Department Chair will read them
- The College Deans will get the statistics
- Manni and I will read every word carefully SIRS forms
- If you'd like to leave additional feedback, use the Piazza thread.