هياكل البيانات

المرحلة الثانية

محاضرة(10)

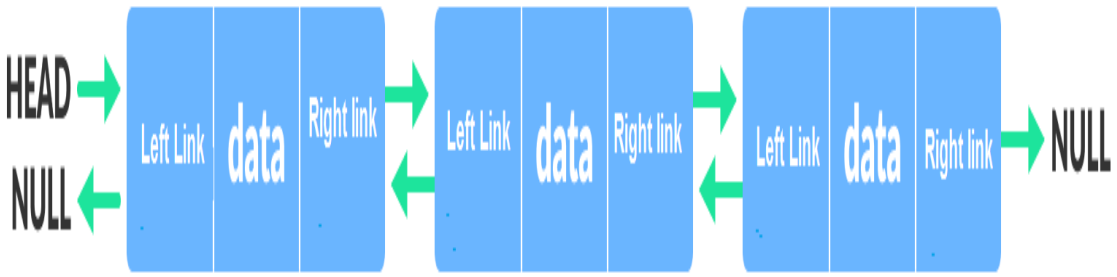
م.م فرح معاذ جاسم

Farahmaath86@uoanbar.edu.iq

DOUBLE LINKED LIST (DLL)

- A double linked list is a two-way list in which all nodes will have two links. This helps in accessing both successor node and predecessor node from the given node position. It provides bi-directional traversing. Each node contains three fields:

- Left link.
- Data.
- Right link.



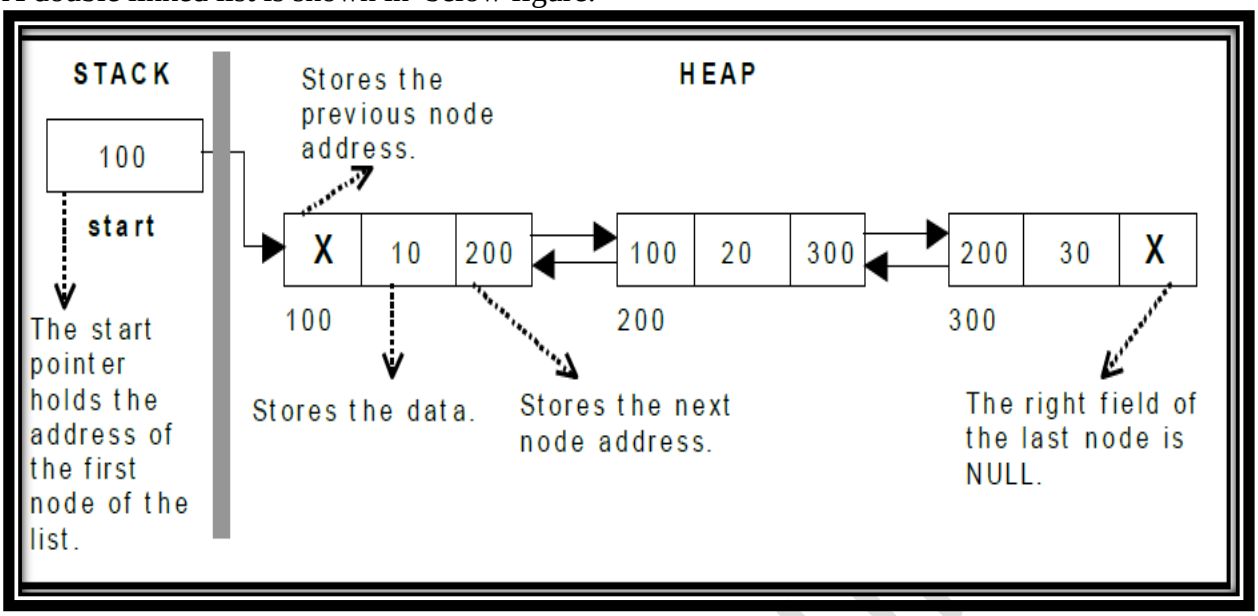
- The left link points to the predecessor node and the right link points to the successor node. The data field stores the required data.

- Many applications require searching forward and backward thru nodes of a list. For example searching for a name in a telephone directory would need forward and backward scanning thru a region of the whole list.

