Insertion in doubly linked list at beginning

As in doubly linked list, each node of the list contain double pointers therefore we have to maintain more number of pointers in doubly linked list as compare to singly linked list.

There are two scenarios of inserting any element into doubly linked list. Either the list is empty or it contains at least one element. Perform the following steps to insert a node in doubly linked list at beginning.

- Allocate the space for the new node in the memory. This will be done by using the following statement.

```
ptr = (struct node *)malloc(sizeof(struct node));
```

- Check whether the list is empty or not. The list is empty if the condition head == NULL holds. In that case, the node will be inserted as the only node of the list and therefore the prev and the next pointer of the node will point to NULL and the head pointer will point to this node.

```
ptr->next = NULL;
    ptr->prev=NULL;
    ptr->data=item;
    head=ptr;
```

- In the second scenario, the condition **head == NULL** become false and the node will be inserted in beginning. The next pointer of the node will point to the existing head pointer of the node. The prev pointer of the existing head will point to the new node being inserted.
- This will be done by using the following statements.

```
ptr->next = head;
  head→prev=ptr;
```

Since, the node being inserted is the first node of the list and therefore it must contain NULL in its prev pointer. Hence assign null to its previous part and make the head point to this node.

```
ptr→prev =NULL
head = ptr
```

Algorithm :

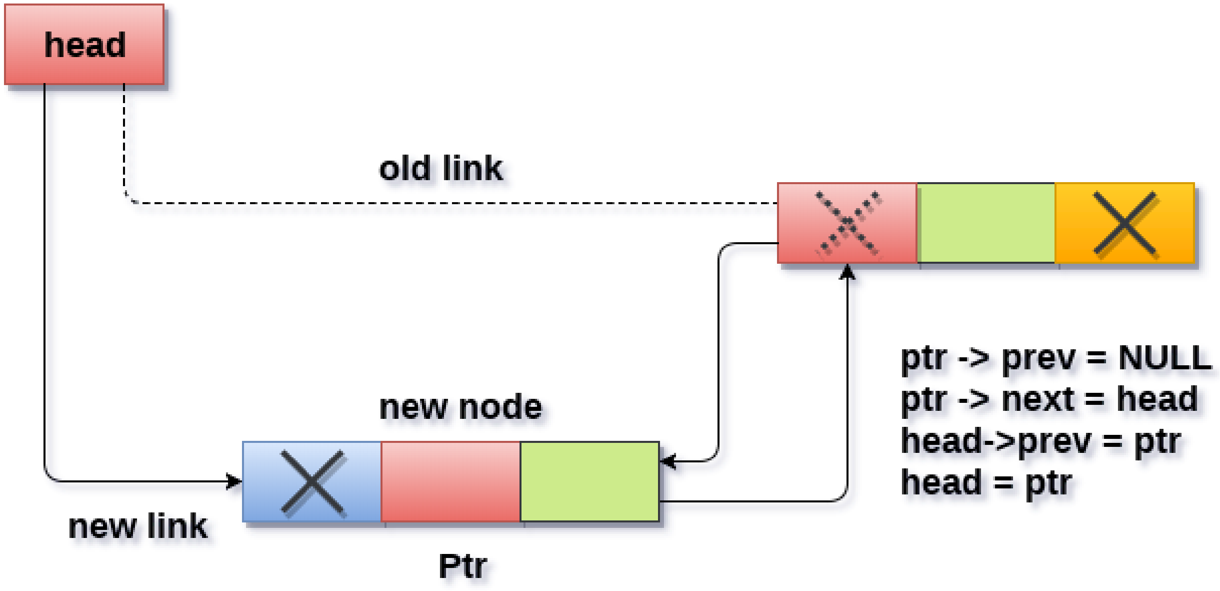
- **Step 1:** IF ptr = NULL

  ```
   Write OVERFLOW
  Go to Step 9
  [END OF IF]
  ```

- **Step 2:** SET NEW_NODE = ptr
- **Step 3:** SET ptr = ptr -> NEXT
- **Step 4:** SET NEW_NODE -> DATA = VAL
- **Step 5:** SET NEW_NODE -> PREV = NULL
- **Step 6:** SET NEW_NODE -> NEXT = START
- **Step 7:** SET head -> PREV = NEW_NODE
- **Step 8:** SET head = NEW_NODE

⇧

- **Step 9:** EXIT



ptr -> prev = NULL
ptr -> next = head
head->prev = ptr
head = ptr

Insertion into doubly linked list at beginning

C Function

⇧

```
#include<stdio.h>
#include<stdlib.h>
void insertbeginning(int);
struct node
{
    int data;
    struct node *next;
    struct node *prev;
};
struct node *head;
void main ()
{
    int choice,item;
    do
    {
        printf("\nEnter the item which you want to insert?\n");
        scanf("%d",&item);
        insertbeginning(item);
        printf("\nPress 0 to insert more ?\n");
        scanf("%d",&choice);
    }while(choice == 0);
}
void insertbeginning(int item)
{

   struct node *ptr = (struct node *)malloc(sizeof(struct node));
   if(ptr == NULL)
   {
       printf("\nOVERFLOW");
   }
   else
   {


   if(head==NULL)
   {
       ptr->next = NULL;
       ptr->prev=NULL;
       ptr->data=item;
       head=ptr;
   }
   else
   {
       ptr->data=item;
       ptr->prev=NULL;
       ptr->next = head;
       head->prev=ptr;
```

```
      head=ptr;
    }
  }


  }
```

Output

```
Enter the item which you want to insert?
12

Press 0 to insert more ?
0

Enter the item which you want to insert?
23

Press 0 to insert more ?
2
```

Output

```
Enter the item which you want to insert?
12

Press 0 to insert more ?
2
```

← prev next →

Please Share

Learn Latest Tutorials

Swift

Pig

Flask

C. Graphics

Automata

Testing

⇧

Preparation

Aptitude

Reasoning

Verbal A.

Interview

B.Tech / MCA

DBMS

DS

DAA

OS

C. Network

Compiler D.

COA

D. Math.

E. Hacking

Web Tech.

Cyber Sec.

C

C++ Java .Net Python

Programs

Control S.

⇧