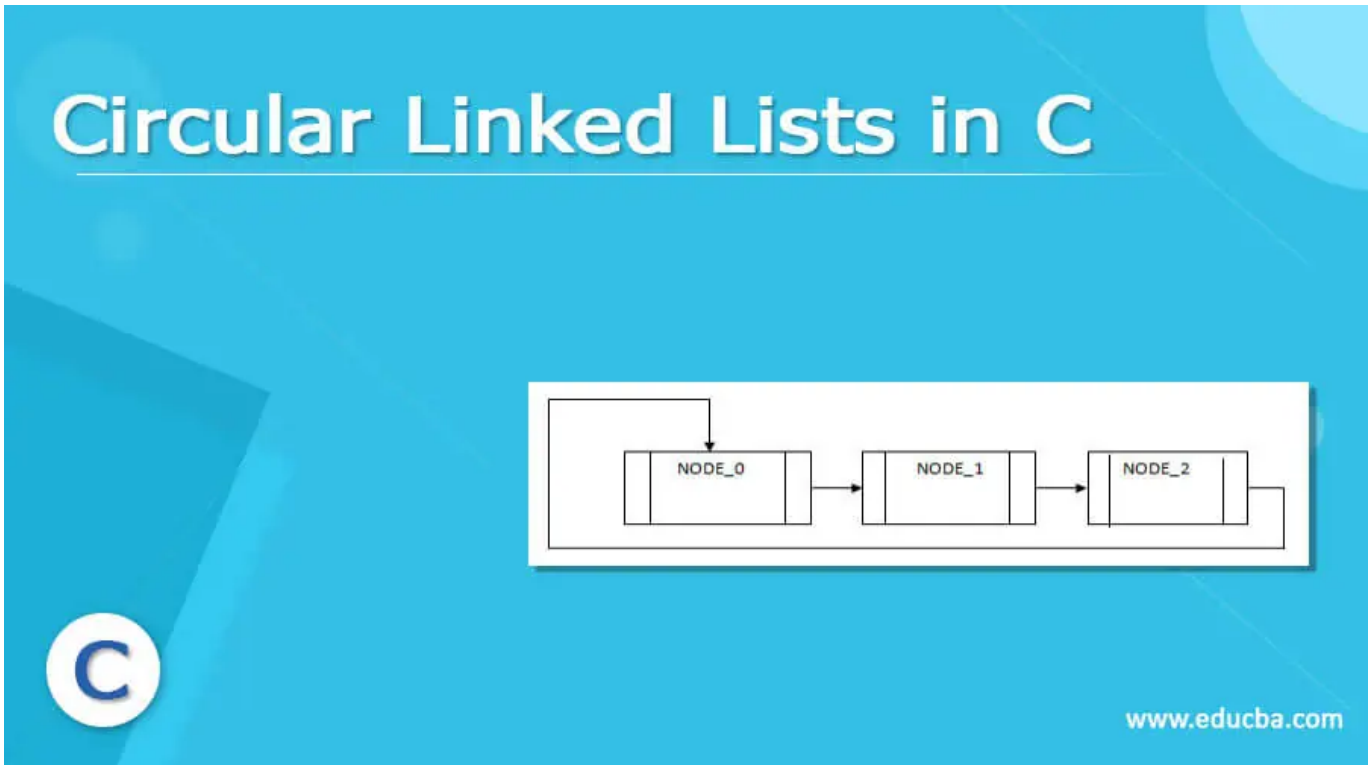
EDUCBA
(https://www.educba.com/software-development/)

← (https://www.educba.com/circular-doubly-linked-list-in-c/)

→ (https://www.educba.com/bfs-algorithm-in-c/)



Definition of Circular Linked Lists in C

Circular linked list is a different form of linked list. In a circular linked list, every node has connected to the next node and the previous node in the sequence as well as the last node has a link or connection to the first node from the list that we called a circular linked list. Normally

linked list.

Syntax:

Basically, we can perform the different operations on circular linked lists such as insert; delete and traverse, etc. so here we see the syntax of insert operation with creating a new node as follows.

```
void create()
{
node *n_node;
n_node=(node*)malloc(sizepof(node));
printf("\n Enter new value for new node: ");
scanf("%d",&n_node->data);
n_node->next=null;
if(rear=null)
front=rear=n_node;
else
{
rear->next=n_node;

rear=n_node;
}
rear-> next=front;
```

new node with the size of the node. After that, we use the pointer concept to create a new node and point to the next and previous node in the circular linked list. In this way, we can perform different insert operations that is insert a node at the start, at the end, and at specified locations as well as we can also perform the delete operation.

How Circular linked lists works in C?

Now let's see how the circular linked list works as follows. Basically, we perform the following operation as follows.

Insertion

Basically, insertion is used to add a new node into the circular linked list at a specified position as per requirement. The first step in insertion operation is that we need to create a new node by using the above code.

Now there are two different possibilities as follows.

The first possibility is that we can insert nodes at the current position in the circular linked list. This operation corresponds to the insert at the beginning and end of the circular linked list because the beginning and end are the same in this linked list.

The second possibility is to perform insert after indexed node. Now let's see the steps to insert operation as follows.

