GeeksQuiz

Computer science mock tests for geeks

Search

- Home
- Latest Questions
- Articles
- C/C++ Programs
- Contribute
- Books

Subscribe

Linked List | Set 1 (Introduction)

March 8, 2013

Like arrays, Linked List is a linear data structure. Unlike arrays, linked list elements are not stored at contiguous location; the elements are linked using pointers.

Why Linked List?
Arrays can be used to store linear data of similar types, but arrays have following limitations.
1) The size of the arrays is fixed: So we must know the upper limit on the number of elements in advance. Also, generally, the allocated memory is equal to the upper limit irrespective of the usage.
2) Inserting a new element in an array of elements is expensive, because room has to be created for the new elements and to create room existing elements have to shifted.

For example, in a system if we maintain a sorted list of IDs in an array id[].

id[] = [1000, 1010, 1050, 2000, 2040].

And if we want to insert a new ID 1005, then to maintain the sorted order, we have to move all the elements after 1000 (excluding 1000).
Deletion is also expensive with arrays until unless some special techniques are used. For example, to delete 1010 in id[], everything after 1010 has to be moved.

Advantages over arrays
1) Dynamic size
2) Ease of insertion/deletion

Drawbacks:
1) Random access is not allowed. We have to access elements sequentially starting from the first node. So we cannot do binary search with linked lists.
2) Extra memory space for a pointer is required with each element of the list.

Representation in C:
A linked list is represented by a pointer to the first node of the linked list. The first node is called head. If the linked list is empty, then value of head is NULL.
Each node in a list consists of at least two parts:
1) data
2) pointer to the next node
In C, we can represent a node using structures. Below is an example of a linked list node with an integer data.

```
struct node
{
  int data;
  struct node *next;
};
```

First Simple Linked List in C Let us create a simple linked list with 3 nodes.

```
#include<stdio.h>
#include<stdlib.h>

struct node
{
  int data;
  struct node *next;
};

// Program to create a simple linked list with 3 nodes
int main()
{
  struct node* head = NULL;
  struct node* second = NULL;
  struct node* third = NULL;

  // allocate 3 nodes in the heap
  head = (struct node*)malloc(sizeof(struct node));
  second = (struct node*)malloc(sizeof(struct node));
  third = (struct node*)malloc(sizeof(struct node));

  /* Three blocks have been allocated  dynamically.
     We have pointers to these three blocks as first, second and third
       head           second           third
        |                |               |
        |                |               |
    +---+-----+     +----+----+     +----+----+
    | #  | #  |     | #  | #  |     |  # |  # |
    +---+-----+     +----+----+     +----+----+

   # represents any random value.
   Data is random because we haven’t assigned anything yet  */

  head->data = 1; //assign data in first node
  head->next = second; // Link first node with the second node
  /* data has been assigned to data part of first block (block
    pointed by head).  And next pointer of first block points to
    second.  So they both are linked.

       head          second         third
        |              |              |
        |              |              |
    +---+---+     +----+----+     +-----+-----+
    | 1  | o----->| #  | #  |     |  #  | #  |
    +---+---+     +----+----+     +-----+-----+
  */

  second->data = 2; //assign data to second node
  second->next = third;

  /* data has been assigned to data part of second block (block pointed by
     second). And next pointer of the second block points to third block.
    So all three blocks are linked.

       head         second         third
        |             |             |
        |             |             |
    +---+---+     +---+---+     +----+----+
    | 1  | o----->| 2 | o-----> |  # |  # |      */

  third->data = 3; //assign data to third node
  third->next = NULL;

  /* data has been assigned to data part of third block (block pointed
    by third). And next pointer of the third block is made NULL to indicate
    that the linked list is terminated here.

     We have the linked list ready.

           head
             |
             |
        +---+---+     +---+---+       +----+------+
        | 1  | o----->|  2  | o-----> |  3 | NULL |
        +---+---+     +---+---+       +----+------+


    Note that only head is sufficient to represent the whole list.  We can
    traverse the complete list by following next pointers.    */

  getchar();
  return 0;
}
```

