

2.4 单向链表之逆序,清除链表_物联网 / 嵌入式工程师 - 慕课网

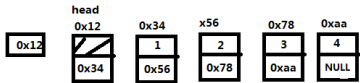
“ 慕课网慕课教程 2.4 单向链表之逆序,清除链表涵盖海量编程基础技术教程，以图图文表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

4. 单向链表之逆序,清除链表

逆序就是把链表中现有的数据，按照相反的顺序来进行存储。例如，原始存储数据为 1 3 5 7 9。逆序之后的存储数据为 9,7,5,3,1。

清除链表就是把链表中所有的结点（包括头结点）都删除。

原来链表数据：



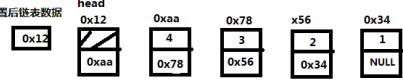
要求实现链表逆置：

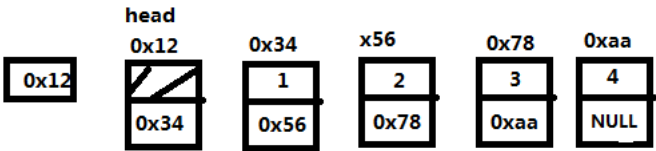
思路：1.把第一个有效数据结点的指针域置为NULL。
2.从第二个结点0x56开始，向后遍历依次向头结点后插入。

原：1 2 3 4 5，要求大家变成5 4 3 2 1
思路：从第二个有效数据开始，依次把数据向后头结点后插入，插入完之后，第一个有效数据的结点就是我们最后的一个结点。

```
int reverse_linklist(linknode *head)
{
    //思路：1、判断链表是否是空链表，若是空链表则返回-1
    // 2、定义指针变量p，保存第二个有效数据结点的地址。
    // 3、并把第一个有效数据结点的指针域置为NULL
    // 4、从p开始遍历链表，用q保存p后面一个结点，并插入到头结点后。
    // 5、数据插入完成，更新p的值，把q赋值给p，继续下一次操作
}
```

逆置后链表数据





思路：

1.从头节点开始删除，删除当前结点。

2.输出后一个结点的数据。

```
void reverse_data_linklist(linknode_t *head)
{
    linknode_t *p = NULL;
    linknode_t *q = NULL;

    p = head->next->next;
    head->next->next = NULL;

    while(p != NULL)
    {
        q = p->next;

        p->next = head->next;
        head->next = p;
    }
}
```

```
        p = q;
    }

    return ;
}

void clean_up_linklist(linknode_t *head)
{

    linknode_t *p = head;
    linknode_t *q = NULL;
    while(p != NULL)
    {
        q = p->next;
        printf_data_linklist(p);
        free(p);
        p = q;
    }

    return ;
}
```

创建一个单向链表, 把 1,5,3,7,9**** 插入链表到链表中, 然后逆序链表, 最后清除整个链表。

全文完

本文由 简悦 SimpRead 优化, 用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta, 点击查看详细说明