1. First we need to break the existing link between the nodes.

2. After that, connect the new node link to the existing node.

3. Now the last node link will point to the newly inserted node.

All this insert operation is shown in the diagram below as follows

(https://www.educba.com/software-development/)

So in this way, we can perform the insertion operation.

Deletion Operation

Now assume there are three nodes in the circular linked list, so the following possibilities may generate.

1. The First possibility is that we can delete the current node.

2. The Second possibility is that we can delete after node.

Perform delete at beginning and end

1. First we need to traverse the first node to the last node.

2. Delete from the last node.

3. Delete link between last node and next node.

4. After that we need to connect the last node to the first node.

5. Free first node from the circular linked list.

All delete operations as shown in the figure below are as follows.

(https://www.educba.com/software-development/)

Now remove the link between the first node and next node as shown in the below figure as follows.

Now free the first node from the circular linked list as shown in the below figure as follows.

The same way we can perform the delete after node by using the same step.

Memory representation of a circular linked list

In the accompanying picture, memory representation of a circular linked list containing marks of a student in 3 subjects. Nonetheless, the picture shows a brief look at how the circular linked list is being put away in the memory. The beginning or top of the rundown is highlighting the

component with file 1 and containing 20 marks in the information part and 3 in the follo part. This means that it is connected with the hub that is being put away on the third list.

Nonetheless, because of the way that we are thinking about the circular linked list connected

(https://www.educba.com/software-development/)

Examples

Now let's see the example of a circular linked list as follows.

```
#include<stdio.h>
#include<stdlib.h>
typedef struct Node
{
int data;
struct Node *next;
}node;
node *start=NULL,*end=NULL,*temp_data;
void create_node();
void del_node();
void display_node();
int main()
{
int choice;
do
{
printf("\nMenu\n 1 create node: ");
printf("\n2 to delete : ");
```

(https://www.educba.com/software-development/)

```
switch(choice)
{
case 1:
create_node();
break;
case 2:
del_node();
break;
case 3:
display_node();
break;
case 4:
return 1;
default:
printf("\nEntered choice is invalid :");
}
}while(1);
return 0;
}
void create_node()
{
node *n_node;
n_node=(node*)malloc(sizeof(node));

printf("\nEnter value : ");
scanf("%d",&n_node->data);
n_node->next=NULL;
```

(https://www.educba.com/software-development/)

```
end->next=n_node;
end=n_node;
}
end->next=start;
}
void del_node()
{
temp_data=start;
if(start==NULL)
printf("\nCircular linked list is Underflow :");
else
{
if(start==end)
{
printf("\n%d",start->data);
start=end=NULL;
}
else
{
printf("\n%d",start->data);
start=start->next;
end->next=start;
}

temp_data->next=NULL;
free(temp_data);
}
```

(https://www.educba.com/software-development/)

```
if(start==NULL)
printf("\nCircular linked list is Empty");
else
{
printf("\n");
for(;temp_data!=end;temp_data=temp_data->next)
printf("\n%d address=%u next=%u\t",temp_data-
>data,temp_data,temp_data->next);
printf("\n%d address=%u next=%u\t",temp_data-
>data,temp_data,temp_data->next);
}
}
```

Explanation

By using the above code we try to implement a circular linked list. The end out of the above code we illustrate by using the following screenshot as follows.

(https://www.educba.com/software-development/)

(https://www.educba.com/software-development/)

Conclusion

We hope from this article you learn the Circular Linked List in C. From the above article, we have learned the basic syntax of Circular Linked List and we also see different examples of Circular Linked List. From this article, we learned how and when we use the Circular Linked List in C.

Recommended Articles

This is a guide to Circular Linked Lists in C. Here we discuss definition, syntax, how the circular linked list works examples with code implementation. You may also have a look at the following articles to learn more –

1. Linked List in C (https://www.educba.com/linked-list-in-c/)
2. Circular Linked List in Data Structure (https://www.educba.com/circular-linked-list-in-data-structure/)
3. Queue in C (https://www.educba.com/queue-in-c/)
4. Merge Sort in Data Structure (https://www.educba.com/merge-sort-in-data-structure/)

About Us

(https://www.educba.com/software-development/)

Privacy Policy (https://www.educba.com/privacy-policy/?source=footer)

Apps

iPhone & iPad (https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8)

Android (https://play.google.com/store/apps/details?id=com.educba.www)

Resources

Free Courses (https://www.educba.com/software-development/free-courses/?source=footer)

Java Tutorials (https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer)

Python Tutorials (https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer)

All Tutorials (https://www.educba.com/software-development/software-development-tutorials/?source=footer)

Certification Courses

All Courses (https://www.educba.com/software-development/courses/?source=footer)

Software Development Course - All in One Bundle (https://www.educba.com/software-development/courses/software-development-course/?source=footer)

Become a Python Developer (https://www.educba.com/software-development/courses/python-certification-course/?source=footer)

Java Course (https://www.educba.com/software-development/courses/java-course/?source=footer)

(https://www.educba.com/software-development/)

VB.NET Course (https://www.educba.com/software-development/courses/vb-net-course/?source=footer)

PHP Course (https://www.educba.com/software-development/courses/php-course/?source=footer)