

Hugh

You know something and I know nothing.

[博客园](#)[首页](#)[管理](#)

随笔 - 73 文章 - 0 评论 - 8

数据结构 | 双向链表简单实现及图示

双向链表

和单向链表相比有以下优势：

插入删除不需要移动元素外，可以原地插入删除

可以双向遍历

初始化+尾插法图示：//head始终指向头结点，p指向尾节点，方便后续算法使用

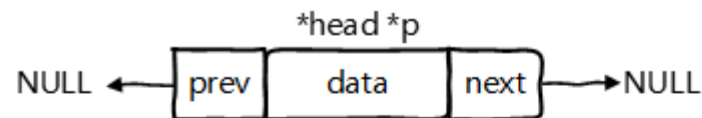
公告

2018年9月						
<	日	一	二	三	四	五
	26	27	28	29	30	1
	2	3	4	5	6	7
	9	10	11	12	13	14
	16	17	18	19	20	21
	23	24	25	26	27	28
	30	1	2	3	4	5

随笔分类(87)

[C语言\(15\)](#)[Photoshop\(1\)](#)[读书\(1\)](#)[考研路\(7\)](#)[前端\(22\)](#)[嵌入式\(24\)](#)

1.初始化时p指向头结点



2.新增一个结点

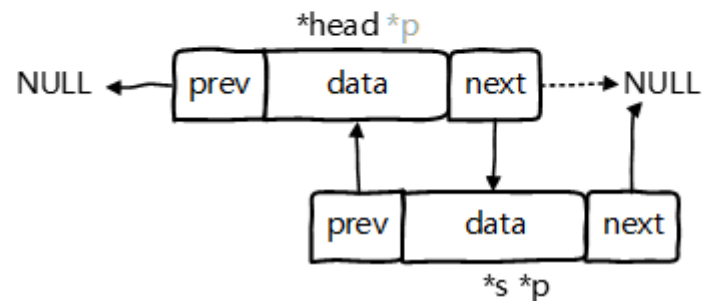
*s申请内存空间

p->next指向尾节点

s->prev指向头结点

s->next指向NULL

p指向尾节点



3.新增一个结点

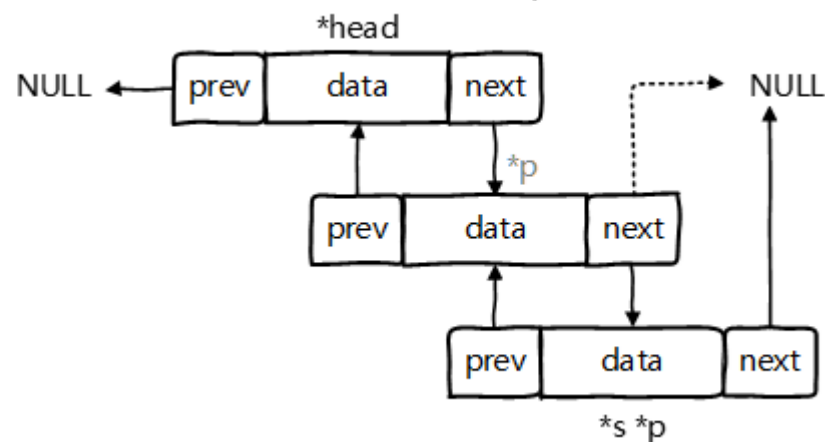
*s申请内存空间

p->next指向尾节点

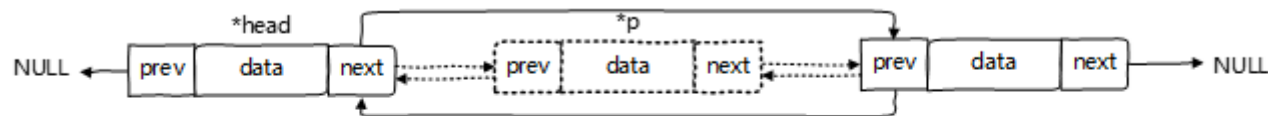
s->prev指向头结点

s->next指向NULL

p指向尾节点



删除单个图示:



删除某个结点:

1. p的下一个结点的prev指向p的上一个结点
2. p的上一个结点的next指向p的下一个结点
3. 释放p

实现代码:

数据结构(15)

网工(2)

随笔档案(73)

2017年8月 (3)

2017年7月 (25)

2017年6月 (8)

2017年5月 (22)

2017年4月 (15)

积分与排名

积分 - 58886

排名 - 7261

最新评论

1. Re:51单片机 | 定时器中断应用实例
@qaw13123你好, 博主的代码15行处, else并不会产生21份计数, 原因是博主用的是++count,在c中++count与count++是不一样的, 前者是先加在等, 不知道说的是否正确, 吸往可以给.....

