

3.3 链式栈讲解_物联网 / 嵌入式工程师 - 慕课网

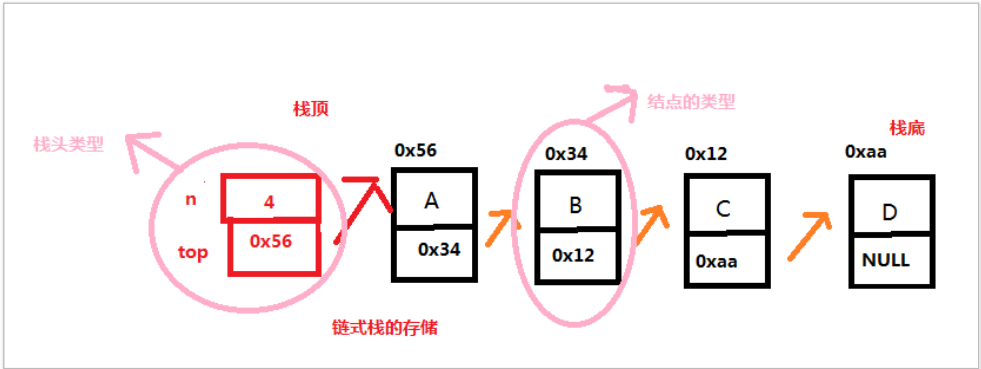
“ 慕课网慕课教程 3.3 链式栈讲解涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

一. 概念讲解

- 概念

链式栈：插入操作和删除操作均在链表头部进行，链表尾部就是栈底，栈顶指针就是头指针。

- 图形



- 本质

栈头 + 不带头节点的链表

二. 数据类型的设计

```
//数据元素的类型
typedef char data_t;

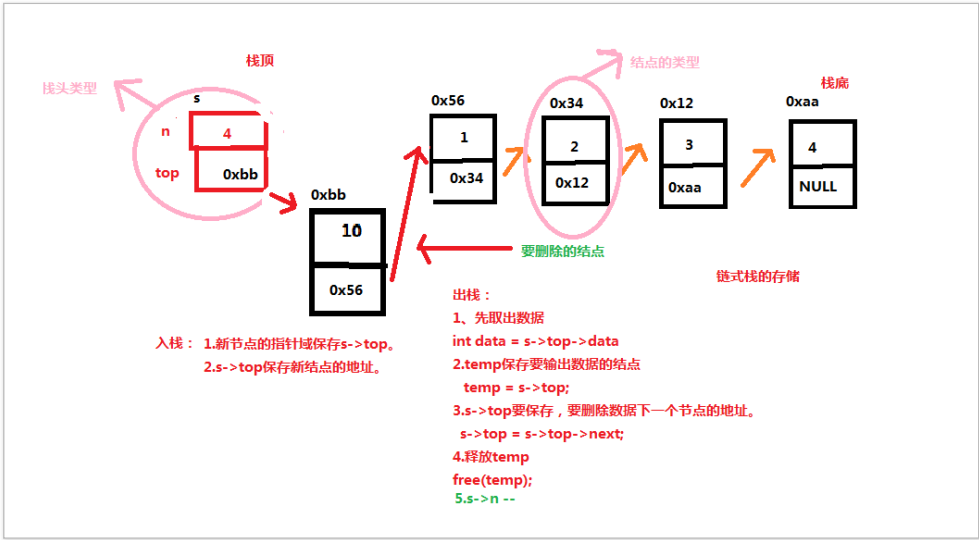
//结点类型
typedef struct node
{
    data_t data;
    struct node *next;
}linknode_t;

//栈头类型
typedef struct
{
    //栈顶指针
    linknode_t *top;

    //记录当前栈中元素个数
    int n;
}linkstack_t;
```

三. 入栈和出栈操作

- 入栈和出栈



四. 操作示例

- 创建一个空栈

```
//1.创建空的链式栈,只有栈头,没有结点,栈头中top == NULL,n == 0
linkstack_t *create_empty_linkstack()
{
    linkstack_t *s = NULL;

    //1.为栈头分配空间
    s = (linkstack_t *)malloc(sizeof(linkstack_t));
    //2.初始化top和n
    s->top = NULL;
    s->n = 0;

    //3.返回栈头首地址
    return s;
}
```

判断栈是否为空

```
//2.链式栈的判空,以top值与NULL比较为准
int is_empty_linkstack(linkstack_t *s)
{
    return s->top == NULL;
}
```

入栈

```
//3.入栈push,类似单链表的头插法
int push_linkstack(linkstack_t *s,data_t data)
{
    linknode_t *temp = NULL;

    //1.为新结点分配空间,存入数据
    temp = (linknode_t *)malloc(sizeof(linknode_t));
    temp->data = data;
```

```
//类似单链表的头插法，插入数据
temp->next = s->top;

//更新top指针和n
s->top = temp;
s->n++;

return 0;
}
```

出栈

```
//4.出栈pop
data_t pop_linkstack(linkstack_t *s)
{
    linknode_t *temp = NULL;
    data_t data;

    //1.保存删除结点(原栈顶元素)首地址,
    temp = s->top;
    //2.取出删除结点数据,
    data = temp->data;
    //3.更新top指针，用删除结点的指针域更新
    s->top = temp->next;
    //4.释放删除结点空间
    free(temp);
    temp = NULL;
    //5.更新n
    s->n--;

    //6.返回出栈元素的值
    return data;
}
```

获得栈顶元素

```
//5.取栈顶元素
data_t get_top_data(linkstack_t *s)
{
    //取栈顶元素的值返回
    return s->top->data;
}
```

主函数

```
int main()
{
    linkstack_t *s = NULL;
    data_t array[] = {'a', 'n', 'i', 'h', 'c'};
    int i = 0;

    s = create_empty_linkstack();

    for(i = 0; i < sizeof(array)/sizeof(array[0]); i++)
    {
        push_linkstack(s, array[i]);
    }

    printf("Top data: %c\n", get_top_data(s));

    while(!is_empty_linkstack(s))
    {
        printf("%c", pop_linkstack(s));
    }
    putchar('\n');

    return 0;
}
```

- 划线
- 写笔记

学习要认真，笔记应当先



公开笔记 0/1000 提交



Sunny_SunshineX

删除 编辑

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

