Southern Technical University
Technical Institute / Qurna
Dep. of Computer Systems Techniques

Second class
Subject : Data Structures
Lecturer : Israa Mahmood Hayder
Lecture no.10,11

الهياكل الموصولة

(Linked Structures)

- الاسبوع العاشر- الحادي عشر-

العاشر - الحادي عشر	١. القائمة الثنائية / قراءة العناصر – طباعة القائمة ٢. القائمة الدائرية / قراءة العناصر – طباعة القائمة

B// Rationale (مبررات الوحدة) :-

The student will learn about doubly linked lists and how linked lists work. Programmers choose linked lists and operations on these lists.

C// Central (الفكرة المركزية):-

- Operations on doubly Linked List

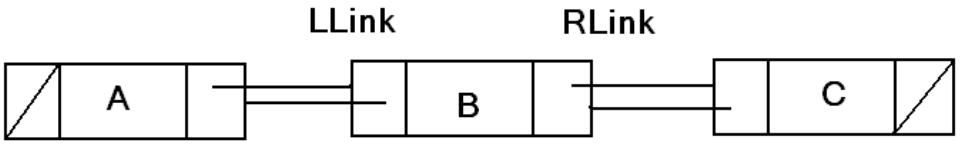
D// Objectives (أهداف الوحدة):-

After studying this unit, the student will be able to write Operations on single and doubly Linked List

Operation on Doubly Linked List :-

In these lists there are two pointers the first points to the previous element (left-link) and the second points to the next element (right-link).

Ex:-



Creating Double linked list of N nodes in C++ :-

```
main( )
{ int n ;
  struct node { int data;
                 struct node *llink, ;
                struct node *rlink, ;
               }
  struct node *p= new  struct node ;
  struct node *start=p;
  struct node *p2;
   p.rlink=null;
   cin>>n;
for(i=1; i<=n ; i++)
   { cin>> p.data;
     if  i !=n then
         struct node *p2=new struct node;
     else
        p2=null;
     p.llink=p2;
     p2.rlink=p;
     p=p2;
}
```

* Insertion (of the first element):-

Algorithm :-

```
new(node);
read(node ↑.data);
node↑.Llink=nil;
node↑.Rlink=nil;
```

* Insertion at the beginning of the list:-

Algorithm :-

```
new(node);
read(node ↑.data);
node↑.Llink=nil;
node↑.Rlink=first;
first=node;
```

* Insert before node(m):-

Algorithm :-

```
new(node);
read(node ↑.data);
node↑.Llink=m↑.Llink;          or    node↑.Llink= p
node↑.Rlink= p↑.Rlink;         or    node↑.Rlink= m
p↑.Rlink = node;
m↑.Llink = node;
```

* Insert after node(m):-

Algorithm :-

```
new(node);
read(node ↑.data);
node↑.Llink=p↑.Llink;          or    node↑.Llink= m
node↑.Rlink= m↑.Rlink;         or    node↑.Rlink= p
```

* Insert at the end of the list:-

Algorithm :-

```
new(node);
read(node ↑.data);
node↑.Llink=m;
node↑.Rlink=nil;
m↑.Rlink = nil;
```

Printing The nodes data of Lists :-

Algorithm :-

```
P= left;
While (p↑.Rlink)< > nil do
```

```
  Begin
     writeln(p↑.data)
     p= p↑.Rlink
   end;
```

Print The list from left to right :-

Algorithm :-

```
P= right;
While (p↑.Llink)< > nil do
  Begin
     writeln(p↑.data)
     p= p↑.Llink
   end;
```

Delete of the first element :-

Algorithm :-

```
m↑.Llink = p↑.Llink   or nil
dispose (p);      { or delete (p)}
```

Delete the last element :-

Algorithm :-

```
m↑.Rlink = nil
dispose (p);      { or delete (p)}
```

Delete the element(P) :-

Algorithm :-

```
x= p↑.Llink ;
m= p↑.Rlink
m↑.Rlink = x;
x↑.Llink= m
dispose (p);      { or delete (p)}
```

Quiz1:

1) Delete the last element of doubly linked list
2) Print the doubly linked list from right to left.
3) Write the algorithm to insert item to the beginning of doubly linked queue

Rferences:

1- Data Structures Demystified, by Jim Keogh and Ken Davidson, ISBN:0072253592, McGraw-Hill/Osborne © 2004

2- هياكل البيانات / الطبعة الثانية، تاليف د.عصام الصفار،اصدارات السفير للنشر/ بغداد، ٢٠٠١

3- -الحقيبة التعليمية مادة "هياكل البيانات "، اعداد : نفارت الياس يوسف ،المعهد التقني كركوك