- The basic operations in a double linked list are:

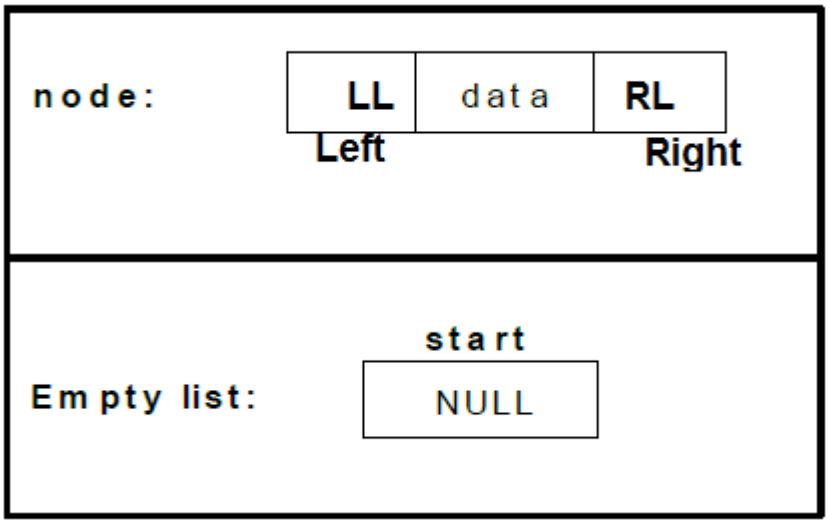
- Creation.
- Insertion.
- Deletion.
- Traversing.

A double linked list is shown in below figure:



- The following code gives the structure definition

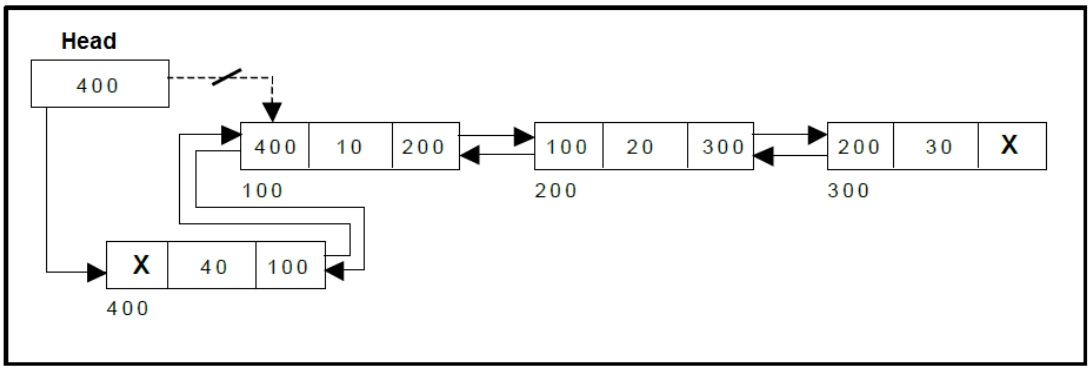
```
struct node
{
        int Data;
        node *LL,*RL;
};
node *head=NULL;
```



INSERT NODE AT FIRST OF DOUBLE LINKED LIST:

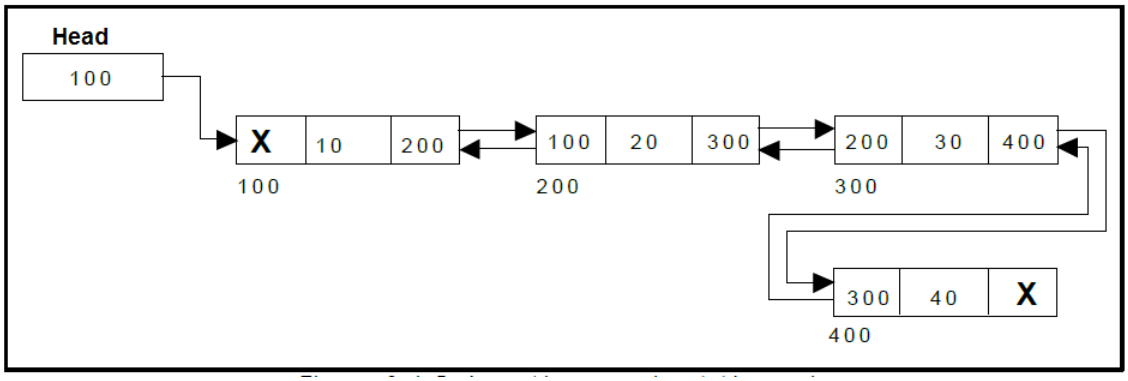
```
void insertF(int n)
{
        node *X=new node;
        X->Data=n;
        X->LL=NULL;

        if (head==NULL)
        X->RL=NULL;
        else
        {X->RL=head;
        head->LL=X;
        }
        head=X;
}
```



INSERT NODE AT THE END OF DOUBLE LINKED LIST:

```
void insertEnd(int n)
{
        node *X=new node;
        X->Data=n;
        X->RL =NULL;
        node *q=head;
        while (q->RL!=NULL)
        q=q->RL;
        q->RL =X;
        X->LL =q;
}
```



////////////////////////////////

INSERT NODE AT THE MID OF DOUBLE LINKED LIST:

```
void insertMid(int n, int y)
{
        node *X=new node;
        X->Data=n;
        node *q=head;
        while (q->Data!=y)
        q=q->RL;
        X->RL=q->RL;
        q->RL->LL=X;
        q->RL=X;
        X->LL=q;
}
```