

Doubly Linked Circular List

In circular or singly linked list, if we want to insert data item in the middle of the list, we have to search the entire list. For example consider following example shown in Figure 4.1. To add a new node C before a node D, we can change the pointer field of C to point to D, but we don't know the address of the node B that precedes the node D.

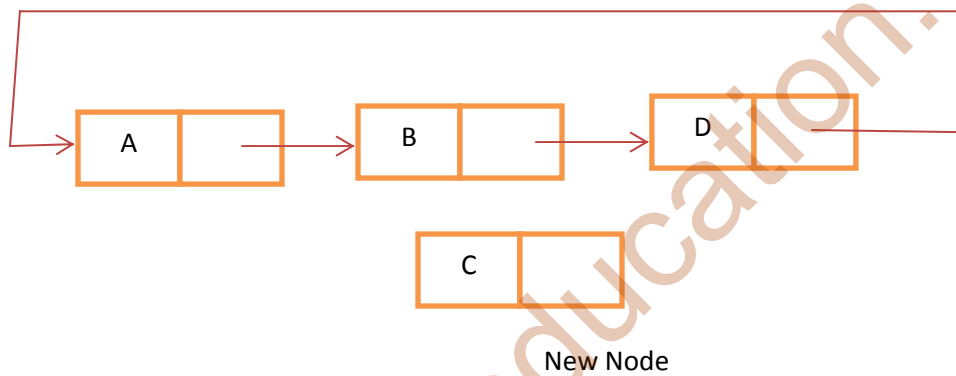


Figure: 4.1

To find the address of the node B, we have to search the entire list, which takes time. To avoid this kind of inefficient process, a doubly circular linked list is introduced in which each node has two pointers such as forward (next) link and the backward (previous) link. The forward link points to the next node in the list and the backward link points to the previous or preceding node, as shown in Figure 4.2.

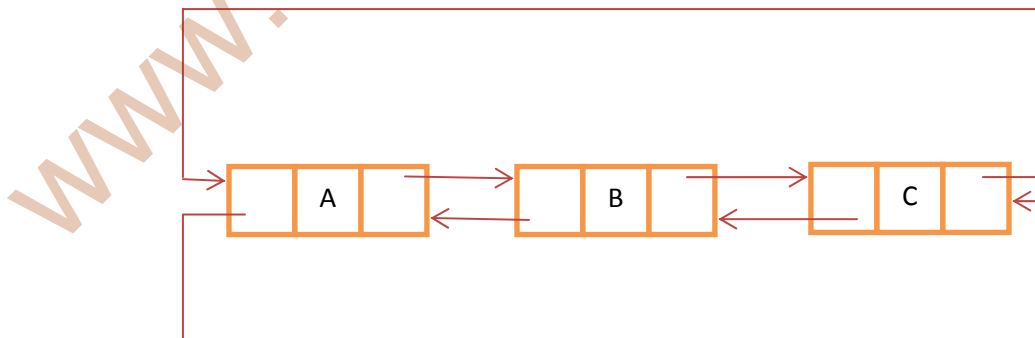


Figure: 4.2

In a doubly linked circular list, the forward link of the last node points to the first node and the backward link of the first node points to the last node. Unlike the circular linked lists, we can easily insert a node in the middle of a doubly linked circular list, because it is not necessary to search the entire list to find the address of the preceding node. The figure below shows, how the node D is inserted before the node B in the doubly linked circular list.

To insert a new node D, the forward link of the node A now points to the node D and the backward link of the node D points to the node A. Similarly the forward link of D points to the node B and the backward link of the node B points to the node C.

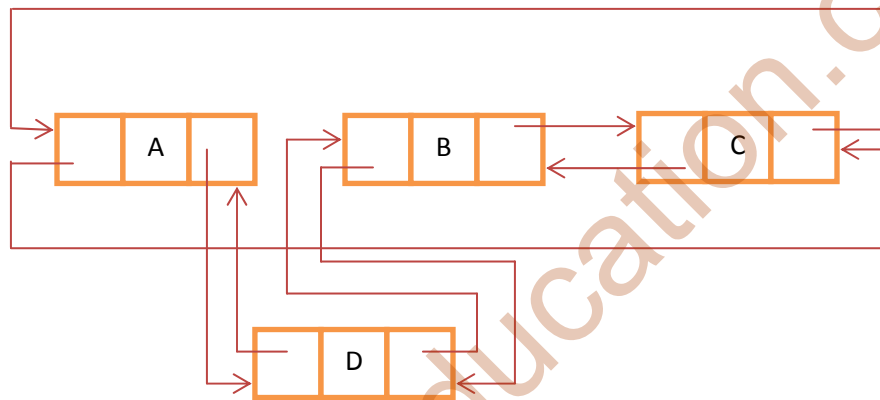


Figure: 4.3

The below figure 4.4 illustrates the deletion of a node B from the doubly linked circular list:

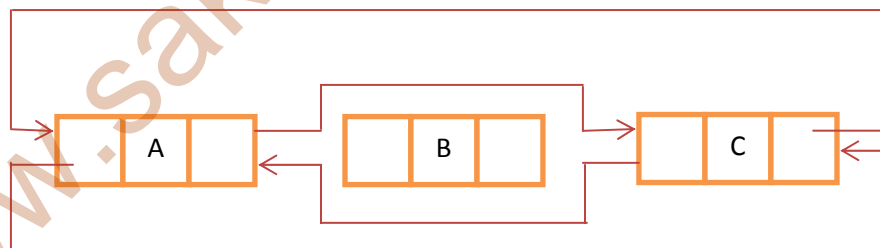


Figure: 4.4

In the above figure, the forward link points to the node C and the backward link points to node A, so that the node A is automatically deleted from the list.

Applications of Linked List

- Linked lists are used to implement the stacks and queues and trees.

- i) If we restrict the process of insertions at end of list and deletions at other end, then we have a model of queue.
- ii) If we restrict the process of insertions and deletions to only beginning of the list, then we have a model of stack. Lists, queues and stacks all are inherently one dimensional.
- A tree represents a two-dimensional linked list. A tree can be implemented by using doubly linked list. Trees are frequently encountered in our daily life. One example is organizational chart of large company.
- Circular linked lists are used to in managing the resources of a computer. We can also use to implement the stack and queue.

www.sakshieducation.com