Linked List Traversal
In the previous program, we have created a simple linked list with three nodes. Let us traverse the created list and print the data of each node. For traversal, let us write a general purpose function printList() that

prints any given list.

```c
#include<stdio.h>
#include<stdlib.h>

struct node
{
  int data;
  struct node *next;
};

// This function prints contents of linked list starting from the given node
void printList(struct node *n)
{
  while (n != NULL)
  {
     printf(" %d ", n->data);
     n = n->next;
  }
}

int main()
{
  struct node* head = NULL;
  struct node* second = NULL;
  struct node* third = NULL;

  // allocate 3 nodes in the heap
  head  = (struct node*)malloc(sizeof(struct node));
  second = (struct node*)malloc(sizeof(struct node));
  third  = (struct node*)malloc(sizeof(struct node));

  head->data = 1; //assign data in first node
  head->next = second; // Link first node with the second node

  second->data = 2; //assign data to second node
  second->next = third;

  third->data = 3; //assign data to third node
  third->next = NULL;

  printList(head);

  getchar();
  return 0;
}
```

Output:

```
 1  2  3
```

You may like to try **Practice MCQ Questions on Linked List**

We will soon be publishing more posts on Linked Lists.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Questions:

- Binary Heap
- Delete all occurrences of a given key in a linked list
- How to create mergable stack?
- Deque | Set 1 (Introduction and Applications)
- A data structure for n elements and O(1) operations
- Convert left-right representation of a bianry tree to down-right
- Print level order traversal line by line
- C Program for Red Black Tree Insertion

Like 18 Tweet 0 +1 2

60 Comments GeeksQuiz Login

Recommend 1 Share Sort by Best

Join the discussion…

sonu431 • 9 months ago

What does struct node*head=NULL mean? and what if it were like struct node*head; only and we passed it as a pointer. what is the difference b/w these two pointers?
plz answer it anybody with explanation

25 | • Reply • Share ›

Ankit Singh sonu431 • 9 months ago

We must ensure that before allocating memory addresses to the nodes,the node pointers must contain null values so that they cannot point to any random memory location which is surely undesirable.That is why the statement struct node*head=NULL.

13 | • Reply • Share ›

sonu431 Ankit Singh • 9 months ago

take this as an example.. here

```
void reverse(struct node** head_ref)
{
struct node* prev = NULL;
struct node* current = *head_ref;
struct node* current = *head_ref;
struct node* next;
while (current != NULL)
{
next = current->next;
current->next = prev;
prev = current;
current = next;
}
*head_ref = prev;
}
```

in above code *prev=NULL,
why not *next=NULL, but if we make it point to
NULL it doesnt make any difference..
but in *prev=NULL instead of it if we take as only
*prev it is giving not proper o/p.. plz explain..

1 ^ | ˅ • Reply • Share ›

unlimited700 ➦ sonu431 • 8 months ago

If you will not initialize prev it will have random value right ? Now when you execute first time above while loop the instruction current->next = prev ; will assign next of current any random value rather than NULL and this current will work as a last node after execution of whole function . In this case when we try to traverse the whole list we can are not sure to terminate because this last node will point to any random value rather than NULL (As you know we usually put a condition in which we check traverse the list while we not reach the NULL) .

5 ^ | ˅ • Reply • Share ›

Manas Sharma • 6 months ago

What does -> operator do? And why can't we use dot operator for the same purpose i.e (head.data=1;) instead of (head->data=1;)

5 ^ | ˅ • Reply • Share ›

Gaurav Suman ➦ Manas Sharma • 6 months ago

First, head.data is wrong , it should be (*head).data because u need to dereference the head pointer first.

-> provides short hand for this.
As an example consider changing next element from head.
using dot operator: (*((*head).next)).data = 1
using -> operator: head->next->data = 1

13 ^ | ˅ • Reply • Share ›

Manas Sharma ↪ Gaurav Suman • 6 months ago

thanks!

⌃ | ⌄ • Reply • Share ›

Pradeep • 10 months ago

Why getcher() is used at end of the program?

5 ⌃ | ⌄ • Reply • Share ›

Alex ↪ Pradeep • 10 months ago

to stop output screen...until you enter a character.

5 ⌃ | ⌄ • Reply • Share ›

vimal • 5 months ago

why there is no previous and next button on this site..

4 ⌃ | ⌄ • Reply • Share ›

Manas Sharma • 6 months ago

In the statement

head = (struct node*)malloc(sizeof(struct node));

What is the significance of type-casting (struct node*)? And how will be my result be affected if I don't do this type-casting.

And what is the significance of * in this type-cast?