--程序大海中的一条鱼

2. Re:51单片机 | 定时器中断应用实例
很棒的文章, 时间计算部分一看豁然开朗。有一点小小的疑问: 实现代码的中断子程序里15行不应该加else吧, 这样的话count为0~20都会占用一次中断, 就成了21次了, 会有5%的时间误差。

--qaw13123

3. Re:读源码 | metisMenu侧边栏插件



```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 typedef struct Node pNode;
5 struct Node
6 {
7     int data;
8     pNode *prev, *next;
9 };
10 /* 初始化链表, 尾插法 */
11 pNode *InitList(pNode **head, int n)
12 {
13     pNode *p, *s;
14     (*head) = (pNode *)malloc(sizeof(pNode));
15     if ((*head) == NULL)
16         exit(0);
17     (*head)->next = NULL; //head的prev和next均指向NULL
18     (*head)->prev = NULL;
19     p = (*head); //p指向head
20     int i;
21     for (i = 0; i < n; ++i)
22     {
23         s = (pNode *)malloc(sizeof(pNode));
24         if (s == NULL)
25             exit(0);
26         printf("Input the value of the %dth node:", i + 1);
27         scanf("%d", &s->data);
28         s->next = NULL;
29         p->next = s;
30         s->prev = p;
31         p = s; //p指向尾节点
32     }

```

老哥请问你的简单版 v1.0, 怎么实现让一个li处于active状态啊? 我去官网发现了一个这样的方法 `$("#menu").metisMenu({activeClass: 'active'})`.....

--发给官兵

4. Re:读源码 | metisMenu侧边栏插件

666

--Harmonia

5. Re:JavaScript | 数据属性与访问器属性
@小若风是的, 我也正在学习中, 所以深入的东西也讲不了什么.....

--hugh.dong

6. Re:JavaScript | 数据属性与访问器属性
很基础。。。

--小若风

7. Re:HTML5 | Web Storage

`<script>alert("董大咪");</script>`

--DogEgg

8. Re:数据结构 | 循环队列 (基本操作及图示)

正在学习, 学完后再把二叉树做出来

--大风98

阅读排行榜

1. Photoshop | 快速抠头发 (调整边缘/选择并遮住) (22532)

2. 数据结构 | 循环队列 (基本操作及图示) (13824)

3. 数据结构 | 双向链表简单实现及图示 (12178)

4. 读源码 | metisMenu侧边栏插件(9151)

5. 51单片机 | 实现四相步进电机控制实例 (6800)

```
33     return p;
34 }
35 /* 遍历打印 */
36 void PrintList(pNode *head)
37 {
38     pNode *p;
39     p = head->next;
40     if (head->next == NULL)
41         printf("the list is empty\n");
42     while(p != NULL)
43     {
44         printf("%d ", p->data);
45         p = p->next;
46     }
47     printf("\n");
48 }
49 /* 清空链表 */
50 void DeleteList(pNode **head)
51 {
52     pNode *p;
53     while((*head)->next != NULL)
54     {
55         p = (*head);
56         p->next->prev = NULL;
57         (*head) = p->next;
58         free(p);
59     }
60 }
61 /* 查找链表内的某个值 */
62 int SearchList(pNode *head)
63 {
64     int number;
65     printf("Values are about to be deleted:");
66     scanf("%d", &number);
67     pNode *p;
```

6. 51单片机 | 定时/计数器原理及结构 (T0和T1) (6219)
7. 51单片机 | 定时器中断应用实例(5328)
8. 数据结构 | 链表队列 (基本操作及图示) (4766)
9. 数据结构 | 双向循环链表实现及图示 (3412)
10. 51单片机 | 串口通信实验 (模拟串口通信 / 多机通信实例) (3094)

评论排行榜

1. 51单片机 | 定时器中断应用实例(2)
2. 读源码 | metisMenu侧边栏插件(2)
3. JavaScript | 数据属性与访问器属性(2)
4. HTML5 | Web Storage(1)
5. 数据结构 | 循环队列 (基本操作及图示) (1)

```
68     p = head->next;
69     while(p != NULL)
70     {
71         if (p->data == number)
72         {
73             return number;
74         }
75         p = p->next;
76     }
77     return 0;
78 }
79 /* 删除链表中某个元素, 令p的前驱节点和后驱节点相互指向即可, 如果p是尾节点则直接将前驱节点指向NULL*/
80 void DelNumqList(pNode **head, int n)
81 {
82     int i;
83     pNode *p;
84     p = (*head)->next;
85     for (i = 1; i < n; ++i)
86         p = p->next;
87     if(p->next == NULL)
88     {
89         p->prev->next = NULL;
90         free(p);
91     }
92     else
93     {
94         p->next->prev = p->prev;
95         p->prev->next = p->next;
96         free(p);
97     }
98 }
99 int main(int argc, char const *argv[])
100 {
101     int n, element, flag;
102     pNode *head, *last;
```

```
103  /***** /
104  printf("Please input the size of the list:");
105  scanf("%d", &n);
106  last = InitList(&head, n); //初始化链表并赋值, 返回尾节点last
107  printf("%d %d \n", head->next->data, last->data); //打印为第一个元素和最后一个元素
108  PrintList(head);
109  /***** /
110  flag = SearchList(head); //搜索某个值并删除节点
111  if (flag > 0 && flag <= n)
112  {
113      DelNumqList(&head, flag);
114      PrintList(head);
115  }
116  else
117      printf("Element does not exist, cannot be deleted\n");
118  /***** /
119  DeleteList(&head); //清空列表
120  PrintList(head);
121  return 0;
122 }
```



分类: C语言 , 数据结构

好文要顶

关注我

收藏该文



[hugh.dong](#)

[关注 - 31](#)

[粉丝 - 30](#)

[+加关注](#)

2

0

posted @ 2017-04-29 15:48 hugh.dong 阅读(12178) 评论(0) 编辑 收藏

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。



最新知识库文章:

- 为什么说 Java 程序员必须掌握 Spring Boot ?
- 在学习中，有一个比掌握知识更重要的能力
- 如何招到一个靠谱的程序员
- 一个故事看懂“区块链”
- 被踢出去的用户
- » [更多知识库文章...](#)