Doubly Linked List Representation

Doubly Linked List

As per the above illustration, following are the important points to be considered.

Doubly Linked List contains a link element called first and last.

Each link carries a data field(s) and two link fields called next and prev.

Each link is linked with its next link using its next link.

Each link is linked with its previous link using its previous link.

The last link carries a link as null to mark the end of the list.

Basic Operations

Following are the basic operations supported by a list.

Insertion − Adds an element at the beginning of the list.

Deletion − Deletes an element at the beginning of the list.

Insert Last − Adds an element at the end of the list.

Delete Last − Deletes an element from the end of the list.

Insert After − Adds an element after an item of the list.

Delete − Deletes an element from the list using the key.

Display forward − Displays the complete list in a forward manner.

Display backward − Displays the complete list in a backward manner.

Insertion Operation

Following code demonstrates the insertion operation at the beginning of a doubly linked list.

Example

```
//insert link at the first location
void insertFirst(int key, int data) {

   //create a link
   struct node *link = (struct node*) malloc(sizeof(struct node));
   link->key = key;
   link->data = data;
   if(isEmpty()) {
      //make it the last link
      last = link;
   } else {
      //update first prev link
      head->prev = link;
   }

   //point it to old first link
   link->next = head;

   //point first to new first link
   head = link;
}
```

Deletion Operation

Following code demonstrates the deletion operation at the beginning of a doubly linked list.

Example

```
//delete first item
struct node* deleteFirst() {

   //save reference to first link
   struct node *tempLink = head;
   //if only one link
   if(head->next == NULL) {
      last = NULL;
   } else {
      head->next->prev = NULL;
   }

   head = head->next;

   //return the deleted link
   return tempLink;
}
```

Insertion at the End of an Operation

Following code demonstrates the insertion operation at the last position of a doubly linked list.

Example

```
//insert link at the last location
void insertLast(int key, int data) {

   //create a link
   struct node *link = (struct node*) malloc(sizeof(struct node));
   link->key = key;
   link->data = data;
```