3 ⌃ | ⌄ • Reply • Share ›

anonymous ↪ Manas Sharma • 6 months ago

Because malloc returns a pointer to void, i.e., it is simply allocating chunks of memory with no regard as to the data that will be stored there.

2 ⌃ | ⌄ • Reply • Share ›

Rahul K Kaushik ↪ Manas Sharma • 2 months ago

as malloc returns a pointer we need to typecast it into a pointer which we do through (node *)

1 ⌃ | ⌄ • Reply • Share ›

peru • 3 months ago

please..cut down some advertisements...they distract a lot..

2 ⌃ | ⌄ • Reply • Share ›

kaps • 5 months ago

In the declaration:

struct node

```
{
int data;
struct node *next;
};
```

how are we able to declare a pointer to a structure, which is not yet declared fully ?

2 ^ | ⌄ • Reply • Share ›

Rajat Kumar Seth ➦ kaps • 5 months ago

struct node is used to create a user defined data type to store data in Primary Memory (RAM)..
Since we are making a user defined datatype, so it can include all the predefined datatypes which includes pointers also..

^ | ⌄ • Reply • Share ›

Satyam • 8 months ago

what is difference between 'struct node *temp' and 'struct node* temp' please comment fast

2 ^ | ⌄ • Reply • Share ›

Adauta Garcia Ariel ➦ Satyam • 2 months ago

Hello Satyam. It's something about "style" I've found people who likes write:

int* iptr;

Other people:

int *iptr;

Just we have to be careful when we declare more than one variable in one line.

int *iptr, var;

iptr is a pointer to int, "var" is an int.

If we want to declare "var" as a pointer we do it as follows.

int *iptr, *var;

There are people who read that declarations as follows.

"var" is variable pointer wich point to an int

see more

3 ^ | ⌄ • Reply • Share ›

Shaurabh ➦ Satyam • 8 months ago

No difference!

2 ⌃ | ⌄ • Reply • Share ›

RK ➦ Satyam • 7 months ago

There is no difference, but it would be better for understanding if you use 'struct node *temp'

⌃ | ⌄ • Reply • Share ›

Himanshu Dagar • a year ago

very very helpful discussion

2 ⌃ | ⌄ • Reply • Share ›

Ashish Patil • 10 months ago

very helpful post,

1 ⌃ | ⌄ • Reply • Share ›

sneha • 2 days ago

in the declaration section,
why we have used struct node *next??
i mean why struct node?

⌃ | ⌄ • Reply • Share ›

rishabh jain • a month ago

what is the use of getchar() method

⌃ | ⌄ • Reply • Share ›

Rahul Ranjan • 2 months ago

since as said in introduction of linked list " Unlike arrays, linked list elements are not stored at contiguous location; the elements are linked using pointers." can we implement linked list in java? as java does not have the concept of pointer.

⌃ | ⌄ • Reply • Share ›

WantAns • 2 months ago

function to clear stream stdin in linux...??

⌃ | ⌄ • Reply • Share ›

Selvaraj Kannaiyan • 2 months ago

can anyone give sample java code for basic list node creation..

⌃ | ⌄ • Reply • Share ›

Jiya CH • 2 months ago

Write a program to modify a value from a single link list consisting of 7 nodes after searching it?

⌃ | ⌄ • Reply • Share ›

romit • 2 months ago

i want to implement stack using doubly link list and pointer please help

```
int initstack(void **stackHeadPtr, uint32 maxStackSize);
int push(void *stackHeadPtr, uint32 data);
int pop(void *stackHeadPtr, uint32 *data);
int front(void *stackHeadPtr, uint32 *data);
int stackRremoveNthNode(void *stackHeadPtr, int position);
int getStackSize(void *stackHeadPtr, uint32 *size);
```

with following function.

• Reply • Share ›

ash • 2 months ago

In the traversal of linked list,

```
void printList(struct node *n)

{
while (n != NULL) // Should it not be while(n->link=NULL)
{
printf(" %d ", n->data);
n = n->next;
}
}
```

What do you mean by n!=NULL?

• Reply • Share ›

Vishwas ➦ ash • 2 months ago

If you use (n->link =null) the last node will not be printed

• Reply • Share ›

Aditya Goel • 5 months ago

Guys, If you think you understand Linked List, think again, and then again.. Making a Linked List program is not enough.. I was no. 1 programmer in my clg. DS was my strong point and I loved Linked List. Little did I know that I don't really "understand" Linked List. Aahhh.. those days.. proud days.. will never be proud of anything except when I'm 100% sure 100 times. :)

• Reply • Share ›

surbhijain93 ➦ Aditya Goel • a month ago

What made you say so?

• Reply • Share ›

Aditya Goel ➦ surbhijain93 • a month ago

Go thru every link list problem given here, you will know

Go thru every link list problem given here, you will know

• Reply • Share ›

Rohith • 6 months ago

What would be the Java equivalent of this?

• Reply • Share ›

Sony ➦ Rohith • 5 months ago

You can make a one class for linklist and another class for node.

• Reply • Share ›

Sunil Kumar • 7 months ago

great...

• Reply • Share ›

pratuksha • 7 months ago

in above program we have created 3 node..if we want to create more than 3..what will we do???will we apply same method to create 20 or 30 nodes??

• Reply • Share ›

anonymous ➦ pratuksha • 2 months ago

u can make a function to create nodes..

• Reply • Share ›

RK ➦ pratuksha • 7 months ago

Yes, in one way or another, you will have to do the same thing.

• Reply • Share ›

pratuksha ➦ RK • 7 months ago

but its not a good method na..that we create 20 nodes and give seperate name of these 20 nodes..

• Reply • Share ›

RK ➦ pratuksha • 7 months ago

use a loop, look at other articles related to linked lists.

• Reply • Share ›

pratuksha • 7 months ago

why we include stdlib.h..i think without it the program will run successfully,,

• Reply • Share ›

Adauta Garcia Ariel ➦ pratuksha • 2 months ago

stdlib.h contains the malloc function, and the NULL macro.

• Reply • Share ›

Reply • Share ›

AKASH BHARGAVA ➦ pratuksha • 7 months ago

creation of any node comes under standard library function .
So we have to write stdlib.h
program may work on some compilers but it will not work on standard compilers like code-block .

Reply • Share ›

Prado ➦ AKASH BHARGAVA • 5 months ago

malloc Function defined in stdlib.h library

3 • Reply • Share ›

Anon • 7 months ago

if i initialize head->next=NULL and second->next=NULL and so on,the compiler gives me a runtime error.Why does it happen?

Reply • Share ›

Kartik ➦ Anon • 7 months ago

You must allocate memory for head first, then only you can assign some values to its members. Hope it helps.

Reply • Share ›

Anon ➦ Kartik • 7 months ago

thanks a lot...

Reply • Share ›

srinivas devaki • 9 months ago

n = n->next;
at this line aren't you modifying the data of head, i think you have to pre declare a pointer(struct node *p=n;) and then
p = p->next;
this will ensure that our function remains a general purpose function.(I think so)
please correct me if I am wrong

Reply • Share ›

Load more comments

Subscribe Add Disqus to your site Privacy

GeeksQuiz

Like

13,860 people like GeeksQuiz.

Facebook social plugin

-
-
- ## Categories

  - Articles (128)
    - Algorithms (24)
    - C (17)
    - C++ (17)
    - Data Structures (29)
    - DBMS (1)
    - Interview Experiences (6)
    - Java (2)
    - Operating Systems (1)
    - Puzzle (15)
    - Searching and Sorting (10)
  - Programs (35)
  - Quizzes (2,107)
    - Aptitude (2)
    - Computer Science Quizzes (2,104)
      - Algorithms (147)
      - C (208)
      - C++ (129)
      - Computer Organization and Architecture (1)
      - Data Structures (132)

    - DBMS (2)
    - GATE (1,406)
    - Java (51)
    - Operating Systems (28)
  - Web technologies (1)
-
- ## Recent Discussions

  - Zsw-seu

    I agree your answer,it should be n-1

    Bubble Sort · 1 hour ago
  - vedraiyani

    I have run it in codeblock giving output 4

    C | Operators | Question 5 · 9 hours ago
  - dileep kumar Grandhi

    please explain how "s" has come in output

    C | Operators | Question 12 · 11 hours ago
  - ASBGeek

    Can someone explain how the assigement...

    GATE | GATE-CS-2014-(Set-2) | Question 21 · 11 hours ago
  - Zsw-seu

    I think there is a problem with binarySearch....

    Binary Insertion Sort · 20 hours ago
  - Kuch_Bhi

    Please UPDATE the answer. option A is correct.

    GATE | GATE-CS-2003 | Question 57 · 22 hours ago
-